

Exemple de fonction @ECARTMED

Le tableau ci-dessous dresse la liste des ventes immobilières pour le mois d'avril. L'âge de chacun des bâtiments est exprimé en années dans un champ nommé LISTE_AGE (C2..C8). Vous désirez connaître l'écart médian de ces âges:

@ECARTMED(LISTE_AGE) = 15,18

A	-----	A	-----	B	-----	C	-----	D	-----
1	ADRESSE			ARRDT		AGE		PRIX (F)	
2	1, square Nollet			17		48		1.030.000	
3	22, bd St-Germain			14		22		2.735.000	
4	176, rue Legendre			17		21		945.000	
5	36, rue Monge			5		70		1.896.000	
6	18, rue Grande			8		52		3.218.000	
7	56, rue Dunois			13		42		1.365.000	
8	80, rue de Rivoli			1		23		4.826.000	

Exemple de fonction @CORREL

Vous voulez caractériser l'éventuelle relation entre la taille et le poids d'une dizaine d'individus après avoir reporté les mesures dans une feuille de travail.

@CORREL(A2..A11;B2..B11) = 0,384947

A	-----	A	-----	B	--
1		TAILLE (cm)		POIDS (kg)	
2		190,50		72,73	
3		187,96		86,36	
4		175,26		68,18	
5		175,26		76,37	
6		180,34		77,27	
7		180,34		72,73	
8		187,96		75,00	
9		172,72		68,18	
10		177,80		70,46	
11		179,07		86,36	

Exemple de fonction @COVAR

Vous voulez caractériser l'éventuelle relation entre la taille et le poids d'une dizaine d'individus après avoir reporté ces mesures dans une feuille de travail.

@COVAR(A2..A11;B2..B11) = 13,8872

A	-----	A	-----	B	--
1		TAILLE (cm)		POIDS (kg)	
2		190,50		72,73	
3		187,96		86,36	
4		175,26		68,18	
5		175,26		76,37	
6		180,34		77,27	
7		180,34		72,73	
8		187,96		75,00	
9		172,72		68,18	
10		177,80		70,46	
11		179,07		86,36	

Exemple de fonction @@

La feuille ci-dessous présente un tableau de commission de ventes. La formule :

`@SI(C3="F";"C7";@SI(C3="T";"C8";@ERR))` de la cellule A10, affiche en résultat C7 ou C8, selon que le code produit entré en C3 est F ou T. La formule `@@(A10)`, enregistrée en C4, restitue le contenu de la cellule dont l'adresse est donnée par la formule figurant en A10. Si vous entrez en C3 une valeur autre que l'un des deux codes produit ci-dessus, les fonctions `@SI` et `@@` donnent toutes deux ERR.

A	-----	A	-----	B	-----	C	-----
1		TABLEAU DE COMMISSION DE VENTES					
2							
3		Entrez le code produit :		F			
4		Taux de commission :		5%			
5							
6		Produit		Code		Taux	
7		Fraiseuse		F		5%	
8		Tronçonneuse		T		3%	
9							
10		C7					

@ABS

@ABS(x) calcule la valeur absolue (valeur positive) de x.

Argument

x peut être une valeur quelconque, ou le nom ou l'adresse d'une cellule contenant une valeur.

Remarque

Utilisez -@ABS si vous souhaitez générer obligatoirement un résultat négatif.

Exemples

@ABS(A5) donne 25 si la cellule A5 contient 25, -25, ou une formule produisant l'une de ces valeurs.

- @ABS(A5) donne -25 si la cellule A5 comporte 25, -25, ou une formule générant l'un de ces résultats.

@ABS(DEBUT-FIN) donne 5,6 si DEBUT et FIN contiennent une combinaison quelconque de nombres positifs ou négatifs présentant une différence de 5,6 ou de -5,6 (5,6-0 ; 0-5,6 ; 6-0,4 par exemple).

@INTCOUR

@INTCOUR(*livraison*; *émission*; *premier-intérêt*; *coupon*; [*valeur-remboursement*]; [*fréquence*]; [*base*]) calcule les intérêts courus pour des obligations à intérêt périodique. @INTCOUR prend en charge les périodes de coupon courte, standard et longue.

Arguments

livraison est la date d'achat de l'obligation, exprimée en nombre date. Si *livraison* est inférieure à *émission*, @INTCOUR retourne ERR.

émission est la date d'émission de l'obligation, exprimée en nombre date.

premier-intérêt correspond à la première date d'intérêt de l'obligation, exprimée en nombre date. Si *premier-intérêt* est inférieur ou égal à *émission*, @INTCOUR retourne ERR.

coupon est le taux nominal annuel de l'obligation. Ce peut être n'importe quel nombre positif ou nul.

valeur_remboursement est un argument facultatif qui permet de spécifier la valeur de remboursement de l'obligation à maturité, c'est-à-dire le principal à payer à cette date. *valeur_remboursement* est un nombre positif. Si vous l'omettez, 1-2-3 utilise 100.

fréquence est un argument facultatif permettant d'indiquer la fréquence de paiement des coupons.

<u>fréquence</u>	<u>Fréquence de paiement des coupons</u>
1	Annuelle
2	Semi-annuelle ; valeur par défaut si vous omettez l'argument
4	Trimestrielle
12	Mensuelle

base est un argument facultatif qui spécifie le type de base de comptage des jours à prendre en compte. *base* est une valeur qui appartient au tableau suivant :

<u>base</u>	<u>Base de comptage des jours</u>
0	30/360 ; défaut si vous omettez l'argument
1	Réel/réel
2	Réel/360
3	Actual/365
4	Européen 30/360

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Exemples

Une obligation est livrable le 1er juillet 1993, a été émise le 1er décembre 1992 et a le 1er juin 1993 pour première date d'intérêt. Les coupons sont payés semi-annuellement; le taux annuel est de 5,5%. La valeur de remboursement est de 100 F à l'échéance. Le décompte des intérêts est de type 30/360.

La formule ci-dessous vous permet de déterminer les intérêts courus produits par l'obligation :

@INTCOUR(@DATE(93;7;1);@DATE(92;12;1);@DATE(93;6;1);0,055;100;2;0) = 0,458333

Fonctions similaires

@INTCOUR2 calcule les intérêts courus produits par les obligations à paiement d'intérêts périodiques, à l'aide des conventions japonaises.

@PRIX calcule le prix d'un bon rapporté à une valeur nominale de 100. @RENDEMENT calcule le taux de rendement de bons à intérêt périodique. @DURATION calcule la durée à l'origine et @SENSIBILITETITRE, la sensibilité à l'origine d'un titre de 100 F à intérêt périodique.

@ACOS

@ACOS(x) calcule l'arc cosinus (cosinus inverse) d'un angle à partir de son cosinus x. L'angle ainsi obtenu, exprimé en radians, est compris entre 0 et π (soit 0 à 180 degrés).

Argument

x, cosinus d'un angle donné, peut être une valeur comprise entre -1 et 1, ou le nom ou l'adresse d'une cellule contenant cette valeur.

Exemples

Le cosinus d'un angle z est 0,5. Pour déterminer la valeur de l'angle z, utilisez **@ACOS(0,5)** qui renvoie 1,0472 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(1,0472)**. Le résultat obtenu est alors 60 degrés.

Fonctions similaires

@COS calcule le cosinus d'un angle. @ACOSH calcule l'arc cosinus hyperbolique d'un angle.

@ACOSH

@ACOSH(x) calcule l'arc cosinus hyperbolique (cosinus hyperbolique inverse) d'un angle à partir de son cosinus hyperbolique x .

Arguments

x , cosinus hyperbolique d'un angle donné, doit être une valeur supérieure ou égale à 1.

Exemples

Le cosinus hyperbolique d'un angle z est 2. Pour déterminer la valeur de l'angle, utilisez **@ACOSH(2)** qui renvoie 1,316958 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(1,316958)**. Le résultat obtenu est alors 75,46 degrés.

Fonctions similaires

@ACOS calcule l'arc cosinus d'un angle. **@COSH** calcule le cosinus hyperbolique d'un angle.

@ACOTAN

@ACOTAN(x) calcule l'arc cotangente (cotangente inverse) d'un angle à partir de sa cotangente x . L'angle ainsi obtenu, exprimé en radians, est compris entre 0 et π (soit 0 et 180 degrés).

Argument

x , cotangente d'un angle donné, peut être une valeur comprise entre $-2^{32}\pi$ et $2^{32}\pi$.

Exemples

La cotangente d'un angle z est 1,732051. Pour déterminer la valeur de l'angle z , utilisez **@ACOTAN(1,732051)** qui renvoie 0,523599 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(0,523599)**. Le résultat obtenu est alors 30 degrés.

Fonctions similaires

@ATAN calcule l'arc tangente d'un angle et **@COTAN** sa cotangente.

@ACOTANH

@ACOTANH(x) calcule l'arc cotangente hyperbolique (cotangente hyperbolique inverse) d'un angle à partir de sa cotangente hyperbolique x.

Arguments

x, cotangente hyperbolique d'un angle, doit être une valeur inférieure à -1 ou supérieure à 1.

Exemples

La cotangente hyperbolique d'un angle z est 2. Pour déterminer la valeur de l'angle z, utilisez @ACOTANH(2), qui renvoie 0,549306 radians. Pour convertir cette valeur en degrés, faites appel à la fonction @RADENDEG(0,549306). Le résultat obtenu est alors 31,47 degrés.

Fonctions similaires

@ACOTAN calcule l'arc cotangente d'un angle et @COTANH, sa cotangente hyperbolique.

@ACOSEC

@ACOSEC(x) calcule l'arc cosécante (cosécante inverse) d'un angle à partir de sa cosécante x . Le résultat de la fonction **@ACOSEC** est un angle exprimé en radians, compris entre $-\pi/2$ et $\pi/2$ (soit -90 et 90 degrés).

Arguments

x , cosécante d'un angle, est une valeur inférieure ou égale à -1 ou supérieure ou égale à 1.

Exemples

La cosécante d'un angle z est 1,743447. Pour déterminer la valeur de l'angle z , utilisez **@ACOSEC(1,743447)**, qui renvoie 0,610865 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(0,610865)**. Le résultat obtenu est alors 35 degrés.

Fonctions similaires

@COSEC calcule la cosécante d'un angle et **@ACOSECH** l'arc cosécante hyperbolique.

@ACOSECH

@ACOSECH(x) calcule l'arc cosécante hyperbolique (cosécante hyperbolique inverse) d'un angle à partir de sa cosécante hyperbolique x . Le résultat de la fonction **@ACOSECH** est un angle exprimé en radians.

Arguments

x , cosécante hyperbolique d'un angle, doit être une valeur différente de 0.

Exemples

La cosécante hyperbolique d'un angle z est 1,54. Pour déterminer la valeur de l'angle z , utilisez **@ACOSECH(1,54)**, qui renvoie 0,61068 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(0,61068)**. Le résultat obtenu est alors 34,99 degrés.

Fonctions similaires

@COSECH calcule la cosécante hyperbolique d'un angle et **@ACOSEC** l'arc cosécante.

@ASEC

@ASEC(x) calcule l'arc sécante (sécante inverse) d'un angle à partir de sa sécante x . Le résultat de la fonction **@ASEC** est un angle exprimé en radians, dont la valeur est comprise entre 0 et π (soit 0 et 180 degrés).

Arguments

x , sécante d'un angle, doit être une valeur inférieure ou égale à -1 ou supérieure ou égale à 1.

Exemples

Dans un triangle rectangle, la sécante d'un angle z est 2. Pour déterminer la valeur de l'angle z , utilisez **@ASEC**(2), qui renvoie 1,047198 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG**(1,047198). Le résultat obtenu est alors 60 degrés.

Fonctions similaires

@ASECH calcule l'arc sécante hyperbolique d'un angle et **@SEC** sa sécante.

@ASECH

@ASECH(x) calcule l'arc sécante hyperbolique (sécante hyperbolique inverse) d'un angle à partir de sa sécante hyperbolique x .

Arguments

x , sécante hyperbolique d'un angle, doit être une valeur strictement supérieure à 0 et inférieure ou égale à 1.

Exemples

La sécante hyperbolique d'un angle z est 0,5. Pour déterminer la valeur de l'angle z , utilisez **@ASECH(0,5)**, qui renvoie 1,316958 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(1,316958)**. Le résultat obtenu est alors 75,46 degrés.

Fonctions similaires

@ASEC calcule l'arc sécante d'un angle. **@SECH** calcule la sécante hyperbolique d'un angle.

@ASIN

@ASIN(x) calcule l'arc sinus (sinus inverse) d'un angle à partir de son sinus x . L'angle ainsi obtenu, exprimé en radians, est compris entre $-\pi/2$ et $\pi/2$ (soit -90 et 90 degrés).

Argument

x , sinus d'un angle donné, peut être une valeur comprise entre -1 et 1, ou le nom ou l'adresse d'une cellule contenant cette valeur.

Exemples

Le sinus d'un angle z est 0,66. Pour déterminer la valeur de l'angle z , utilisez @ASIN(0,66), qui renvoie 0,72082 radians. Pour convertir cette valeur en degrés, faites appel à la fonction @RADENDEG(0,72082). Le résultat obtenu est alors 41,3 degrés.

Fonction similaire

@SIN calcule le sinus d'un angle. @ASINH calcule l'arc sinus hyperbolique d'un angle à partir de son sinus hyperbolique.

@ASINH

@ASINH(x) calcule l'arc sinus hyperbolique (sinus hyperbolique inverse) d'un angle à partir de son sinus hyperbolique x.

Arguments

x est une valeur quelconque et représente le sinus hyperbolique d'un angle.

Exemples

Le sinus hyperbolique d'un angle z est 2. Pour déterminer la valeur de l'angle z, utilisez @ASINH(2), qui renvoie 1,443635 radians. Pour convertir cette valeur en degrés, faites appel à la fonction @RADENDEG(1,443635). Le résultat obtenu est alors 82,71 degrés.

Fonctions similaires

@ASIN calcule l'arc sinus d'un angle et @SINH son sinus hyperbolique.

@ATAN

@ATAN(x) calcule l'arc tangente (tangente inverse) d'un angle à partir de sa tangente x . L'angle ainsi obtenu, exprimé en radians, est compris entre $-\pi/2$ et $\pi/2$ (soit -90 et 90 degrés).

Argument

x , tangente d'un angle donné, peut être une valeur quelconque, ou le nom ou l'adresse d'une cellule contenant cette valeur.

Exemples

La tangente d'un angle z est 2. Pour déterminer la valeur de l'angle z , utilisez **@ATAN(2)**, qui renvoie 1,10715 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(1,10715)**. Le résultat obtenu est alors 63,4 degrés.

Fonctions similaires

@ATANH calcule l'arc tangente hyperbolique d'un angle à partir de sa tangente hyperbolique x . **@ATAN2** détermine la valeur d'un angle dont la tangente est y/x . **@TAN** calcule la tangente d'un angle.

@ATAN2

@ATAN2(x;y) calcule l'arc tangente d'un angle à partir de sa tangente y/x . L'angle ainsi obtenu, exprimé en radians, est compris entre $-\pi$ et π (soit -180 et 180 degrés), suivant les signes de x et de y (les différents cas de figure sont présentés dans le tableau ci-après).

Arguments

x et y peuvent être des valeurs quelconques, ou les noms ou adresses des cellules contenant ces valeurs. **@ATAN2** donne la valeur 0 si y est égal à 0, et la valeur ERR, si x et y sont tous deux nuls.

Remarques

A la différence de **@ATAN**, **@ATAN2** donne une valeur comprise entre $-\pi$ et π . Le tableau suivant récapitule les valeurs possibles de **@ATAN2**.

Si x et y sont positifs, le résultat est compris entre 0 et $\pi/2$ (quadrant I).

Si x est négatif et y positif, le résultat est compris entre $\pi/2$ et π (quadrant II).

Si x et y sont négatifs, le résultat est compris entre $-\pi$ et $-\pi/2$ (quadrant III).

Si x est positif et y négatif, le résultat est compris entre $-\pi/2$ et 0 (quadrant IV).

Lorsque x et y sont tous deux positifs (quadrant I), ou que x est positif et y négatif (quadrant IV), **@ATAN2** donne le même résultat que **@ATAN**.

Exemples

Les deux côtés formant l'angle droit d'un triangle rectangle mesurent respectivement 1 et 2. Pour évaluer le plus grand des deux autres angles, utilisez **@ATAN2(1;2)**, qui renvoie 1,10715 radians. Pour convertir cette valeur en degrés, faites appel à la fonction **@RADENDEG(1,10715)**. Le résultat obtenu est alors 63,4 degrés.

Fonctions similaires

@ATAN calcule l'arc tangente d'un angle, connaissant sa tangente x . **@TAN** calcule la tangente d'un angle.

@ATANH

@ATANH(x) calcule l'arc tangente hyperbolique (tangente hyperbolique inverse) d'un angle à partir de sa tangente hyperbolique x.

Arguments

x, tangente hyperbolique d'un angle, est une valeur comprise entre -1 et 1.

Exemples

La tangente hyperbolique d'un angle z est 0,544736. Pour déterminer la valeur de l'angle z, utilisez @ATANH(0,544736), qui renvoie 0,610865 radians. Pour convertir cette valeur en degrés, faites appel à la fonction @RADENDEG(0,610865). Le résultat obtenu est alors 35 degrés.

Fonctions similaires

@ATAN calcule l'arc tangente et @TANH la tangente hyperbolique d'un angle.

@ATAN2, renvoie la mesure d'un angle dont la tangente est y/x (ou dont les côtés sont x et y).

@ECARTMED

@ECARTMED(*liste*) calcule l'écart médian des valeurs d'une *liste*, c'est-à-dire la moyenne des valeurs absolues des écarts à la moyenne de ces valeurs.

Arguments

liste peut comporter un ou plusieurs des éléments suivants: nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

@ECARTMED

Fonctions similaires

@STD et @STDNUM calculent l'écart type des valeurs d'une liste. @SOMECART2 calcule, dans une liste, la somme des carrés des écarts à la moyenne.

@MOYENNE, @MOYENNENUM

@MOYENNE(*liste*) calcule la moyenne des valeurs de *liste*.

@MOYENNENUM(*liste*) calcule la moyenne des valeurs de *liste*, sans tenir compte des cellules vides ou contenant un libellé.

Arguments

liste peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

On considère que les cellules A1..A5 contiennent respectivement: le libellé JANVIER, les valeurs 160, 227, 397 et 227.

@MOYENNE(A2..A5) renvoie 252,75 .

@MOYENNE(A1..A5) renvoie 202,20 car le libellé JANVIER entre dans la moyenne bien qu'il soit évalué à 0.

@MOYENNENUM(A1..A5) renvoie 252,75 car le libellé JANVIER n'est pas pris en compte par la fonction

@MOYENNENUM.

Fonctions similaires

@BMOYENNE calcule la moyenne des valeurs d'une rubrique de table de base de données répondant aux critères spécifiés. **@MOYGEOM** renvoie la moyenne géométrique et **@MOYHARM**, la moyenne harmonique des valeurs d'une liste. **@MEDIANE** calcule la valeur médiane d'une liste.

@BESSELI, @BESSELJ, @BESSELK, @BESSELY

@BESSELI($x;n$) renvoie la fonction de Bessel modifiée d'ordre n : $I_n(x)$.

@BESSELJ($x;n$) renvoie la fonction de Bessel d'ordre n : $J_n(x)$.

@BESSELK($x;n$) renvoie la fonction de Bessel modifiée d'ordre n : $K_n(x)$.

@BESSELY($x;n$) renvoie la fonction de Bessel d'ordre n : $Y_n(x)$, (connue également sous le nom de Fonction de Neumann).

Arguments

x , valeur quelconque, est la variable à laquelle s'applique la fonction de Bessel.

n est l'ordre de la fonction de Bessel (entier positif ou nul).

Remarques

@BESSELI, @BESSELJ, @BESSELK, et @BESSELY approchent les fonctions de Bessel respectives à $\pm 5 \cdot 10^{-8}$ près.

Les fonctions de Bessel sont fréquemment utilisées dans les problèmes de symétrie cylindrique relatifs à la propagation d'ondes, la mécanique des fluides, l'élasticité, la diffusion, ...

Exemples

@BESSELI(2;2) = 0,688948

@BESSELJ(1;0) = 0,765198

@BESSELK(3;0) = 0,03474

@BESSELY(1;1) = -0,781213

@BETA

@BETA(z;w) calcule la fonction Béta appliquée aux variables z et w .

Arguments

z et w sont des valeurs quelconques autres que zéro ou un entier négatif.

Remarques

Le résultat de la fonction @BETA est exact sur, au moins, les 6 premiers chiffres.

Exemples

@BETA(0,5;0,5) = 3,141593

Fonctions similaires

@BETA calcule la fonction Béta incomplète. @GAMMA calcule la fonction Gamma.

@BETAI

@BETAI($a;b;x$) calcule la fonction Béta incomplète.

Arguments

a et b sont des valeurs quelconques.

x est une valeur comprise entre 0 et 1.

Remarques

Le résultat de la fonction @BETA est exact sur, au moins, les 6 premiers chiffres.

Exemples

@BETAI(5;0,5;0,668271) = 0,050012

Fonctions similaires

@BETA calcule la fonction Béta.

@BINOMIALE

@BINOMIALE(*essais*; *succès*; *probabilité*; [*type*]) calcule, conformément à la loi binomiale, la probabilité d'un événement ou la distribution cumulée (effectifs cumulés).

Arguments

essais est le nombre d'épreuves réalisées. *essais* est un nombre entier positif.

succès est le nombre d'événements favorables à obtenir sur *essais* épreuves. C'est un entier positif ou nul, inférieur ou égal à *essais*.

Si *essais* et *succès* ne sont pas entiers, 1-2-3 ne prendra en compte que leur partie entière.

probabilité est la probabilité de l'événement favorable et est donc comprise entre 0 et 1 inclus.

type est un argument facultatif qui permet à 1-2-3 de calculer soit la probabilité d'un événement, soit sa distribution cumulée:

<u><i>type</i></u>	<u>1-2-3 calcule</u>
0	La probabilité pour que le nombre d'événements favorables soit exactement égal à <i>succès</i> (0 est la valeur par défaut)
1	La probabilité pour que le nombre d'événements favorables soit au plus égal à <i>succès</i>
2	La probabilité pour que le nombre d'événements favorables soit au moins égal à <i>succès</i>

Remarques

@BINOMIALE donne un résultat exact à $\pm 3 \cdot 10^{-7}$ près.

Exemples

Vous avez choisi au hasard dix personnes pour sélectionner le meilleur café parmi deux marques proposées. Vous donnez à chacune d'entre elles deux tasses de café apparemment identiques. Sous l'une, vous avez inscrit la lettre A, sous l'autre B, pour repérer les deux marques. Partant du principe que les participants n'ont aucune prédisposition à préférer le café A au café B, la probabilité pour qu'une personne choisisse le café A est de 50% (soit *probabilité* = 0,5).

On déterminera la probabilité pour qu'exactly 7 personnes parmi les 10 préfèrent le café A de la façon suivante:

@BINOMIALE(10;7;0,5) = 0,117188 soit 11,72%.

De même, pour qu'au moins 7 personnes sur 10 préfèrent le café A:

@BINOMIALE(10;7;0,5;2) = 0,171875 soit 17,19%.

Fonctions similaires

@CRITBINOM calcule la plus grande valeur telle que l'effectif cumulé soit inférieur ou égal à un critère donné.

@COMBIN renvoie le nombre de combinaisons possibles de valeurs. @ARRANG renvoie le nombre d'arrangements possibles de valeurs.

@CELLULE, @POINTCELL

@CELLULE(*attribut;emplacement*) donne des informations sur la première cellule d'*emplacement*. Ces informations sont fonction de l'argument *attribut* spécifié.

@POINTCELL(*attribut*) donne des informations sur la cellule actuelle.

Arguments

emplacement est le nom ou l'adresse d'un champ.

attribut, qui doit figurer entre guillemets, peut être l'un des éléments ci-dessous, ou le nom ou l'adresse d'une cellule contenant l'un de ces éléments.

<u>attributs</u>	<u>Résultat</u>
"coord"	Adresse absolue de la cellule, sous forme complète (lettre de feuille, lettre de colonne et numéro de ligne)
"couleur"	1 si les valeurs négatives apparaissent en couleur 0 dans le cas contraire
"couleurbordbas"	Nombre entier, compris entre 0 et 15, qui indique la couleur du trait inférieur
"couleurborddroit"	Nombre entier, compris entre 0 et 15, qui indique la couleur du trait droit
"couleurbordgauche"	Nombre entier, compris entre 0 et 15, qui indique la couleur du trait gauche
"couleurbordhaut"	Nombre entier, compris entre 0 et 15, qui indique la couleur du trait supérieur
"couleurfond"	Nombre entier compris entre 0 et 239 indiquant la couleur de fond sélectionnée dans la palette de couleurs
"couleurmotif"	Nombre entier compris entre 0 et 239 qui indique la couleur de motif sélectionnée dans la palette de couleurs.
"couleurtexte"	Nombre entier, compris 0 et 239, qui indique la couleur des données sélectionnée dans la palette de couleurs
"datefichier"	Nombre entier indiquant la date et l'heure à laquelle le classeur contenant la cellule a été sauvegardé pour la dernière fois. Sa partie entière est un <u>nombre date</u> et sa partie décimale, un <u>nombre temps</u>
"feuille"	Lettre de feuille, exprimée sous forme d'une valeur comprise entre 1 et 256 (1 pour la feuille A, 2 pour la feuille B, etc.)
"format"	Format de la cellule : M0 à M15 pour le format Monnaie, 0 à 15 décimales F0 à F15 pour le format Fixe, 0 à 15 décimales L pour le format Libre, un libellé ou une cellule vierge %0 à %15 pour le format %, 0 à 15 décimales S0 à S15 pour le format Scientifique, 0 à 15 décimales .0 à .15 pour le format Milliers, 0 à 15 décimales D1 à D5 pour les formats de date et H1 à H4 pour les formats d'heure : D1: JJ-MMM-AA

D2: JJ-MMM

D3: MMM-AA

D4: JJ/MM/AA

D5: JJ/MM

H1: 11:59:59 AM

H2: 11:59 AM

H3: 23:59:59

H4: 59:59

T pour le format Texte

. pour le format Milliers

C pour le format Caché

LL pour le format Libellé

A pour le format Automatique

-- si les valeurs négatives s'affichent en couleur

() si la case Parenthèses est cochée dans la boîte de dialogue obtenue en sélectionnant Style Paramètres de feuille Format ou Style Format des nombres.

"gras"	1 si la cellule est formatée en gras 0 dans le cas contraire
"hauteur"	Hauteur de ligne en points
"italique"	1 si la cellule est formatée en italiques 0 dans le cas contraire
"justifiersurcolonnes "	1 si le type d'alignement des données de la cellule est calculé sur plusieurs colonnes 0 dans le cas contraire
"largeur"	Largeur de colonne
"ligne"	Numéro de ligne compris entre 1 à 8192
"motif"	Nombre entier, compris entre 0 et 63, qui indique le motif
"nomfeuille"	Nom de la feuille (ou dans le cas d'une feuille sans nom, lettre de feuille)
"nomfichier"	Nom du classeur contenant la cellule, unité et répertoire compris
"orientation"	Nombre entier, compris entre 0 et 4, qui indique l'orientation du texte
"parenthèses"	1 si le format de la cellule impose l'affichage de parenthèses 0 dans le cas contraire
"police"	Police des données de la cellule
"préfixe"	' si la cellule contient le préfixe ' (libellé aligné à gauche) " si la cellule contient le préfixe " (libellé aligné à droite) ^ si la cellule contient le préfixe ^ (libellé centré) \ si la cellule contient le préfixe de répétition

	si la cellule contient le préfixe de non-impression pas de préfixe de libellé si la cellule est vierge ou comporte une valeur
"protégé"	1 pour une cellule protégée 0 pour une cellule non protégée
"retourautomatique"	1 si les données sont automatiquement renvoyées à la ligne 0 dans le cas contraire
"rotation"	Nombre entier, compris entre 0 et 90 degrés, qui indique l'angle de rotation du texte
"souligné"	Nombre entier qui indique le type de soulignement et qui peut être compris entre 0 (absence de soulignement) et 3
"taille"	Taille, en points, de la police des données de la cellule
"type"	Type de données figurant dans la cellule : i pour une cellule vierge v pour une cellule contenant une valeur numérique, <u>une formule numérique</u> ou une <u>formule texte</u> l pour une cellule contenant un libellé
"typedonnées"	Type des données de la cellule : b si la cellule est vierge v si la cellule contient un nombre ou une formule numérique l si la cellule contient un libellé ou une formule texte e si la cellule contient la valeur <u>ERR</u> n si la cellule contient la valeur <u>ND</u>
"typeformule"	Type de formule de la cellule : b si la cellule est vierge v si la cellule contient un nombre l si la cellule contient un libellé fv si la cellule contient une formule numérique fl si la cellule contient une formule texte fe si la cellule contient une formule dont le résultat est ERR fn si la cellule contient une formule dont le résultat est ND

Remarques

Pour garantir l'exactitude du résultat, recalculer la feuille de travail en appuyant sur F9 (CALCUL) avant d'utiliser les fonctions @CELLULE ou @POINTCELL.

@CELLULE et @POINTCELL s'utilisent dans les macros, souvent associées à @SI. La fonction @CELLULE permet, par exemple, de contrôler les entrées effectuées pendant l'exécution d'une macro (afin d'empêcher la saisie de certains types de données), ou de mettre en place des branchements conditionnels à l'aide de sous-routines répondant aux données entrées. @CELLULE rend également possible la modification des attributs de cellule par des modèles ou des applications appelés automatiquement en fonction des données entrées par l'utilisateur.

Utilisez @POINTCELL pour déterminer la position du pointeur ou pour évaluer une formule en fonction de la cellule actuelle. Le traitement à effectuer dépendra ainsi directement du contenu ou du type de la cellule.

Vous pouvez remplacer le type 2 nom d'attribut par typedonnées et le type3 nom d'attribut par typeformule.

Exemples

Dans l'exemple suivant, @CELLULE est associée à @SI et @ERR. Si l'utilisateur n'entre aucune valeur dans la cellule MAS, @CELLULE affiche ERR ; elle reproduit le contenu de la cellule dans le cas contraire.

```
@SI(@CELLULE("type";MAS)="v";MAS;@ERR)
```

@POINTCELL peut s'utiliser dans une macro pour tester la présence d'une cellule vierge dans une liste d'éléments : dans l'exemple suivant, si 1-2-3 rencontre ce type de cellule durant l'exécution de la macro, il émet un signal sonore et se branche sur une sous-routine.

```
{SI @POINTCELL("type")="i"}{BIP}{BRANCHE Etape2}
```

Fonctions similaires

@ renvoie le contenu d'une cellule adressée indirectement.

@CAR

@CAR(x) génère le caractère du jeu de caractères LBMCS (Lotus Multibyte Character Set) correspondant au nombre x.

Argument

x peut être un nombre entier, ou le nom ou l'adresse d'une cellule contenant ce type de nombre. Les valeurs qui ne correspondent pas à un caractère du jeu LMBCS donnent ERR. Lorsque x est une valeur décimale, seule sa partie entière est prise en compte.

Remarque

Si le caractère correspondant à x ne peut être affiché à l'écran, 1-2-3 le remplace, le cas échéant, par le caractère le plus approchant, ou par un rectangle plein. Vérifiez que l'imprimante est en mesure de reproduire les caractères demandés.

Exemple

@CAR(34) donne " (guillemet).

@CAR(D9) donne A, si la cellule D9 contient la valeur 65.

Fonction similaire

@CODE restitue le code LMBCS correspondant au caractère spécifié.

@DISTCHI

@DISTCHI(x ;degrés-liberté; $[type]$) renvoie la fonction Chi-carrée (χ^2) appliquée à la variable x .

Arguments

x est la variable passée à la fonction @DISTCHI. Cette variable est interprétée différemment selon la valeur donnée à $type$:

<u>Si type vaut</u>	<u>x représente</u>
0	la valeur critique (ou valeur de seuil) de la variable aléatoire de la fonction χ^2 . x doit être supérieur ou égal à 0; valeur par défaut si vous omettez l'argument.
1	une probabilité (probabilité de risque). x doit être compris entre 0 et 1.

degrés-liberté est un entier positif représentant le nombre de degrés de liberté de la variable aléatoire. Si *degrés-liberté* n'est pas entier, 1-2-3 ne prendra en compte que sa partie entière.

<u>type</u>	<u>1-2-3 renvoie</u>
0	La probabilité de risque correspondant à la valeur de seuil x , valeur par défaut si vous omettez l'argument
1	La valeur de seuil correspondant à la probabilité de risque x , valeur par défaut si vous omettez l'argument

Remarques

@DISTCHI renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$. Si après 100 itérations, 1-2-3 ne peut pas fournir un résultat exact à 0,0000001 près, @DISTCHI renvoie ERR.

La fonction χ^2 est une fonction de distribution mono-variable continue et constitue un cas particulier de la fonction de distribution Gamma.

Utilisez @DISTCHI pour vérifier la validité d'une hypothèse par comparaison entre les résultats obtenus et attendus.

Exemples

@DISTCHI(12,592;6) = 0,05

@DISTCHI(0,05;6;1) = 12,59159

Fonctions similaires

@TESTCHI renvoie la probabilité associée à un test de χ^2 . @DISTF évalue la distribution de F et @DISTT la distribution de Student (distribution de t).

@CHOIX

@CHOIX(*x*; *liste*) donne le *x*ème élément (valeur ou libellé) de *liste*.

Arguments

x peut être une valeur, ou le nom ou l'adresse d'une cellule contenant une valeur. *x* représente le numéro de position d'un élément de *liste*.

liste se compose d'un groupe de valeurs et/ou de libellés, ou du nom ou de l'adresse des cellules contenant ces valeurs et libellés, séparés par le séparateur d'arguments. 1-2-3 numérote chaque entrée de *liste*, puis sélectionne l'entrée correspondant à *x*.

Exemple

Soit une liste de libellés, dans le champ A1..A4, et les numéros de position correspondants (0, 1, 2, 3), en B1..B4. La fonction **@CHOIX**(B3;A1;A2;A3;A4) restitue le libellé présent en A3 et repéré par le numéro de position 2 (B3) dans *liste*.

Fonctions similaires

@CONSH et **@CONSV** renvoient le contenu d'une cellule dans un tableau de consultation horizontale ou verticale.

@INDEX renvoie le contenu d'une cellule située à l'intersection d'une colonne et d'une ligne identifiées par leur numéro de position dans le champ. **@XINDEX** restitue le contenu d'une cellule située à l'intersection d'une colonne

et d'une ligne identifiées par leur contenu. **@CORRESP** indique la position relative d'une cellule identifiée par son contenu.

@CONSMAX renvoie la référence absolue de la cellule qui contient la valeur la plus élevée de liste-

champs. **@CONSMIN** renvoie la référence absolue de la cellule qui contient la valeur la plus basse de liste-champs.

@EPURE

@EPURE(*texte*) élimine les caractères non imprimables de *texte*.

Arguments

texte est du texte entre guillemets, l'adresse ou le nom d'une cellule qui contient un libellé, ou une formule ou fonction @ générant du texte.

Exemples

Après importation de données issues d'un traitement de texte, la cellule A45 contient le texte

le jeu de caractères AICS® permet l'utilisation de ∩ sous 1-2-3v2

@EPURE(A45) = le jeu de caractères AICS permet l'utilisation de sous 1-2-3v2

Fonctions similaires

@CAR retourne le caractère du jeu LMBCS correspondant au numéro de code spécifié. @SUPPRESPEACE élimine les espaces de début et de fin de chaîne, ainsi que les espaces superflus.

@CODE

@CODE(*chaîne*) retourne le numéro de code du premier caractère de *chaîne* dans le jeu de caractères LMBCS (Lotus Multibyte Character Set).

Argument

chaîne peut être du texte mis entre guillemets, le nom ou l'adresse d'une cellule contenant un libellé, ou bien une formule ou fonction @ générant du texte.

Remarque

Si l'argument se réfère à une cellule vierge ou à une valeur, @CODE donne ERR.

Exemples

@CODE("A") donne 65.

@CODE(C5) donne 77, si C5 contient le libellé Mme Jonard (77 est le code LMBCS de M).

Fonction similaire

@CAR restitue le caractère correspondant au numéro de code spécifié du jeu de caractères LMBCS.

@COL

@COL(*champ*) renvoie le nombre de colonnes de *champ*.

Argument

champ peut être le nom ou l'adresse d'un champ.

Remarque

@COL permet de déterminer la largeur d'un champ pour réaliser une opération dépendant de cette information (répétition d'une macro en fonction du nombre de colonnes d'un champ, par exemple).

@COL peut être associée à l'instruction {POUR}, dans une macro effectuant la même opération sur une série de colonnes, afin d'indiquer à quel moment 1-2-3 doit interrompre l'exécution de cette dernière.

Exemples

@COL(D9..J25) donne 7, car *champ* contient 7 colonnes (de D à J).

@COL(RERESULTAT) donne 2, si vous avez affecté au champ B3..C45 le nom RESULTAT.

Fonctions similaires

@CONVREF convertit les lettres de colonne 1-2-3 A à IV (ou les lettres de feuille) en nombres de 1 à 256. @LIGNES et @FEUILLES calculent respectivement le nombre de lignes et de feuilles d'un champ.

@COMBIN

@COMBIN($n;p$) renvoie le nombre de combinaisons de p éléments pris parmi n . L'ordre des éléments dans chaque combinaison est indifférent.

Arguments

n , entier positif ou nul, est le nombre total d'éléments.

p , entier positif ou nul, inférieur ou égal à n , est le nombre d'éléments d'une combinaison.

Si n et p ne sont pas entiers, 1-2-3 ne prend en compte que leur partie entière.

Remarques

@COMBIN renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$ près.

Exemples

Une urne contient 5 boules de couleurs différentes. On en extrait trois au hasard. Le nombre de combinaisons de couleurs que l'on peut obtenir se calcule par:

@COMBIN(5;3) = 10

Fonctions similaires

@BINOMIALE calcule la probabilité d'un événement ou la distribution cumulée. @CRITBINOM calcule la plus grande valeur telle que l'effectif cumulé soit inférieur ou égal à une valeur critère. @ARRANG renvoie le nombre d'arrangements possibles de p éléments pris dans n , l'ordre des éléments dans chaque arrangement n'étant pas indifférent.

@COORD

@COORD(*feuille*; *colonne*; *ligne*; *absolu*) crée une adresse de cellule à partir des valeurs des arguments *feuille*, *colonne* et *ligne*.

Arguments

feuille et *colonne* sont des nombres entiers compris entre 1 et 256, ou les noms ou adresses des cellules contenant ces nombres. *feuille* et *colonne* correspondent aux lettres de feuille et de colonne (1 pour la colonne A, 2 pour la colonne B, etc.).

ligne, nombre entier compris entre 1 et 8192, représente le numéro de ligne.

absolu est un nombre entier compris entre 1 et 8.

Si *feuille*, *colonne*, *ligne* et *absolu* ne sont pas des entiers, 1-2-3 les tronque pour en faire des entiers.

Remarque

@COORD crée une adresse de cellule relative, absolue ou mixte, selon la valeur de l'argument *absolu*. Le tableau ci-dessous présente les valeurs possibles d'*absolu* et leur effet sur l'adresse de cellule A1 de la feuille A.

<u>Valeur de <i>absolu</i></u>	<u>Valeur de @COORD(1;1;1;<i>absolu</i>)</u>
1	\$A:\$A\$1
2	\$A:A\$1
3	\$A:\$A1
4	\$A:A1
5	A:\$A\$1
6	A:A\$1
7	A:\$A1
8	A:A1

Utilisée avec @INDEX, @CONSV ou @CONSH, @COORD permet de créer des adresses de cellule à partir de tableaux de valeurs figurant dans le classeur actuel. Associée à @@, elle renvoie la valeur dans l'adresse de cellule créée par @COORD.

Exemples

@COORD(3;7;25;8) donne l'adresse de cellule relative C:G25.

@@(@COORD(C1;D1;E4;8)) donne la valeur figurant en A:A4 (C1, D1 et E4 contiennent respectivement les valeurs 1, 1 et 4).

Fonction similaire

@CONVREF convertit les lettres de colonne 1-2-3 A à IV (ou les lettres de feuille) en nombres de 1 à 256.

@CORREL

@CORREL(*champ1*;*champ2*) renvoie le coefficient de corrélation entre les valeurs de *champ1* et celles de *champ2*.

Arguments

champ1 et *champ2* sont des noms ou des adresses de champs. Ils doivent être de même taille et de même forme et ne comporter que des valeurs numériques. Dans le cas contraire, **@CORREL** renvoie ERR.

1-2-3 associe deux à deux, de haut en bas, de gauche à droite et de la première feuille à la dernière, les valeurs des deux champs.

Remarques

Le coefficient de corrélation et la covariance caractérisent la relation entre deux ensembles de valeurs.

Contrairement à la covariance, le coefficient de corrélation est indépendant de l'unité de mesure de ces valeurs.

Exemples

@CORREL

Fonctions similaires

@COVAR calcule la covariance de deux ensembles de valeurs.

@COS

@COS(z) calcule le cosinus de l'angle z (dans un triangle rectangle, le cosinus d'un angle est le rapport des longueurs du côté adjacent à cet angle et de l'hypoténuse). Le résultat est compris entre -1 et 1.

Argument

z, angle exprimé en radians, peut être une valeur quelconque, comprise entre $-2^{63}\pi$ et $2^{63}\pi$

Emploi

Dans un triangle rectangle, @COS permet, par exemple, de déterminer la longueur d'un côté, connaissant la longueur de l'hypoténuse et la mesure de l'angle entre celle-ci et le côté adjacent à l'angle.

Exemples

@COS(@DEGENRAD(30)) = 0,866 qui est le cosinus d'un angle de 30 degrés.

Fonctions similaires

@ACOS calcule l'arc cosinus d'une valeur afin de déterminer la valeur d'un angle. @COSH calcule le cosinus hyperbolique d'un angle.

@COSH

@COSH(z) calcule le cosinus hyperbolique d'un angle z. Le résultat de la fonction @COSH est une valeur supérieure ou égale à 1.

Arguments

z, angle exprimé en radians, peut être une valeur quelconque comprise entre -709,7827 et 709,7827 environ.

Exemples

@COSH(@DEGRAD(30)) = 1,140238 qui est le cosinus hyperbolique d'un angle de 30 degrés.

Fonctions similaires

@COS calcule le cosinus d'un angle et @ACOS l'arc cosinus (cosinus inverse).

@COTAN

@COTAN(z) calcule la cotangente d'un angle z. Dans un triangle rectangle, la cotangente d'un angle est le quotient de la longueur du côté adjacent à l'angle par celle du côté opposé.

Arguments

z, exprimé en radians, est la mesure d'un angle compris entre $-2^{63}\pi$ et $2^{63}\pi$.

Exemples

@COTAN(@DEGRAD(30)) = 1,73205 qui est la cotangente d'un angle de 30 degrés.

Fonctions similaires

@ACOTAN calcule l'arc cotangente, @COTANH, la cotangente hyperbolique et @TAN la tangente d'un angle.

@COTANH

@COTANH(z) calcule la cotangente hyperbolique d'un angle z.

Arguments

z, exprimé en radians, est une valeur quelconque mais différente de 0, comprise entre environ -709,7827 et 709,7827.

Exemples

@COTANH(@DEGENRAD(30)) = 2,081283 qui est la cotangente hyperbolique d'un angle de 30 degrés.

Fonctions similaires

@ACOTANH calcule l'arc cotangente hyperbolique, @COTAN, la cotangente et @TANH, la tangente hyperbolique d'un angle.

@COMPTE, @COMPTENUM

@COMPTE(*liste*) compte les cellules occupées d'une *liste* de champs.

@COMPTENUM(*liste*) compte les cellules contenant une valeur numérique, ERR ou ND, d'une *liste* de champs.

Argument

liste se compose d'une combinaison d'adresses ou de noms de champ, séparés par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Remarques

@COMPTE prend en compte toutes les cellules occupées de *liste*, y compris celles comportant un libellé, un préfixe de libellé, ou les valeurs ERR et ND.

@COMPTE et @COMPTENUM s'utilisent fréquemment pour détourner ou interrompre une macro réalisant une tâche sur une série de champs lorsque le pointeur de cellule atteint un champ dépourvu d'entrées.

Exemples

@COMPTE(A2..A3;A5) donne 1 si le champ A2..A3 est vierge, que la cellule A5 soit ou non vierge (car A5 est une adresse de cellule unique).

{SI @COMPTENUM(SEPTEMBRE)=0}{BRANCHE CUMUL} provoque un branchement sur la macro CUMUL (chargée de calculer, par exemple, les totaux annuels), si le champ SEPTEMBRE est vierge ou s'il contient un libellé, un préfixe de libellé ou une formule renvoyant du texte.

Fonctions similaires

@BCOMPTE et @BCOMPTENUM comptent, dans une rubrique de base de données, les cellules occupées répondant aux critères spécifiés.

@COVAR

@COVAR(*champ1*;*champ2*;*[type]*) calcule la covariance des populations ou des échantillons *champ1* et *champ2*.

Arguments

champ1 et *champ2* sont des noms ou des adresses de champs. Ils doivent être de même taille et de même forme et ne comporter que des valeurs numériques. Dans le cas contraire, **@COVAR** renvoie **ERR**.

1-2-3 associe deux à deux, de haut en bas, de gauche à droite et de la première feuille à la dernière, les valeurs des deux champs.

type est un argument facultatif qui spécifie le type de covariance à calculer:

<i>type</i>	1-2-3 calcule
0	La covariance des populations ; valeur par défaut si vous omettez l'argument
1	La covariance des échantillons

Remarques

La covariance est la moyenne des produits des écarts de chaque valeur à la moyenne des valeurs de son échantillon.

Le coefficient de corrélation et la covariance caractérisent la relation entre deux ensembles de valeurs.

Contrairement à la covariance, le coefficient de corrélation est indépendant de l'unité de mesure de ces valeurs.

Exemples

@COVAR

Fonctions similaires

@VAR et **@VARNUM** calculent la variance d'une population, **@VARE** et **@VARENUM** la variance d'un échantillon.

@CORREL calcule le coefficient de corrélation entre les valeurs de deux champs.

@COSEC

@COSEC(z) calcule la cosécante d'un angle z. La fonction cosécante renvoie l'inverse de la fonction sinus. Sa valeur est inférieure ou égale à -1 ou supérieure ou égale à 1.

Arguments

z, exprimé en radians, est un angle dont la mesure est comprise entre $-2^{63} \cdot \pi$ et $2^{63} \cdot \pi$ mais qui ne peut être égale ni à 0.

Exemples

@COSEC(@DEGENRAD(30)) = 2 qui est la cosécante d'un angle de 30 degrés.

Fonctions similaires

@ACOSEC calcule l'arc cosécante, @COSECH, la cosécante hyperbolique et @SIN, le sinus d'un angle.

@COSECH

@COSECH(z) calcule la cosécante hyperbolique d'un angle z. La fonction cosécante hyperbolique renvoie l'inverse de la fonction sinus hyperbolique.

Arguments

z, angle exprimé en radians peut être une valeur quelconque mais différente de 0, comprise entre -709,7827 et 709,7827 environ.

Exemples

@COSECH(@DEGENRAD(30)) = 1,825306 qui est la cosécante hyperbolique d'un angle de 30 degrés.

Fonctions similaires

@ACOSECH calcule l'arc cosécante hyperbolique, @COSEC, la cosécante et @SINH, le sinus hyperbolique d'un angle.

@DUREECOMP

@DUREECOMP(*taux*; *valeur-future*; *valeur-actuelle*) calcule le nombre de périodes nécessaires pour qu'un investissement (*valeur-actuelle*) placé à un *taux* d'intérêt fixe, atteigne la *valeur-future*.

Arguments

taux peut être une valeur supérieure à -1 et différente de 0, ou le nom ou l'adresse d'une cellule contenant cette valeur.

valeur-future et *valeur-actuelle* sont des valeurs quelconques, ou les noms ou adresses des cellules comportant ces valeurs. *valeur-future* et *valeur-actuelle* doivent être de même signe (positif ou négatif).

Emploi

@DUREECOMP permet de déterminer le délai nécessaire pour qu'un investissement donné, placé à un taux d'intérêt fixe, atteigne le montant fixé.

Exemple

Vous venez de déposer 10.000 F sur un compte rémunéré à un taux annuel de 10 % (0,10), calculé mensuellement, et vous voulez connaître le nombre d'années nécessaire pour doubler votre capital.

@DUREECOMP(0,10/12;20000;10000)/12 donne 6,960313. Il vous faudra donc presque 7 ans pour doubler la somme initiale et atteindre 20.000 F.

@DUREECOMP effectuant son calcul sur le nombre total de périodes nécessaires pour atteindre l'unité de durée spécifiée, vous devez inclure cette information dans l'argument *taux* d'intérêt. Ainsi, si le taux d'intérêt annuel de 10 % est calculé mensuellement (comme dans l'exemple ci-dessus), entrez-le sous la forme 0,10/12 (*taux* d'intérêt divisé par le nombre de périodes nécessaires pour atteindre l'unité de temps, l'année).

Fonctions similaires

@DUREE et @DUREEBIS déterminent le temps nécessaire pour qu'un investissement, constitué de versements périodiques égaux, atteigne une valeur donnée.

@@

@@(*emplacement*) renvoie le contenu de la cellule spécifiée dans *emplacement*.

Argument

emplacement est le nom ou l'adresse d'une cellule contenant un nom ou une adresse de cellule, ou une formule générant un nom ou une adresse de cellule. L'argument *emplacement* joue le rôle de pointeur, en désignant une cellule dont la fonction **@@** affiche le contenu. Si *emplacement* ne correspond pas à un nom de champ ou à une adresse de cellule valide ou se réfère à un champ à cellules multiples, **@@** donne ERR.

Remarque

@@ est utile pour le développement de formules conditionnelles du fait que sa référence indirecte peut automatiquement modifier sa propre valeur. Par exemple, la formule

@SI(D2="Y";"D8";@IF(D2="N";"D9";@ERR))

dans la cellule A10, et la formule **@@(A10)** dans la cellule E2, renvoient le contenu de la cellule D8 ou D9, ou ERR, dans E2, selon que D2 contient Y ou N, ou quelque chose d'autre.

Si l'argument *emplacement* se réfère à une cellule contenant une formule, vous devez actualiser la fonction **@@** en appuyant sur F9 (CALCUL) après chaque recalcul automatique. Si vous n'utilisez pas F9, la fonction **@@** donne 0.

Exemple

@@

Fonctions similaires

@CONSH et **@CONSV** permettent d'obtenir le contenu de la cellule spécifiée dans un tableau de consultation horizontale ou verticale. **@CHOIX** donne une valeur ou un libellé extrait d'une *liste* et **@INDEX** affiche le contenu de la cellule située à l'intersection des colonne, ligne et feuille spécifiées.

@J360, @JOURS360

@J360(*date-début*; *date-fin*) calcule le nombre de jours entre deux dates, sur la base d'une année de 360 jours.

@JOURS360(*date-début*; *date-fin*) calcule le nombre de jours entre deux dates, sur la base d'une année de 360 jours et dans le respect des règles boursières américaines.

Arguments

date-début et *date-fin* sont des nombres date.

Remarques

La formule utilisée pour calculer **@JOURS360** se conforme à l'édition 1986 de *Standard Security Calculation Methods*, dans sa mise à jour de 1990.

Typiquement, **@JOURS360** et **@J360** retournent des résultats différents pour des données identiques si la date de début ou la date de fin tombe le dernier jour d'un mois.

Exemples

@JOURS360(**@DATE**(89;4;16);**@DATE**(89;9;25)) = 159

@J360(33290;33524) = 232, nombre de jours entre le 21 février et le 13 octobre 1991, sur la base d'une année de 360 jours.

Fonctions similaires

@DIFDATE calcule le nombre d'années, mois, ou jours entre deux dates. **@JOURS** calcule le nombre de jours entre deux dates, sur la base du calendrier jours/mois spécifié. **@JOUROUV** calcule la date en ajoutant ou retranchant un certain nombre de jours à une date spécifiée, sans tenir compte des week-ends et des jours de congé.

@JOURSOUVRES calcule le nombre de jours entre *date-début* et *date-fin* en excluant les week-ends et les jours de congé. **@MOISDECALER** calcule la date en ajoutant ou retranchant un certain nombre de mois à une date spécifiée.

@DATE

@DATE(*année;mois;jour*) calcule le nombre date correspondant aux arguments *année*, *mois* et *jour* spécifiés.

Arguments

année est un nombre entier compris entre 0 (année 1900) et 199 (année 2099), ou le nom ou l'adresse d'une cellule contenant ce nombre.

mois est un nombre entier compris entre 1 et 12, ou le nom ou l'adresse d'une cellule contenant ce nombre.

jour est un nombre entier compris entre 1 et 31, ou le nom ou l'adresse d'une cellule contenant ce nombre. La valeur de l'argument *jour* doit être compatible avec celle spécifiée pour l'argument *mois*. Ainsi, vous ne pouvez utiliser 31 comme *jour* si vous avez indiqué 4 comme *mois* (avril).

Remarques

Bien que le 29 février 1900 n'ait pas existé (ce n'était pas une année bissextile), 1-2-3 lui assigne un nombre date. Ceci n'a aucune influence sur les calculs qui portent sur des dates postérieures à ce jour, mais vous devez en tenir compte pour tous ceux qui incluent des dates entre le 1er janvier 1900 et le 28 février 1900. Par exemple, pour calculer le nombre de jours du mois de février 1900, vous devrez soustraire 1 au résultat renvoyé par **@DATE**(0;3;1) -**@DATE**(0;2;1) pour corriger 29 en 28.

Si vous voulez faire apparaître le résultat de **@DATE** comme une date actuelle, formatez la cellule qui contient la fonction au moyen d'un des formats de date

Exemples

@DATE(92;2;21) renvoie 33655, ou 21-Fév-92 si la cellule est dotée du format JJ-MMM-AA.

@DATE(91;2;29) renvoie ERR, car 1991 n'était pas une année bissextile.

Fonctions similaires

@DATVAL renvoie le nombre date d'une date entrée sous forme de libellé. **@TEMPS** renvoie le nombre temps correspondant à l'heure spécifiée. **@ACTUEL** renvoie le nombre date et le nombre temps à partir de la date et de l'heure actuelles.

@DIFDATE

@DIFDATE(*date-début*; *date-fin*; *type*) calcule, de façon absolue ou relative, le nombre d'années, mois ou jours entre deux nombres date.

Arguments

date-début et *date-fin* sont deux nombres date.

format est un code du tableau suivant, entré sous forme de texte, qui spécifie le format que doit renvoyer @DATEDIF.

<i>format</i>	Retourne le nombre des
a	Années
m	Mois
j	Jours
mj	Jours, sans tenir compte des mois ni des années
am	Mois, sans tenir compte des années
aj	Jours, sans tenir compte des années

Exemples

Les exemples suivants utilisent le 15 février 1990 comme date de départ et le 15 septembre 1993 comme date de fin.

@DIFDATE(**@DATE**(90;2;15);**@DATE**(93;9;15);"m") retourne 43, nombre de mois entre la date de départ et la date de fin.

@DIFDATE(**@DATE**(90;2;15);**@DATE**(93;9;15);"mj") retourne 0, parce que le jour du mois est le même pour la date de départ et pour la date de fin.

@DIFDATE(**@DATE**(90;2;15);**@DATE**(93;9;15);"am") retourne 7, nombre de mois entre février et septembre.

Fonctions similaires

@J360 et **@JOURS360** calculent le nombre de jours entre deux dates, sur la base d'une année de 360 jours (12 mois de 30 jours). **@JOURS** calcule le nombre de jours entre deux dates, sur la base du calendrier spécifié.

@INFODATE

@INFODATE(*date*;*attribut*) retourne des informations relatives à la date spécifiée sous forme d'un nombre date.

Arguments

date est un nombre date.

attribut est l'un des nombres entiers référencés ci-dessous:

<u><i>attribut</i></u>	<u>Renvoi</u>
1	Jour de la semaine comme libellé, au format court (Lun)
2	Jour de la semaine comme libellé, au format long (Lundi)
3	Jour de la semaine comme entier de 0 (Lundi) à 6 (Dimanche)
4	Semaine de l'année comme entier de 1 à 53
5	Mois de l'année comme libellé, au format court (Jan)
6	Mois de l'année comme libellé, au format long (Janvier)
7	Nombre de jours dans le mois spécifié par <i>date</i>
8	Nombre de jours restant dans le mois spécifié par <i>date</i>
9	Dernier jour du mois spécifié par <i>date</i>
10	La <i>date</i> Trimestre est entrée comme un entier de (T1) à 4 (T4)
11	1 si l'année spécifiée par <i>date</i> est une année bissextile ; 0 dans le cas contraire
12	Jour de l'année spécifié par <i>date</i> , comme nombre de 1 à 366
13	Jours restants dans l'année spécifiés <i>date</i> , sous forme d'un nombre

Exemples

@INFODATE(23063;7) = 28, le nombre de jour du mois de février 1963.

@INFODATE(@DATE(92;10;5);10) = 4, parce que le mois d'octobre est dans le 4ème trimestre.

@DATVAL

@DATVAL(*chaîne*) calcule le nombre date correspondant à la date spécifiée dans l'argument *chaîne*.

Argument

chaîne peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte. *chaîne* doit avoir l'un des formats suivants :

Un format de date apparaissant dans la liste "Utilisation fréquente" ;

Un des cinq formats de date acceptés dans les cellules de format Libre : 31-Déc-96, Déc-96, 31-Déc, et les styles de date longue et courte spécifiés dans les paramètres régionaux (pays) de votre système d'exploitation.

Emploi

Faites appel à **@DATVAL** lorsque vous utilisez des données importées (émanant d'un traitement de texte, par exemple), ou pour convertir en nombre date des dates entrées sous forme de libellés afin de pouvoir les utiliser dans des calculs.

Remarque

Si vous voulez faire apparaître le résultat de **@DATVAL** comme une date et non comme un nombre date, formatez la cellule qui contient la fonction au moyen d'un des formats de date

Exemples

@DATVAL("21-Fév-91") génère le nombre date 33290.

@DATVAL(NAISSDOM) donne le nombre date 18306, si la cellule NAISSDOM contient le libellé 12-Fév-50.

Fonction similaire

@DATE calcule le nombre date correspondant à la date spécifiée. **@DATECHAINE** convertit un nombre date en son équivalent date et l'affiche sous forme d'un libellé.

@JOUR

@JOUR(*nombre-date*) extrait le jour du mois (1 à 31) de l'argument *nombre-date*.

Argument

nombre-date est une valeur comprise entre 1 (1er janvier 1900) et 73050 (31 décembre 2099), ou le nom ou l'adresse d'une cellule contenant cette valeur.

Emploi

Utilisez @JOUR pour extraire le quantième du mois, et non la date dans sa totalité. Cette fonction peut servir d'argument *jour* à une autre fonction de date.

Remarque

Vous pouvez aussi utiliser une autre fonction de date pour générer la valeur de l'argument *nombre-date* de la fonction @JOUR.

Exemples

@JOUR(@ACTUEL) donne le jour du mois.

@JOUR(D9) donne 12, lorsque la cellule D9 contient 33250 (nombre date correspondant au 12-Jan-91).

Fonctions similaires

@MOIS renvoie le numéro du mois, @ANNEE renvoie le numéro de l'année, et @JOURSEM renvoie le numéro du jour de la semaine.

@JOURS

@JOURS(*date-début*; *date-fin*; [*base*]) calcule le nombre de jours entre deux dates sur la *base* du calendrier spécifié.

Arguments

date-début et *date-fin* sont des nombre date. Si *date-début* est antérieure à *date-fin*, le résultat de @JOURS est positif. Si *date-début* est postérieure à *date-fin*, le résultat de @JOURS est négatif. Si *date-début* et *date-fin* sont identiques, le résultat est 0.

base est un argument facultatif qui spécifie le type de calendrier jours/mois à utiliser, par référence au tableau ci-dessous:

<u>base</u>	<u>Base de compte des jours</u>
0	30/360; valeur par défaut si vous omettez l'argument
1	Réel/réel
2	Réel/360
3	Réel/365
4	Européen 30/360

Exemples

@JOURS(@DATE(93;4;16);@DATE(93;9;25)) = 159, nombre de jours entre le 16 avril 1993 et le 25 septembre 1993, sur la base d'un calendrier de 12 mois de 30 jours par an.

@JOURS(@DATE(93;4;16);@DATE(93;9;25);1) = 162, nombre de jours entre les mêmes dates que dans l'exemple précédent, mais sur la base du calendrier réel.

Fonctions similaires

@DIFDATE calcule le nombre d'années, mois ou jours entre deux dates. @J360 et @JOURS360 calculent le nombre de jours entre deux dates, sur la base d'un calendrier de 360 jours par an. @JOURSOUVRES calcule le nombre de jours entre *date-début* et *date-fin* en excluant les week-ends et les jours de congé.

@AMORDEGTC

@AMORDEGTC(*coût; valeur-récupération; durée; période*) calcule la charge d'amortissement, pour une *période* donnée, d'une immobilisation en fonction de son *coût* initial, de la *durée* d'utilisation prévue et de la *valeur-récupération* finale. La méthode utilisée est celle de l'amortissement dégressif à taux constant.

Arguments

coût est le coût initial de l'immobilisation. Il peut être n'importe quel nombre positif ou nul. S'il est égal à 0, le résultat de **@AMORDEGTC** est 0.

valeur-récupération est la valeur estimée de l'immobilisation à la fin de sa durée d'utilisation. Ce peut être n'importe quel nombre positif ou nul. Si elle est supérieure à *coût*, **@AMORDEGTC** renvoie un résultat négatif.

durée correspond au nombre de périodes nécessaires pour atteindre le montant de *valeur-récupération*. Ce nombre doit être supérieur ou égal à 1.

période représente la période pour laquelle vous désirez connaître le montant de l'amortissement. Ce nombre doit être supérieur ou égal à 1, et inférieur ou égal à *durée*.

Les arguments *durée* et *période* doivent être exprimés dans la même unité de temps (généralement l'année).

Remarques

La méthode d'amortissement dégressif à taux constant implique une dépréciation plus lente que celle à taux double. L'amortissement ainsi obtenu est plus marqué dans les dernières années, ce qui autorise des déductions plus élevées à la fin. L'amortissement prend fin lorsque la valeur comptable de l'immobilisation (partie du coût d'acquisition non encore passée en charge d'amortissement) atteint la valeur de récupération.

Exemples

Vous venez d'acquérir une machine de bureau pour un montant de 10.000 F. Sa durée d'utilisation est estimée à huit ans, au bout desquels sa valeur de récupération sera de 1.200 F. Vous souhaitez calculer l'annuité d'amortissement pour la cinquième année, en utilisant la méthode de l'amortissement dégressif à taux constant:

@AMORDEGTC(10000;1200;8;5) = 806,51

Fonctions similaires

@AMDEGVAR calcule l'amortissement selon la méthode de l'amortissement dégressif à taux variable. **@AMORDEG** fait appel à la méthode à taux double. **@AMORLIN** utilise la méthode linéaire, tandis que **@AMORANN** applique un amortissement proportionnel à l'ordre numérique inversé des années.

@AMORDEG

@AMORDEG(*coût; valeur-récupération; durée; période*) calcule, pour une *période* donnée, la charge d'amortissement d'une immobilisation en fonction de son *coût* initial, de la *durée* d'utilisation prévue et de la *valeur-récupération* finale. La méthode utilisée est celle de l'amortissement dégressif à taux double. Cette méthode fiscale américaine consiste à calculer l'amortissement d'un bien en multipliant sa valeur comptable par un taux égal au double de celui de l'amortissement linéaire (taux de l'amortissement linéaire x coefficient de 2). Elle peut néanmoins s'adapter à certaines situations fiscales françaises (amortissement d'immobilisations dont la durée d'utilisation est de 5 ou 6 ans).

Arguments

coût, qui représente le coût initial de l'immobilisation, peut être une valeur quelconque, ou le nom ou l'adresse d'une cellule contenant cette valeur. Il doit être supérieur ou égal à *valeur-récupération*.

valeur-récupération est la valeur estimée de l'immobilisation à la fin de sa durée d'utilisation. Ce peut être une valeur quelconque, ou le nom ou l'adresse d'une cellule comportant cette valeur.

durée correspond au nombre de périodes nécessaires pour atteindre le montant de *valeur-récupération*. *durée* doit être une valeur supérieure à 2, ou le nom ou l'adresse d'une cellule contenant cette valeur.

période représente la période pour laquelle vous cherchez le montant de l'amortissement. *période* doit être une valeur supérieure à 1, ou le nom ou l'adresse d'une cellule comportant cette valeur. Les arguments *durée* et *période* doivent être exprimés dans la même unité de temps (généralement l'année).

Emploi

@AMORDEG fait appel à la méthode de l'amortissement dégressif à taux double. L'amortissement accéléré ainsi obtenu est nettement plus marqué dans les premières années, ce qui autorise des déductions plus élevées au début. L'amortissement prend fin lorsque la valeur comptable de l'immobilisation (partie du coût d'acquisition non encore passée en charge d'amortissement) atteint la valeur de récupération.

Remarques

Lorsque la valeur de récupération est relativement faible, il est possible que l'immobilisation ne soit pas complètement amortie au terme de la durée d'utilisation spécifiée. Dans ce cas, recourez plutôt à **AMDEGVAR**, qui amortit totalement l'immobilisation pendant sa durée d'utilisation.

Exemples

Vous venez d'acquérir un équipement de bureau d'une valeur de 50.000 F. Sa durée d'utilisation est estimée à cinq ans, au bout desquels sa valeur de récupération sera de 5.000 F. Vous souhaitez calculer l'annuité d'amortissement pour la troisième année, en utilisant la méthode de l'amortissement dégressif à taux double. La fonction **@AMORDEG(50000;5200;5;3)** donne 7.200 F.

Fonctions similaires

@AMDEGVAR calcule l'amortissement selon la méthode de l'amortissement dégressif à taux variable.

@AMORDEGTC procède selon la méthode à taux constant. **@AMORLIN** utilise la méthode d'amortissement linéaire.

@AMORANN fait appel à la méthode d'amortissement proportionnel à l'ordre numérique inversé des années.

@DECIMAL

@DECIMAL(*hexa*) convertit un nombre hexadécimal en sa valeur décimale signée.

Arguments

hexa est un nombre hexadécimal compris entre 00000000 et FFFFFFFF, passé sous forme d'une chaîne de caractères. *hexa* peut contenir jusqu'à 8 caractères hexadécimaux parmi les chiffres de 0 à 9 et les lettres de A à F, sans distinction entre majuscules et minuscules.

Remarques

Les nombres hexadécimaux compris entre 00000000 et 7FFFFFFF représentent 0 et des valeurs décimales positives.

Les nombres hexadécimaux compris entre 80000000 et FFFFFFFF représentent des valeurs décimales négatives.

Exemples

@DECIMAL("1A") = 26

@DECIMAL("FFFFFFFE") = -2

Fonction similaire

@HEX convertit un nombre décimal en nombre hexadécimal.

@DEGENRAD

@DEGENRAD(*degrés*) convertit en radians un angle exprimé en *degrés*.

Arguments

degrés est une valeur quelconque.

Exemples

@DEGENRAD(30) = 0,523599 radians

@COS(@DEGENRAD(45)) renvoie 0,707107 qui est le cosinus d'un angle de 45 degrés.

Fonction similaire

@RADENDEG convertit des radians en degrés.

@SOMECART2

@SOMECART2(*liste*) renvoie la somme des carrés des écarts à la moyenne des valeurs de *liste*.

Arguments

liste peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de l'argument *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

@SOMECART2(2;3;9;8;15;2;1) = 159,4286

Fonctions similaires

@STD et @STDNUM calculent l'écart-type d'une série de valeurs.

@FERR

`@FERR(limite-inf ; [limite-sup])` renvoie la fonction d'erreur intégrée entre *limite-inf* et *limite-sup*.

Arguments

limite-inf, valeur quelconque, représente la borne inférieure d'intégration de la fonction @FERR.

limite-sup, valeur quelconque, mais supérieure ou égale à *limite-inf*, représente la borne supérieure d'intégration de la fonction @FERR. Si cet argument est omis, l'intégration de la fonction d'erreur s'effectue entre 0 et *limite-inf*.

Remarques

@FERR renvoie un résultat exact à $\pm 1,2 \cdot 10^{-7}$ près.

Exemples

`@FERR(0,7)` = 0,677801

`@FERR(0,8)` = 0,742101

`@FERR(0,7;0,8)` = 0,0643 (qui est, en fait, la différence des valeurs obtenues dans les deux exemples précédents).

Fonctions similaires

`@FERRC` renvoie la fonction d'erreur complémentaire. `@DERFERR` renvoie la fonction d'erreur dérivée.

@FERRC

@FERRC(x) renvoie la fonction d'erreur complémentaire intégrée entre x et $+\infty$.

Arguments

x est une valeur quelconque.

Remarques

@FERRC(x) est égale à $1 - \text{@FERR}(x)$.

@FERRC renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$ près.

Exemples

@FERRC(0,7) = 0,322199

Fonctions similaires

@FERR renvoie la fonction d'erreur, intégrée entre deux bornes. @DERFERR renvoie la valeur de la fonction d'erreur dérivée.

@DERFERR

@DERFERR(x) renvoie la valeur de la fonction d'erreur dérivée appliquée à la variable x.

Arguments

x peut prendre n'importe quelle valeur.

Exemples

@DERFERR(0,7) = 0,691275

Fonction similaire

@FERR renvoie la fonction d'erreur, intégrée entre deux bornes.

@ERR

@ERR renvoie la valeur ERR.

Emploi

@ERR signale la présence d'erreurs dans les calculs. Elle se rencontre rarement seule. Ainsi, utilisée comme argument de la fonction @SI, @ERR donne ERR sous certaines conditions (lorsqu'une formule génère une valeur non valide, comme un versement mensuel négatif, par exemple).

Remarque

Dans les formules, la valeur ERR et le libellé ERR ne sont pas équivalents. Ainsi, +A2+34 génère ERR si la cellule A2 contient @ERR, mais 34 si elle comporte le libellé ERR.

Exemple

@SI(B14>3;@ERR;B14) donne ERR, si la cellule B14 contient une valeur supérieure à 3.

Fonctions similaires

@ND restitue la valeur ND (non disponible). @ESTERR vérifie la présence de la valeur ERR.

@PAIR

@PAIR(x) arrondit le nombre x au nombre pair supérieur si x est positif, au nombre pair inférieur si x est négatif. Dans le cas où x est un nombre pair (en particulier, 0), la fonction retourne x .

Arguments

x est un nombre quelconque.

Exemples

@PAIR(2,25) = 4

@PAIR(2) = 2

@PAIR(-2,25) = -4

Fonctions similaires

@IMPAIR arrondit un nombre au nombre impair le plus proche. **@ARRONDI**, **@ARRDEF**, et **@ARREXC** arrondissent un nombre aux nombres de décimales spécifiés. **@ARRMUL** arrondit un nombre au multiple spécifié. **@ENT** renvoie la partie entière d'un nombre. **@TRONQUE** tronque un nombre à droite ou à gauche de la virgule.

@EGAL

`@EGAL(chaîne1; chaîne2)` compare deux chaînes de caractères, et donne 1 (vrai) si elles sont identiques, et 0 (faux) dans le cas contraire.

Arguments

chaîne1 et *chaîne2* peuvent être du texte mis entre guillemets, des formules texte, ou les noms ou adresses de cellules contenant des libellés ou des formules texte.

Emploi

Servez-vous de `@EGAL` pour comparer une chaîne de caractères à une entrée de cellule. Dans une macro, vous pouvez ainsi vérifier si la réponse de l'utilisateur est conforme à l'entrée de cellule attendue avant de continuer l'exécution de la macro. C'est donc le moyen idéal de protéger vos macros par des mots de passe.

`@EGAL` permet également de comparer des entrées de cellules existantes, notamment dans les bases de données, pour vérifier si tous les enregistrements d'une rubrique sont identiques.

Remarque

Dans les formules, la fonction `@EGAL` est plus précise que l'opérateur d'égalité (=), dans la mesure où elle distingue les majuscules des minuscules et les caractères accentués de ceux qui ne le sont pas.

Exemples

`@EGAL("ATHENES";"Athènes")` donne 0 (faux).

`@EGAL("Dû";B2)` donne 1 (vrai), si B2 contient le libellé Dû.

`@EGAL("400";400)` donne ERR, car *chaîne2* est une valeur numérique.

@EXP

@EXP(x) calcule la valeur de la constante e (environ 2,718282) élevée à la puissance x.

Arguments

x peut être une valeur comprise entre -709,7827 et 709,7827 environ.

Remarque

Lorsque x est supérieur à 709,7827 ou inférieur à -709,7827 environ, le résultat est trop élevé pour être enregistré par 1-2-3 et la fonction donne ERR.

Exemples

@EXP(0.7) = 2.013753

Fonction similaire

@EXP2 calcule la valeur de la constante élevée à la puissance (x^2). @LN est la fonction inverse de @EXP.

@EXP2

@EXP2(x) calcule la valeur de la constante e (environ 2,718282) élevée à la puissance (x^2).

Arguments

x est une valeur comprise entre -106,52 et 106,52 environ.

Remarques

Si x est en dehors de l'intervalle [-106,56 ; 106,56], 1-2-3 n'a pas la capacité suffisante pour effectuer le calcul. Dans ce cas, @EXP2 renvoie ERR. Si x est en dehors de l'intervalle [-15,102 ; 15,102], la fonction peut être évaluée mais pas affichée (le résultat affiché sera alors une série d'astérisques).

Exemples

@EXP2(0,7) = 0,612626

Fonction similaire

@EXP renvoie l'exponentielle de x.

@FACT

@FACT(n) renvoie la factorielle de n .

Arguments

n est un nombre entier, positif ou nul.

Remarques

La factorielle de n est égale au produit de tous les nombres entiers compris entre 1 et n .

Si n est supérieur ou égal à 171, 1-2-3 n'a pas la capacité suffisante pour effectuer le calcul. Dans ce cas, @FACT renvoie ERR.

Exemples

@FACT(0) = 1

@FACT(5) = 120, soit: $1*2*3*4*5$.

Fonctions similaires

@LNFACT renvoie le logarithme népérien de la factorielle de n . @PRODUIT multiplie les valeurs d'une liste.

@LNFACT

@LNFACT(n) renvoie le logarithme népérien de la factorielle de n .

Arguments

n est un nombre entier, positif ou nul.

Remarques

La factorielle de n est égale au produit de tous les nombres entiers compris entre 1 et n .

Par convention mathématique, la factorielle de 0 est égale à 1. Donc @LNFACT(0) = ln(0) = 0.

Exemples

@LNFACT(5) = 4,787492 qui est le résultat de $\ln(5!) = \ln(1 \cdot 2 \cdot 3 \cdot 4 \cdot 5)$.

Fonctions similaires

@FACT renvoie la factorielle de n . @LN renvoie le logarithme népérien d'un nombre.

@FAUX

@FAUX génère la valeur logique 0 (faux).

Emploi

@FAUX s'emploie dans les macros ou les fonctions de type @SI ou @CHOIX, qui utilisent la valeur logique 0 (faux). @FAUX correspond alors au 3ème argument de la fonction @SI (argument indiquant que la condition n'est pas remplie).

Remarques

Lorsqu'une instruction logique de type $A1=B1$ est vraie, sa valeur logique est 1. Si elle est fausse, sa valeur logique est 0. Vous pouvez entrer aussi bien la fonction @FAUX que la valeur 0 dans des formules testant des conditions logiques, mais @FAUX a l'avantage de rendre la formule plus lisible.

Exemple

@SI(A6>500;@VRAI;@FAUX) donne 0 si A6 contient une valeur inférieure ou égale à 500.

Fonction similaire

@VRAI restitue la valeur logique 1.

@DISTF

@DISTF(*x*; *degré-liberté1*; *degré-liberté2*; [*type*]) renvoie la fonction de distribution de *F* appliquée à *x*.

Arguments

x est la variable passée à la fonction @DISTF. Cette variable est interprétée différemment selon la valeur donnée à *type*:

<u>Si <i>type</i> vaut</u>	<u><i>x</i> représente</u>
0 (défaut)	la valeur critique (ou valeur de seuil) de la fonction de distribution de <i>F</i> . <i>x</i> doit alors être supérieur ou égal à 0.
1	une probabilité. <i>x</i> doit alors être compris entre 0 et 1.

degré-liberté1 et *degré-liberté2*, entiers positifs, représentent respectivement les degrés de liberté des premier et second échantillons.

type est un argument facultatif qui détermine le type du résultat renvoyé par la fonction @DISTF.

<u><i>type</i></u>	<u>1-2-3 calcule</u>
0	La probabilité de risque correspondant à la valeur de seuil <i>x</i> ; valeur par défaut si vous omettez l'argument
1	La valeur de seuil correspondant à la probabilité de risque <i>x</i>

Remarques

@DISTF renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$. Si après 100 itérations, 1-2-3 ne peut pas fournir un résultat exact à 0,0000001 près, @DISTF renvoie ERR.

La distribution de *F* est une fonction de distribution continue obtenue par le rapport de deux distributions de Chi-carrée (χ^2), chacune divisée par son degré de liberté.

Utilisez @DISTF pour déterminer le degré de variation de deux échantillons.

Exemples

@DISTF(3,07;8;10) = 0,05

@DISTF(0,05;8;10) = 0.999865

Fonctions similaires

@TESTF renvoie la probabilité associée à un test *F*. @DISTCHI évalue la distribution de Chi-carrée (χ^2) et @DISTT, la distribution de Student (distribution de *t*).

@CHERCHE

@CHERCHE(*chaîne1*;*chaîne2*;*n*) recherche la position de la première occurrence de *chaîne1* dans *chaîne2*, à partir du *n*ème caractère.

Arguments

chaîne1 et *chaîne2* peuvent être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant du texte ou une formule texte.

n, numéro de position, peut être un nombre entier positif ou nul, ou le nom ou l'adresse d'une cellule contenant ce type de valeur.

Emploi

@CHERCHE permet de déterminer la position d'un caractère donné ou d'un groupe de caractères dans une chaîne. Elle s'utilise aussi dans les macros dont l'exécution dépend de la localisation d'une série spécifique de caractères. Associée à @MILIEU ou à @SUBSTITUE, @CHERCHE localise, extrait ou remplace une chaîne.

Remarques

La valeur ERR s'affiche lorsque @CHERCHE ne trouve pas *chaîne1* dans *chaîne2*, ou quand *n* dépasse le nombre de caractères de *chaîne2* ou est une valeur négative.

@CHERCHE distingue les majuscules des minuscules, et les caractères accentués de ceux qui ne le sont pas. Ainsi, @CHERCHE ne trouve pas l'argument *chaîne1* "e" dans *chaîne2* "TOULOUSE."

Exemple

@CHERCHE("F";"Comptes Fournisseurs";0) donne 8, car F (*chaîne1*) se trouve en 8ème position dans la chaîne Comptes Fournisseurs (*chaîne2*).

@VALFUT et @VALFUTBIS

@VALFUT(*versements*; *taux*; *périodes*) calcule la valeur future d'un investissement, sur la base de *versements* égaux, placés à un *taux* d'intérêt périodique, sur un nombre de *périodes* donné.

@VALFUTBIS(*versements*; *taux*; *périodes*; [*type*]; [*valeur-actuelle*]) calcule la valeur future d'un investissement dont vous pouvez spécifier la *valeur-actuelle*, sur la base de *versements* égaux, placés à un *taux* d'intérêt périodique, sur un nombre de *périodes* donné. Selon la valeur spécifiée pour *type*, **@VALFUTBIS** calcule la valeur future pour des versements effectués soit en fin, soit en début de période.

Arguments

versements et *périodes* sont des nombres.

Pour **@VALFUTBIS**, *périodes* doit être positif.

taux est un nombre supérieur à -1.

type est un argument facultatif qui permet de préciser si les versements ont lieu en fin ou début de période:

<u>type</u>	<u>les versements interviennent</u>
0	A la fin de chaque période ; valeur par défaut si vous omettez l'argument
1	Au début de chaque période

valeur-actuelle est un argument facultatif qui permet de spécifier la valeur actuelle de la série de versements futurs. *valeur-actuelle* peut être un nombre quelconque. Si vous l'omettez, 1-2-3 utilise 0.

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarques

Les arguments utilisés doivent faire référence à la même unité de temps. Par exemple, si les versements sont mensuels, vous devrez rapporter au mois un taux d'intérêt annuel (généralement en le divisant par 12), et exprimer les périodes en mois en multipliant par 12 un nombre d'années.

Exemples

Vous projetez de déposer 10.000 F par an sur un compte d'épargne retraite pendant les 20 années à venir. Le compte est rémunéré en fin d'année, à un taux annuel de 7,5%. Pour calculer la valeur de votre compte dans 20 ans, si vous effectuez vos versements en fin de période (le dernier jour de chaque année), entrez la formule suivante:

@VALFUT(10000;0,075;20)=433046,81 F

Si vous effectuez vos versements le premier jour de chaque année, entrez la formule suivante:

@VALFUTBIS(10000;0,075;20;1)=465525,32 F

Fonctions similaires

@VALFUT2 calcule la valeur future d'un investissement, sur la base de *versements* égaux, placés à un *taux* d'intérêt périodique, sur un nombre de *périodes* donné, dans le cadre d'une convention d'annuités. **@VALFUTMONTANT** renvoie la valeur future d'une somme forfaitaire placée à un taux donné pour un certain nombre de périodes.

@VALACT et **@VALACTBIS** calculent la valeur actuelle d'un investissement, **@VAN** calcule sa valeur actuelle nette.

Exemple de fonction @CONSH

Soit EXPED, un tableau de consultation horizontale figurant en A2..E7 et présentant le montant des frais d'expédition d'un colis dans diverses villes. La fonction @CONSH("Francfort";EXPED;3), entrée dans une cellule formatée en monnaie 2 décimales, renvoie 24,00 F, coût d'expédition d'un colis de type 3 à Francfort (1-2-3 recherche le libellé Francfort dans la première ligne du tableau, puis se déplace de 3 lignes vers le bas dans la colonne correspondante, conformément à l'argument position-ligne).

A	----	A	-----	B	-----	C	-----	D	-----	E	----
1			-----	COLIS		DESTINATION	-----				
2	Type de colis	Londres		Madrid		Francfort		Lyon			
3	1	18,36 F		19,33 F		20,12 F		9,29 F			
4	2	20,32 F		21,66 F		22,03 F		11,25 F			
5	3	22,44 F		23,88 F		24,00 F		13,25 F			
6	4	24,14 F		25,26 F		25,75 F		16,85 F			
7	5	28,32 F		29,00 F		29,80 F		19,54 F			

Exemple de fonction @INDEX

Soit le tableau nommé AUGMENTATION (A3..D7), récapitulant les augmentations de salaire accordées aux employés en fonction de leur taux de productivité.

@INDEX(AUGMENTATION;2;3), entré dans une cellule dotée du format % à 0 décimales, renvoie 5%, soit une augmentation accordée à un salarié de catégorie 2 (numéro de position de colonne 2 dans le champ A3..D7) ayant un taux de productivité de 3 (numéro de position de ligne 3).

@INDEX(AUGMENTATION;1;2), renvoie 7%, soit une augmentation accordée à un salarié de catégorie 1 (numéro de position de colonne 1 dans le champ A3..D7) ayant un taux de productivité de 2 (numéro de position de ligne 2).

A ----- A ----- B --- C --- D --- E -----

1

2 ---- AUGMENTATION ---

3 cat1 cat2 cat3 cat4

4 prod1 10% 9% 8% 7%

5 prod2 7% 6% 5% 4%

6 prod3 6% 5% 4% 3%

7 prod4 3% 2% 1% 0%

@GAMMA

@GAMMA(x) renvoie la fonction Gamma, appliquée à x.

Arguments

x est une valeur quelconque, hormis 0 et les entiers négatifs.

Remarques

Le résultat de la fonction @GAMMA est exact sur, au moins, les 6 premiers chiffres.

Exemples

@GAMMA(0,5) = 1,772454

@GAMMA(5) = 24

Fonctions similaires

@BETA renvoie la fonction Béta. @GAMMAI renvoie la fonction Gamma incomplète. @LNGAMMA renvoie le logarithme népérien de la fonction Gamma.

@GAMMAI

@GAMMAI($a;x$;*[complément]*) renvoie la fonction Gamma incomplète.

Arguments

a est une valeur strictement positive.

x est une valeur positive ou nulle.

complément est un argument facultatif qui détermine le type du résultat renvoyé par la fonction @GAMMAI:

<u>complément</u>	<u>1-2-3 renvoie</u>
0	$P(a;x)$; valeur par défaut si vous omettez l'argument
1	$Q(a;x)$ qui est égal à $1-P(a;x)$

Remarques

Le résultat de la fonction @GAMMAI est exact sur, au moins, les 6 premiers chiffres.

Exemples

@GAMMAI(7,5;12,497;1) = 0,050024

Fonctions similaires

@GAMMA renvoie la fonction Gamma. @LNGAMMA renvoie le logarithme népérien de la fonction Gamma.

@LNGAMMA

@LNGAMMA(x) renvoie le logarithme népérien de la fonction Gamma appliquée à x.

Arguments

x est une valeur quelconque, strictement positive.

Remarques

Le résultat de la fonction @LNGAMMA est exact sur, au moins, les 6 premiers chiffres.

Exemples

@LNGAMMA(0,5) = 0,572365

@LNGAMMA(5) = 3,178054

Fonctions similaires

@GAMMA renvoie la fonction Gamma et @GAMMAI la fonction Gamma incomplète.

@MOYGEOM

@MOYGEOM(*liste*) calcule la moyenne géométrique des nombres de *liste*.

Arguments

liste peut contenir une combinaison quelconque des arguments suivants : nombres, formules numériques, adresses ou noms de champ contenant des nombres ou des formules numériques. Ces arguments devant être séparés par le séparateur d'arguments.

Tous les nombres de *liste* doivent être supérieurs à 0.

Voir aussi Arguments des fonctions @ statistiques.

Remarques

La moyenne géométrique d'une *liste* de *n* nombres est la racine *nième* du produit de ces nombres.

La moyenne géométrique calculée par **@MOYGEOM** est inférieure à la moyenne arithmétique calculée par **@MOYENNE** sauf si tous les nombres de la *liste* sont égaux. Dans ce dernier cas, la moyenne géométrique est égale à la moyenne arithmétique.

Exemples

@MOYGEOM(A1..A4) = 239,1886, si A1..A4 contient les nombres 160, 227, 397, et 227.

Fonctions similaires

@MOYHARM calcule la moyenne harmonique des nombres d'une liste. **@MOYENNE** et **@MOYENNENUM** calculent la moyenne arithmétique des nombres d'une liste. **@MEDIANE** calcule la médiane des nombres d'une liste.

@MOYHARM

@MOYHARM(*liste*) calcule la moyenne harmonique des nombres contenus dans *liste*.

Arguments

liste peut contenir une combinaison quelconque des arguments suivants : nombres, formules numériques, adresses ou noms de champ contenant des nombres ou des formules numériques. Ces arguments devant être séparés par le séparateur d'arguments.

Tous les nombres de *liste* doivent être supérieurs à 0.

Voir aussi Arguments des fonctions @ statistiques.

Remarques

La moyenne harmonique des nombres de *liste* est l'inverse de la moyenne arithmétique des inverses des nombres de *liste*. Par exemple, la moyenne arithmétique de 1/2, 1/3 et 1/4 est 13/36; en conséquence, la moyenne harmonique de 2, 3 et 4 est 36/13.

Pour une même *liste*, la moyenne harmonique est toujours inférieure à la moyenne géométrique.

Exemples

@MOYHARM(25;50;75) = 40,90909

Fonctions similaires

@MOYGEOM calcule la moyenne géométrique des nombres d'une liste. @MOYENNE et @MOYENNENUM calculent la moyenne arithmétique des nombres d'une liste. @MEDIANE calcule la médiane d'une liste de nombres.

@HEX

@HEX(x) convertit un nombre décimal en son équivalent hexadécimal.

Arguments

x est un entier entre -2.147.483.648 et 2.147.483.647. Si x n'est pas entier, 1-2-3 le tronque à sa partie entière.

Remarques

Les nombres hexadécimaux 00000000 à 7FFFFFFF correspondent à 0 et à des nombres décimaux positifs.

Les nombres hexadécimaux de 80000000 à FFFFFFFF correspondent à des nombres décimaux négatifs.

Exemples

@HEX(162) = A2

Fonction similaire

@DECIMAL convertit des nombres hexadécimaux en nombres décimaux.

@CONSH

@CONSH(*x;champ;position-ligne*) recherche *x* dans la première ligne de *champ*, s'arrête sur la colonne contenant soit le nombre le plus proche de *x* (égal ou inférieur), soit le libellé *x*, descend de *position-ligne* dans *champ* et retourne le contenu de la cellule atteinte. Dans le cas où *x* est un nombre, la recherche n'a de sens que si les nombres de la première ligne de *champ* sont classés par ordre croissant.

Arguments

x peut être un nombre ou une chaîne.

<u>Première ligne</u>	<u>valeur de x</u>
Valeurs	valeur supérieure ou égale à la première valeur de <i>champ</i> . Si <i>x</i> est inférieur à la première valeur de <i>champ</i> , @CONSH renvoie ERR. Si <i>x</i> est supérieur à la dernière valeur de la première ligne de <i>champ</i> , @CONSH s'arrête en dernière colonne et renvoie le contenu de la cellule repérée par <i>position-ligne</i> dans cette colonne
Libellés	Texte entre " " (guillemets), une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte. <i>x</i> doit reproduire exactement le contenu d'une des cellules de la première ligne du <i>champ</i> , faute de quoi @CONSH renvoie ERR

champ, qui correspond au champ où figure le tableau de consultation horizontale, peut être un nom ou une adresse de champ. Si *champ* est multifeuille, 1-2-3 ne tient compte que de la première feuille.

position-ligne représente le numéro de position de la ligne dans *champ*. Il doit être inférieur au nombre de lignes de *champ*.

Emploi

@CONSH permet de choisir les éléments d'un tableau, ou d'automatiser la sélection de données destinées à des formules ou des macros (application fournissant des prix extraits d'une liste de prix, par exemple).

@CONSH facilite la localisation de données dans des tableaux proposant des choix dépendant de variables spécifiques (tels les tableaux d'imposition ou de commissions).

Exemple

@CONSH

Fonctions similaires

@CONSV retourne le contenu d'une cellule dans un tableau de consultation verticale. **@INDEX** retourne le contenu de la cellule située à l'intersection d'une colonne et d'une ligne identifiées par leur numéro de position dans le champ.

@CHOIX est utilisée en remplacement d'un tableau de consultation d'une seule ligne. **@CORRESP** indique la position relative d'une cellule identifiée par son contenu. **@XINDEX** restitue le contenu de la cellule située à l'intersection d'une colonne et d'une ligne identifiées par leur contenu. **@CONSMAX** renvoie la référence absolue de la cellule qui contient la valeur la plus élevée de liste-champs. **@CONSMIN** retourne la référence absolue de la cellule qui contient la valeur la plus basse de liste-champs.

@HEURE

@HEURE(*nombre-temps*) extrait l'heure d'un *nombre-temps*, sous la forme d'un nombre compris entre 0 et 23 (23:00 ou 11:00 PM).

Argument

nombre-temps est une valeur comprise entre 0,000000 (minuit) et 0,999988 (23:59:59 ou 11:59:59 PM), ou le nom ou l'adresse d'une cellule contenant cette valeur. L'argument *nombre-temps* est généralement généré par une autre fonction @.

Emploi

@HEURE extrait l'heure des valeurs temps produites par les fonctions @ACTUEL, @TEMPS et @HEURVAL, ce qui permet d'effectuer des calculs portant sur des heures entières (salaires horaires, nombre d'heures écoulées depuis le début d'un projet, horodatage d'une feuille, etc.).

Exemples

@HEURE(0,51565) donne 12 (midi), car 0,51565 est le nombre temps correspondant à 12:22:32.

@HEURE(@TEMPS(13;45;18)) donne 13, car 13 est l'argument *heure* de la fonction @TEMPS(13;45;18).

Fonctions similaires

@MINUTE et @SECONDE extraient respectivement les minutes et les secondes d'un nombre temps.

@SI

@SI(*condition*;x;y) teste l'argument *condition* et, selon le résultat obtenu, génère *x* (si *condition* est vérifiée) ou *y* (si elle ne l'est pas).

Arguments

condition est généralement une formule logique, ou le nom ou l'adresse d'une cellule contenant cette formule. Les formules, les nombres, le texte mis entre guillemets, et les noms ou adresses de cellule peuvent aussi servir de *condition*. Toute *condition* est considérée comme vraie si elle a pour résultat un nombre différent de 0, et comme fausse si ce résultat est égal à 0. Par exemple, si la condition renvoie à une cellule vierge, ou à une cellule contenant du texte, elle est considérée comme fausse et **@SI** retourne *y*. Si la condition résulte en la valeur ERR ou la valeur ND, ou renvoie à une cellule contenant ce type de valeur, **@SI** retourne respectivement ERR ou ND.

x et *y* peuvent être des valeurs, du texte mis entre guillemets ou les noms ou adresses de cellule contenant ce type d'élément.

Emploi

@SI peut s'utiliser en combinaison avec @ERR ou @ND pour repérer des erreurs ou des données manquantes dans des formules. **@SI** est également utile pour prévenir les résultats ERR ou ND lorsqu'une formule fait référence à une cellule dont le contenu est susceptible d'être inutilisable (notamment, pour interdire l'erreur provoquée par une division par 0).

Remarque

Vous pouvez imbriquer plusieurs fonctions **@SI** de façon à créer une condition complexe. Ainsi la formule :

@SI(TOT>100000;TOT*0,15;**@SI**(TOT>50000;TOT*0,10;TOT*0,02)) imbrique deux fonctions **@SI** pour déterminer des pourcentages de commissions en fonction de trois critères de ventes différents : ventes supérieures à 100.000 F, supérieures à 50.000 F, ou bien inférieures ou égales à 50.000 F.

Exemple

@SI(SOLDE>=0;SOLDE;"Découvert") donne la valeur figurant dans SOLDE lorsqu'elle est positive ou nulle, et le libellé Découvert lorsqu'elle est négative.

@INDEX

@INDEX(*champ;colonne;ligne;[feuille]*) restitue le contenu de la cellule située à l'intersection de la *colonne*, de la *ligne* et (facultativement) de la *feuille d'un champ*.

Arguments

champ est un nom ou une adresse de champ.

colonne est le numéro de position à l'intérieur de *champ* de la colonne utilisée par @INDEX, ou le nom ou l'adresse d'une cellule contenant un entier positif ou nul.

ligne est le numéro de position de la ligne utilisée par @INDEX, ou le nom ou l'adresse d'une cellule contenant un entier positif ou nul.

feuille, argument facultatif, est le numéro de position de la feuille utilisée par @INDEX, ou le nom ou l'adresse d'une cellule contenant un entier positif ou nul. Si vous ne spécifiez pas d'argument *feuille*, @INDEX utilise la première feuille de *champ*.

Emploi

Utilisez @INDEX lorsque vous souhaitez rechercher le contenu d'une cellule dans un tableau de consultation en vous servant des positions relatives des lignes, des colonnes (et éventuellement des feuilles), et non en fonction de valeurs spécifiques.

Exemple

@INDEX

Fonctions similaires

@CONSH et @CONSV localisent des entrées de cellules dans des tableaux de consultation horizontale et verticale.

@CHOIX est utilisée en remplacement d'un tableau de consultation d'une seule ligne.

@CORRESP indique la position relative d'une cellule identifiée par son contenu. @XINDEX restitue le contenu de la cellule située à l'intersection d'une colonne et d'une ligne identifiées par leur contenu.

@CONSMAX renvoie la référence absolue de la cellule qui contient la valeur la plus élevée de liste-champs.

@CONSMIN retourne la référence absolue de la cellule qui contient la valeur la plus basse de liste-champs.

@INFO

@INFO(*attribut*) fournit des informations sur la session 1-2-3 actuelle.

Arguments

attribut peut être l'un des éléments répertoriés ci-dessous, entré sous forme de texte entre guillemets , ou le nom ou l'adresse d'une cellule contenant un tel élément.

<u>Attribut</u>	<u>Résultat</u>
Auteur	Le nom d'utilisateur de la première personne à avoir enregistré le classeur actuel.
codecommande	ND
codecommandeb d	ND
date-création	Un <u>nombre date</u> qui correspond à la date du premier enregistrement du classeur actuel.
datederrévision	Un nombre date qui correspond à la date du dernier enregistrement du classeur actuel.
derrévisonpar	Le nom d'utilisateur de la dernière personne à avoir enregistré le classeur actuel.
écran-hauteur	Hauteur de l'écran, exprimée en pixels.
écran-largeur	Largeur de l'écran, exprimée en pixels.
messagepilotebd	Dernier message généré par DataLens.
macro- pas_à_pas	Oui lorsque le mode pas à pas est activé pour les macros et Non, s'il ne l'est pas.
macro-suivi	Oui lorsque le mode suivi est activé pour les macros et Non, s'il ne l'est pas.
mémdispo	Mémoire disponible.
mémtot	Espace mémoire disponible total (mémoire disponible et mémoire actuellement utilisée).
mode	Mode actuel: 0 Attente 1 Pret 3 Menu 4 Valeur 5 Pointe 6 Modifie 7 Erreur 8 Trouve 9 Fichier 10 Aide 11 Stat. 13 Noms 99 Tous les autres modes (modes définis par l'utilisateur via <u>{INDIQUE}</u> , par exemple).
nbenregistbd	ND

nbfeuille	Nombre de feuilles dans le classeur actuel.
nbfich	Nombre de classeurs ouverts.
origin	Adresse absolue de la cellule supérieure gauche dans la feuille actuelle.
personnalisation-nom_utilisateur	Renvoie votre nom utilisateur (messagerie ou réseau).
recalcul	Mode de recalcul actuel, soit les chaînes "automatique" ou "manuel".
répertoire	<u>Voie d'accès</u> actuelle, lettre d'unité comprise.
sélection	L'adresse du champ, ou nom du graphique, de l'objet dessiné ou de la table de base de données actuellement sélectionné.
sélection-partie	ND
sélection-type	Type de la sélection : champ, graphique, dessin ou interrogation.
systempl	Nom du système d'exploitation.
taille-classeur	La taille du classeur actuel en kilo-octets (KO).
temps-édition	Un <u>nombre temps</u> qui correspond au nombre total d'heures et de minutes d'ouverture du classeur actuel.
version	Numéro de la version de 1-2-3 pour Windows utilisée. Ce numéro comprend le numéro de la version de base, le niveau de mise à jour et le numéro de version.
versionse	ND
windir	Chemin d'accès au répertoire Windows, unité comprise.

Remarques

Outre les attributs mentionnés pour la fonction, vous pouvez spécifier pour l'argument *attribut* l'un des attributs 1-2-3 utilisés dans les macros.

Recalculez la feuille avant d'utiliser @INFO (en appuyant sur F9 (CALC)) afin d'être sûr d'obtenir des résultats corrects.

Utilisée dans une macro, @INFO informe l'utilisateur ou la macro sur l'état de 1-2-3. Elle peut servir, par exemple, à indiquer la voie d'accès actuelle d'une macro de sauvegarde automatique, ou à signaler qu'il reste peu d'espace mémoire.

Associez @INFO à @SI pour vérifier l'état de 1-2-3 et indiquer à une macro la tâche à exécuter dans certaines conditions : changement de voie d'accès, suppression de données inutiles, fermeture de certains des classeurs actifs en cas de mémoire insuffisante, etc.

Exemples

@INFO("nbfich") donne 2, lorsque deux classeurs sont actifs.

@INFO(B4) donne 3, lorsque B4 contient l'attribut mode et que 1-2-3 est en mode MENU.

Similar @fonctions

@CELLULE envoie des information sur la première cellule du champ.

@POINTCELL renvoie des information sur la cellule actuelle.

@ENT

@ENT(x) retourne la partie entière de x.

Argument

x est un nombre, ou le nom ou l'adresse d'une cellule contenant un nombre. Si x est un libellé, la fonction retourne 0.

Emploi

@ENT s'utilise dans les calculs portant sur des nombres entiers.

Remarque

@ENT extrait la partie entière d'un nombre et élimine sa partie décimale. Lorsque vous souhaitez que les valeurs soient affichées sous forme de nombres entiers dans la feuille, mais que les décimales soient prises en considération dans les calculs, attribuez aux cellules concernées format fixe.

Exemples

@ENT(35,67) donne 35.

@ENT(@ACTUEL) donne le nombre date correspondant à la date actuelle, et élimine le nombre temps (car celui-ci est représenté par la partie décimale de @ACTUEL).

Fonction similaire

@ARRONDI, @ARREXC et @ARRDEF arrondissent une valeur au multiple le plus proche de la valeur de 10 spécifiée. @ARRMULI arrondit une valeur au multiple indiqué. @PAIR et @IMPAIR arrondissent une valeur respectivement au nombre pair et au nombre impaire le plus proche. @TRONQUE tronque une valeur au nombre de décimales spécifié.

@INTVER, @PRINCVER

@INTVER(*principal*; *taux*; *périodes*; *période-début*; [*période-fin*]; [*type*]; [*valeur-future*]) calcule la part des intérêts correspondant au remboursement d'un emprunt (*principal*) au *taux* d'intérêt spécifié et sur le nombre de *périodes* donné.

@PRINCVER(*principal*; *taux*; *périodes*; *période-début*; [*période-fin*]; [*type*]; [*valeur-future*]) calcule la part du principal correspondant au remboursement d'un emprunt (*principal*) au *taux* d'intérêt spécifié et sur le nombre de *périodes* donné.

Arguments

principal et *périodes* sont des nombres. *périodes* peut être quelconque, à l'exception de 0.

taux est un nombre supérieur à -1.

période-début est le numéro de la période à partir de laquelle vous voulez que commence le calcul des intérêts ou du principal. Ce numéro doit être un nombre supérieur ou égal à 1, et inférieur ou égal au nombre de *périodes*.

période-fin est un argument facultatif qui permet de spécifier le numéro de la période sur laquelle vous voulez que s'arrête le calcul des intérêts ou du principal. Ce numéro doit être un nombre supérieur à *période-début*. Si vous omettez cet argument, 1-2-3 utilise *période-début*.

type est un argument facultatif qui permet de préciser si les versements ont lieu en fin ou en début de période.

<u><i>type</i></u>	<u>les versements interviennent</u>
0	A la fin de chaque période ; valeur par défaut si vous omettez l'argument
1	Au début de chaque période

valeur-future est un argument facultatif qui permet de spécifier la valeur future d'une série de versements. *valeur-future* peut être un nombre quelconque. Si vous l'omettez, 1-2-3 utilise 0.

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarques

Les arguments utilisés doivent faire référence à la même unité de temps. Par exemple, si les versements sont mensuels, vous devrez rapporter au mois un taux d'intérêt annuel (généralement en le divisant par 12), et exprimer les périodes en mois en multipliant par 12 un nombre d'années.

Exemples

Vous avez emprunté 8.000 F à rembourser sur 3 ans à un taux d'intérêt annuel de 10,5%. Le montant des versements mensuels est de 260,02 F. Pour déterminer les intérêts payés au cours de la dernière année de remboursement:

@INTVER(8000;0,105/12;36;25;36) = 170,45

Pour déterminer le principal correspondant à cette même année:

@PRINCVER(8000;0,105/12;36;25;36) = 2.949,79

Fonction similaire

@REMBOUR calcule le montant du remboursement périodique d'un emprunt.

@TAUXBIS

@TAUXBIS(*périodes*; *versements*; *valeur-actuelle*; [*type*]; [*valeur-future*]; [*estimation*]) calcule le taux d'intérêt périodique nécessaire pour qu'un investissement (*valeur-actuelle*) effectué en *versements* égaux atteigne une *valeur-future* sur un nombre de *périodes* donné.

Arguments

périodes est un nombre positif entier.

versements et *valeur-actuelle* sont des nombres.

type est un argument facultatif qui permet de préciser si les versements ont lieu en fin ou en début de période :

<i>type</i>	les versements interviennent
0 (défaut)	A la fin de chaque période
1	Au début de chaque période

valeur-future est un argument facultatif qui permet de spécifier la valeur future de la série des versements. Vous pouvez spécifier une valeur quelconque. Si vous omettez cet argument, 1-2-3 utilise 0.

estimation est un argument facultatif qui permet de proposer une première estimation du taux à calculer. C'est un nombre 0 et 1. Si vous omettez cet argument, 1-2-3 utilise 0,10 (10%).

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarques

Pour calculer le taux d'intérêt, @TAUXBIS effectue une série d'approximations à partir de l'*estimation* proposée. Comme plusieurs solutions peuvent être possibles, essayez une autre *estimation* si le résultat est inférieur à 0 ou supérieur à 1.

Si @TAUXBIS ne parvient pas à fournir un résultat approché à 0,0000001 près après 30 itérations, le résultat est **ERR**. Si vos estimations successives continuent à retourner ERR, utilisez @VAN pour les contrôler. Si @VAN renvoie un nombre positif, votre estimation est trop faible; si @VAN renvoie un nombre négatif, elle est trop forte. Votre estimation est correcte si @VAN renvoie 0.

Les arguments utilisés doivent faire référence à la même unité de temps. Par exemple, si les versements sont mensuels, vous devrez rapporter au mois un taux d'intérêt annuel (généralement en le divisant par 12), et exprimer les périodes en mois en multipliant par 12 un nombre d'années.

Exemples

Vous avez déposé 6.000 F sur un compte et vous voulez pouvoir retirer 100 F par mois pendant 8 ans. Pour calculer le taux d'intérêt nécessaire, utilisez la fonction ci-dessous :

@TAUXBIS(96;100;6000;0;0;0,01) = 0,010623, soit 1,06% d'intérêt mensuel.

Fonctions similaires

@VAN calcule la valeur actuelle nette d'une série de rentrées de fonds futures. @VALACT et @VALACTBIS calculent la valeur actuelle d'un investissement sur la base d'une série de versements égaux. @VALFUT et @VALFUTBIS calculent la valeur future d'un investissement. @TAUX calcule le taux d'intérêt périodique nécessaire pour qu'un investissement atteigne une valeur capitalisée.

@AMORT

@AMORT(*estimation;champ*) calcule le taux de rentabilité interne d'une série de mouvements de trésorerie générés par un investissement. Le taux de rentabilité interne est le pourcentage pour lequel la valeur actualisée d'une série de rentrées de fonds futures est égale à l'investissement initial.

Arguments

estimation est une valeur décimale ou un pourcentage représentant le taux de rentabilité prévu, ou le nom ou l'adresse d'une cellule contenant ce type d'élément. *estimation* est généralement un pourcentage compris entre 0 (0%) et 1 (100%). Lorsque les calculs portent sur des fonds importants, la valeur d'*estimation* doit être la plus précise possible.

champ est le nom ou l'adresse du champ contenant les mouvements de trésorerie (sorties et rentrées de fonds). Les nombres négatifs sont considérés comme des sorties de fonds et les nombres positifs, comme des rentrées de fonds. La première valeur du champ doit être un nombre négatif (sortie de fonds), car il représente l'investissement initial. 1-2-3 affecte la valeur 0 aux cellules vierges et aux libellés figurant dans *champ* et les intègre dans ses calculs.

Emploi

@AMORT détermine la rentabilité d'un investissement. Combinée à d'autres fonctions financières, telle **@VAN**, elle permet d'évaluer un investissement.

Remarques

1-2-3 considère que les rentrées de fonds sont enregistrées à intervalles réguliers.

@AMORT calcule le taux de rentabilité interne en effectuant une série d'approximations, dont la première utilise la valeur de l'argument *estimation*. Spécifiez d'abord une estimation qui vous semble adaptée au taux de rentabilité. Sachant que plusieurs solutions peuvent être possibles, réitérez votre *estimation* si vous obtenez un résultat inférieur à 0 ou supérieur à 1. Lorsque **@AMORT** ne parvient à approcher le résultat à 0,0000001 près au bout de 30 itérations, elle génère la valeur **ERR**. Si, après plusieurs tentatives, vous obtenez toujours **ERR**, faites appel à **@VAN** pour déterminer une meilleure approximation : **@VAN** donne une valeur positive lorsque l'estimation est trop faible, une valeur négative lorsqu'elle est trop forte, et 0 lorsqu'elle est correcte. Recourez à **@MOYENNE** pour déterminer le taux de rentabilité interne à partir de plusieurs taux.

Exemple

Vous souhaitez calculer le taux de rentabilité interne d'un investissement initial de 100.000 F, suivi de 12 rentrées de fonds mensuelles de 15.000 F. L'*estimation* (12,00 %) et les rentrées de fonds figurent respectivement dans les champs ESTIMATION et FONDS.

@AMORT(ESTIMATION;FONDS) renvoie un taux de rentabilité interne de 10,45 %.

Fonctions similaires

@VAN calcule la valeur actuelle nette d'une série de rentrées de fonds futures. **@VALACT** et **@VALACTBIS** calculent la valeur actuelle d'un investissement sur la base d'une série de versements égaux. **@VALFUT** et **@VALFUTBIS** calculent la valeur future d'un investissement. **@TAUX** donne le taux d'intérêt nécessaire pour qu'un investissement atteigne une valeur future donnée. **@AMORTMOD** calcule le taux de rentabilité interne modifié d'une série de mouvements de trésorerie.

@ISAAF, @ISAPP, @ISMACRO

@ISAAF(*nom*) renvoie 1 (vrai) si le *nom* spécifié correspond à une fonction LotusScript globale apportée par une application complémentaire (ou "add-in"), et 0 (faux) dans le cas contraire.

@ISAPP(*nom*) renvoie 1 (vrai) si le *nom* spécifié correspond à une application complémentaire actuellement rattachée, et 0 (faux) si l'application complémentaire n'est pas en mémoire.

@ISMACRO(*nom*) renvoie 1 (vrai) si le *nom* spécifié correspond à une sous-routine LotusScript globale apportée par une application complémentaire, et 0 (faux) dans le cas contraire.

Arguments

nom est le nom du fichier classeur complémentaire (excluant l'extension .12A) ou de la fonction ou sous-routine que vous souhaitez tester, entrée en tant que texte.

Remarques

@ISAPP retourne 1 (vrai) uniquement dans le cas d'applications complémentaires chargées au moyen de la séquence de commandes Fichier - Applications complémentaires - Gérer appl. complémentaire.

@ISAAF et @ISMACRO renvoient vrai si vous indiquez une fonction par son nom et que cette fonction se trouve dans le classeur actuel ou dans une application complémentaire chargée. Pour vérifier une fonction dans un autre classeur, indiquez le nom du fichier et de la fonction, comme dans "<<monfichier>>mafonction".

1-2-3 reconnaît les fonctions et les sous-routines à condition que leurs arguments et types de renvoi soient autorisés. Pour les nombres à virgules flottantes, 1-2-3 ne reconnaît pas le type Simple : toutes les fonctions et sous-routines doivent utiliser le type Double.

Les champs sont des arguments autorisés, mais ils doivent être indiqués comme arguments Variante. Les noms des objets 1-2-3 ne sont pas des types d'argument autorisés.

Exemples

@ISAAF("degrees") = 1 si DEGREES est une fonction script apportée par une application complémentaire.

@ISAAF("bsomme") = 0, parce que @BSOMME est une fonction @ 1-2-3 et non une fonction script complémentaire.

@ISAPP("finance") = 1 si une application complémentaire appelée FINANCE est actuellement en mémoire.

@ISMACRO("largeurchamp") = 1 si {LARGEURCHAMP} est une sous-routine script complémentaire.

@ESTERR

@ESTERR(x) vérifie si x contient la valeur ERR et renvoie 1 (vrai) le cas échéant, et 0 (faux) dans le cas contraire.

Argument

x peut être une valeur, un emplacement, du texte ou une condition.

Emploi

Associée à @SI dans les formules, @ESTERR met fin à l'effet de répercussion de la valeur ERR. Ainsi, @SI(@ESTERR(C3);0;C3) renvoie 0 si C3 contient ERR, et le contenu de C3 dans le cas contraire.

@ESTERR permet d'intercepter une erreur générée par une division par zéro. Ainsi, la formule @SI(@ESTERR(A1/A2);0;A1/A2) teste le résultat de la division de A1 par A2. @ESTERR renvoie 0 si A1/A2 génère la valeur ERR, et la valeur effectivement obtenue dans le cas contraire.

Remarque

Le libellé ERR et la valeur ERR ne sont pas équivalents. Ainsi, @ESTERR(B2) donne 0 (faux) lorsque B2 comporte le libellé ERR.

Exemple

La sous-routine VERIF_QTE se compose de trois sous-routines qui testent le contenu des cellules QUANTITE et PRIX. VERIF_QTE vérifie si QUANTITE contient une valeur et transfère, le cas échéant, le contrôle à la sous-routine VERIFERRND ; si QUANTITE ne comporte pas de valeur, AUTRE_QTE demande une nouvelle entrée avant de redonner le contrôle à VERIF_QTE.

@ESTERR permet à VERIFERRND de déterminer si QUANTITE comprend la valeur ERR. Lorsque @ESTERR donne 1 (vrai), vous devez indiquer une nouvelle valeur. Si QUANTITE et PRIX n'affichent pas respectivement ERR et ND, la sous-routine multiplie les valeurs des deux cellules et affiche le résultat dans la cellule TOTAL.

...

```
VERIF_QTE {SI @ESTNUM(QUANTITE)}{BRANCHE VERIFERRND}
AUTRE_QTE {LITVALEUR "Entrez la quantité : ";QUANTITE}
          {BRANCHE VERIF_QTE}
VERIFERRND {SI @ESTERR(QUANTITE)}{BRANCHE AUTRE_QTE}
           {SI @ESTND(PRIX)}{LITVALEUR "Entrez un nouveau prix : ";PRIX}{BRANCHE VERIF_QTE}
           {VA}TOTAL~+QUANTITE*PRIX~
```

...

Fonction similaire

@ESTND teste la valeur ND.

@ESTND

@ESTND(x) vérifie si x contient la valeur ND et renvoie 1 (vrai) le cas échéant, et 0 (faux) dans le cas contraire.

Argument

x peut être un nombre, du texte, une condition ou un emplacement.

Emploi

Associée à @SI dans les formules, @ESTND met fin à l'effet de répercussion de la valeur ND. Ainsi, @SI(@ESTND(C3);0;C3) donne 0 si C3 contient ND, et le contenu de C3 dans le cas contraire.

Remarque

Le libellé ND et la valeur ND ne sont pas équivalents. Ainsi, @ESTND(B2) donne 0 (faux) lorsque B2 comporte le libellé ND.

Exemple

La sous-routine VERIF_QTE se compose de trois sous-routines qui testent le contenu des cellules QUANTITE et PRIX. VERIF_QTE vérifie si QUANTITE contient une valeur et transfère, le cas échéant, le contrôle à la sous-routine VERIFERRND ; si QUANTITE ne comporte pas de valeur, AUTRE_QTE demande une nouvelle entrée avant de redonner le contrôle à VERIF_QTE.

@ESTND permet à VERIFERRND de déterminer si PRIX comprend la valeur ND. Lorsque @ESTND renvoie 1 (vrai), vous devez indiquer une nouvelle valeur. Si PRIX et QUANTITE n'affichent pas respectivement ND et ERR, la sous-routine multiplie les valeurs des deux cellules et affiche le résultat dans la cellule TOTAL.

```
...
VERIF_QTE {SI @ESTNUM(QUANTITE)}{BRANCHE VERIFERRND}
AUTRE_QTE {LITVALEUR "Entrez la quantité : ";QTE}
          {BRANCHE VERIF_QTE}
VERIFERRND {SI @ESTERR(QUANTITE)}{BRANCHE AUTRE_QTE}
           {SI @ESTND(PRIX)}{LITVALEUR "Entrez un nouveau prix : ";PRIX}{BRANCHE VERIF_QTE}
           {VA}TOTAL~+QTE*Prix~
...
```

Fonction similaire

@ESTERR teste la valeur ERR.

@ESTNUM

@ESTNUM(x) vérifie si x contient une valeur. **@ESTNUM** renvoie 1 (vrai) si x contient un nombre, la valeur **ND**, la valeur **ERR** ou est vierge, et 0 (faux) si x est du texte, ou une cellule contenant un libellé ou une formule générant un libellé.

Argument

x peut être une valeur, un emplacement, du texte ou une condition. Lorsque x est un champ à cellules multiples, **@ESTNUM** donne 0 (faux), même si la première cellule du champ comprend une valeur.

Emploi

Utilisée dans les macros, **@ESTNUM** permet de vérifier que l'utilisateur a bien entré le type de données qui convient. **@SI(@ESTNUM(C6);"";"ERREUR - Entrez une valeur")**, par exemple, renvoie la chaîne vide si C6 contient une valeur, ou un message d'erreur si elle contient un libellé.

Exemple

La sous-routine **VERIF_QTE** se compose de trois sous-routines qui vérifient le contenu des cellules **QUANTITE** et **PRIX**.

@ESTNUM permet à **VERIF_QTE** de déterminer si **QUANTITE** contient une valeur, et transfère, le cas échéant, le contrôle à la sous-routine **VERIFERRND**. Si **QUANTITE** ne contient pas de valeur, **AUTRE_QTE** vous demande une autre entrée.

```
...
VERIF_QTE {SI @ESTNUM(QUANTITE)}{BRANCHE VERIFERRND}
AUTRE_QTE {LITVALEUR "Entrez la quantité : ";QUANTITE}
          {BRANCHE VERIF_QTE}
VERIFERRND {SI @ESTERR(QUANTITE)}{BRANCHE AUTRE_QTE}
           {SI @ESTND(PRIX)}{LITVALEUR "Entrez un nouveau prix : ";PRIX}{BRANCHE VERIF_QTE}
           {VA}TOTAL~+QTE*Prix~
...

```

Fonctions similaires

@ESTCHAINE vérifie la présence d'un libellé. **@CELLULE** et **@POINTCELL** déterminent si une cellule contient une valeur ou un libellé.

@ESTCHAMP

@ESTCHAMP(*champ*) vérifie si *champ* correspond à un nom de champ défini ou à une adresse de champ valide (composée des lettres de feuille et de colonne allant de A à IV et d'un numéro de ligne allant de 1 à 8192).

@ESTCHAMP renvoie 1 (vrai) quand *champ* est un nom de champ défini ou une adresse de champ valide, et 0 (faux) dans le cas contraire.

Arguments

champ est un nom ou une adresse de champ.

Emploi

Associée à @SI, la fonction @ESTCHAMP permet de vérifier la validité des noms de champ devant servir dans les appels de sous-routines ou les branchements effectués à l'aide de {LITVA}.

Elle ne peut être utilisée que si les classeurs correspondants sont chargés en mémoire.

Remarques

Le tableau ci-dessous présente les différents cas de figure.

Valeur de l'argument champ	Résultat
Adresse de cellule valide	1 (vrai)
Adresse de champ valide	1
Champ nommé	1
Adresse de cellule précédée du signe plus (+)	0 (faux)
Nom de champ non défini	0
Nombre ou chaîne	0
@ERR ou @ND	0

Exemples

@ESTCHAMP(A1) renvoie 1 (vrai).

@ESTCHAMP(+A1) renvoie 0 (faux).

@ESTCHAMP(A1..C3) renvoie 1 (vrai).

@ESTCHAMP(VENTES) renvoie 1 (vrai), si VENTES est un nom de champ défini.

@ESTCHAMP(PRIX) renvoie 0 (faux), si PRIX est un nom de champ non défini.

@ESTCHAMP(3) renvoie 0 (faux).

@ESTCHAMP("COMMISSION") renvoie 0 (faux), car l'argument est une chaîne (texte entre guillemets).

@ESTCHAINE

@ESTCHAINE(x) vérifie si x contient du texte ou un libellé. **@ESTCHAINE** renvoie 1 (vrai) lorsque x correspond à du texte ou une cellule comportant un libellé ou une formule texte, et 0 (faux) lorsque x est un nombre, la valeur ERR ou ND, ou une cellule vierge.

Argument

x est une valeur, un emplacement, du texte ou une condition.

Emploi

Utilisée dans les macros, **@ESTCHAINE** permet de vérifier que l'utilisateur a bien tapé le type de données qui convient. Ainsi, **@SI(@ESTCHAINE(C6);"";"ERREUR - Entrez un libellé")** renvoie la chaîne vide si C6 contient un libellé, ou un message d'erreur s'il s'agit d'une valeur.

Exemple

La sous-routine **VERIFCHAINE** vérifie le contenu de la cellule **CLIENT**. Si cette dernière comporte un libellé (**@ESTCHAINE(CLIENT)=1**), **VERIFCHAINE** transfère le contrôle à une autre sous-routine nommée **COMMANDE**. Dans le cas contraire, **VERIFCHAINE** demande une autre entrée.

...

```
VERIFCHAINE {SI @ESTCHAINE(CLIENT)}{BRANCHE COMMANDE}
            {LITLIBELLE "Entrez un nom de client : ";CLIENT}
            {VERIFCHAINE}
```

...

Fonctions similaires

@ESTNUM vérifie la présence d'une valeur. **@CELLULE** et **@POINTCELL** déterminent si une cellule contient une valeur ou un libellé.

@KURTOSIS

@KURTOSIS(*champ*;*[type]*) renvoie le Kurtosis des valeurs de *champ*.

Arguments

champ est le nom ou l'adresse d'un champ contenant les valeurs d'un échantillon . Si *champ* contient moins de quatre valeurs, @KURTOSIS renvoie ERR.

type est un argument facultatif qui précise la nature de *champ* et donc, le type de résultat renvoyé par la fonction @KURTOSIS.

<u>type</u>	<u>1-2-3 renvoie</u>
0	Le Kurtosis d'une population ; valeur par défaut si vous omettez l'argument
1	Le Kurtosis d'un échantillon

Remarques

Le Kurtosis caractérise le "relief" d'une distribution. Un Kurtosis positif indique une distribution plutôt inégale. Au contraire, un résultat négatif est caractéristique d'une distribution de nature aplatie.

Exemples

Le champ DONNEES contient les valeurs suivantes : 2, 5, 5, 9, 1, 2, 4.

@KURTOSIS(DONNEES;1) = 1,021488

@KURTOSIS(DONNEES) = -0,32438

Fonction similaire

@SYMDIST évalue la symétrie des valeurs d'un champ par rapport à la moyenne.

@GRAND

@GRAND(*champ*;n) renvoie la *n*ième plus grande valeur trouvée dans *champ*.

Arguments

champ est le nom ou l'adresse du champ où se fait la recherche.

n peut être n'importe quel nombre positif entier. Si *n* est supérieur au nombre de valeurs dans *champ*, @GRAND renvoie ERR.

Exemples

Le champ RESULTATS contient les nombres: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86.

@GRAND(RESULTATS;3) renvoie 90, la troisième plus grande valeur de la liste.

Fonctions similaires

@PETIT renvoie la *n*ième plus petite valeur trouvée dans un champ. @MAX et @MAXNUM renvoient la plus grande valeur. @MIN et @MINNUM renvoient la plus petite.

@DEBUT

@DEBUT(*chaîne*;n) extrait les *n* premiers caractères de l'argument *chaîne*.

Arguments

chaîne peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

n, numéro de position, est un nombre entier positif ou nul, ou le nom ou l'adresse d'une cellule comportant ce type de nombre. Si *n* est égal à 0, la fonction donne une chaîne vide. Si *n* est supérieur au nombre de caractères de *chaîne*, **@DEBUT** affiche la chaîne dans son intégralité.

Emploi

@DEBUT permet de copier le début d'un libellé d'une cellule dans une autre (pour séparer les titres ou qualités - tels Dr ou Mlle - des noms, par exemple). Intégré dans les macros, **@DEBUT** peut servir à extraire le début d'un libellé pour le transférer dans une base de données, l'utiliser pour un appel de sous-routine, ou modifier la macro elle-même. Associez **@DEBUT** à **@CHERCHE**, lorsque la valeur exacte de *n* est variable ou inconnue.

Remarque

@DEBUT compte les signes de ponctuation et les espaces comme des caractères.

Exemples

@DEBUT(TEL;3) donne les trois premiers chiffres du numéro de téléphone enregistré dans la cellule TEL.

@DEBUT(A1;**@CHERCHE**(" ";A1;0)) restitue le premier mot figurant en A1. **@CHERCHE** calcule d'abord la position du premier espace rencontré dans A1, puis **@DEBUT** extrait les caractères de A1 situés avant cette position. Ainsi, elle renvoie Manuela si A1 contient Manuela Cuenca.

Fonctions similaires

@MILIEU extrait des caractères d'une *chaîne*. **@FIN** extrait les *n* derniers caractères d'une *chaîne*.

@LONG

@LONG(*chaîne*) compte les caractères de la *chaîne*.

Argument

chaîne peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

Emploi

Associée à @SUPPRESPEACE, @LONG calcule le nombre de caractères d'une *chaîne* en excluant les espaces initiaux, finaux et superflus.

@LONG peut s'utiliser également dans toute application requérant des libellés d'une longueur donnée (identificateurs de bons de commande, par exemple).

Remarque

@LONG compte les signes de ponctuation et les espaces comme des caractères.

Exemples

@LONG("fiscal") renvoie 6.

@LONG(A5&G12) calcule le nombre total de caractères présents dans les cellules A5 et G12.

@LONG(@SUPPRESPEACE(" Mlle Pinson")) renvoie 9.

@LN

@LN(x) calcule le logarithme népérien (en base e) de x.

Argument

x est une valeur positive ou le nom ou l'adresse d'une cellule contenant une telle valeur.

Emploi

@LN s'utilise dans les calculs scientifiques faisant intervenir des logarithmes népériens, tels les calculs de croissance ou de perte composée.

Remarque

Les logarithmes népériens (ou logarithmes naturels) utilisent le nombre e (environ 2,718282) comme base.

Exemples

@LN(2) donne 0,6931.

@LN(@EXP(1)) donne 1, car @EXP(1) est égal à 2,718282.

Fonctions similaires

@EXP est l'inverse de @LN. @LOG calcule le logarithme décimal (en base 10) de x.

@LOG

@LOG(x) calcule le logarithme décimal (en base 10) de x .

Argument

x est une valeur positive ou le nom ou l'adresse d'une cellule contenant une telle valeur.

Emploi

@LOG s'utilise dans les calculs faisant intervenir des logarithmes décimaux (formule calculant la racine carrée d'un nombre, par exemple).

Exemples

$10^{(@LOG(8)/3)}$ donne 2 (racine cubique de 8).

@LOG(B3) donne 0,6021, si B3 contient la valeur 4.

Fonction similaire

@LN calcule le logarithme népérien (en base e) d'une valeur.

@MINUSC

@MINUSC(*chaîne*) met en minuscules toutes les lettres de l'argument *chaîne*.

Argument

chaîne peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

Emploi

@MINUSC permet d'homogénéiser un fichier contenant des libellés émanant de différentes sources.

Remarque

Si vous utilisez l'ordre de tri ASCII (au moyen de la séquence de commandes Fichier - Personnalisation - Paramètres par défaut, sous "Ordre de tri par pays de l'onglet Général), la mise en majuscules affecte l'ordre de tri des libellés ; deux libellés, identiques par ailleurs, ne seront donc pas considérés comme ex aequo si leur mise en majuscule est différente.

Exemples

@MINUSC("Prévisions Ventes") renvoie prévisions ventes ;

Fonctions similaires

@MAJUSC met en majuscules tous les caractères d'une *chaîne*. @NOMPROPRE convertit en majuscule l'initiale de chacun des mots d'une *chaîne*, et en minuscules les autres caractères.

@MAX, @MAXNUM

@MAX(*liste*) renvoie la plus grande valeur trouvée dans *liste*.

@MAXNUM(*liste*) renvoie la plus grande valeur trouvée dans *liste*, sans tenir compte des cellules qui contiennent des libellés.

Arguments

liste peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

Le champ nommé TEST contient les données suivantes: -5, -7, -9, -11, Janvier.

@MAX(TEST) retourne 0, le libellé Janvier étant compté pour 0, comme plus grande valeur dans le champ TEST.

@MAXNUM(TEST) ignore le libellé Janvier et retourne -5.

Fonctions similaires

@MIN et **@MINNUM** renvoient la plus petite valeur d'une liste. **@BMAX** renvoie la plus grande valeur d'une rubrique de base de données qui répond aux critères de sélection. **@GRAND** renvoie la *nième* plus grande valeur d'une liste.

@MEDIANE

@MEDIANE(*liste*) renvoie la valeur médiane de *liste*.

Arguments

liste peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Remarques

Si *liste* contient un nombre impair d'éléments, **@MEDIANE** retourne la valeur centrale. Si *liste* contient un nombre pair d'éléments, **@MEDIANE** retourne la moyenne arithmétique des deux valeurs centrales.

Exemples

@MEDIANE(5;12;65;82;9) = 12

@MEDIANE(5;12;65;82;9;78) = 38,5

Fonctions similaires

@PUREMEDIAN renvoie la valeur médiane dans *liste* ; elle ne tient pas compte des cellules vides, des libellés ni des formules qui renvoient des libellés.

@MOYGEOM calcule la moyenne géométrique des valeurs d'une liste. **@MOYHARM** calcule la moyenne harmonique des valeurs d'une liste. **@MOYENNE** et **@MOYENNENUM** calculent la moyenne arithmétique des valeurs d'une liste.

@MILIEU

@MILIEU(*chaîne;position;n*) extrait *n* caractères de *chaîne*, à partir du caractère indiqué par l'argument *position*.

Arguments

chaîne peut être du texte, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

position, numéro de position, est un nombre entier positif ou nul, ou le nom ou l'adresse d'une cellule comportant ce nombre. Si la valeur de *position* est supérieure au nombre de caractères de *chaîne*, @MILIEU génère une chaîne vide.

n, numéro de position, est un nombre entier positif ou nul, ou le nom ou l'adresse d'une cellule comportant ce type de nombre. Si *n* est égal à 0, la fonction donne une chaîne vide.

n peut être un entier positif ou nul, ou le nom ou l'adresse d'une cellule contenant un tel nombre. Si *n* est égal à 0, @MILIEU renvoie une chaîne vide; si *n* est supérieur au nombre de caractères de *chaîne*, elle renvoie tous les caractères compris entre *position* et la fin de *chaîne*.

Emploi

@MILIEU extrait une chaîne ne figurant ni en début, ni en fin de libellé. Lorsque vous ignorez la *position* du premier caractère à extraire, servez-vous de @MILIEU et de @CHERCHE conjointement.

@MILIEU s'utilise dans les macros, pour enregistrer des parties de libellés, créer des appels de sous-routines, ou modifier la macro elle-même.

Remarque

@MILIEU traite les signes de ponctuation et les espaces comme des caractères. Si vous ignorez la longueur de *chaîne*, indiquez une valeur *n* élevée : 1-2-3 affiche la totalité de *chaîne*, en partant de la *position* spécifiée.

Exemple

@MILIEU("Compte courant crédité";7;7) renvoie courant.

Fonctions similaires

@DEBUT extrait les *n* premiers caractères d'une *chaîne*, et @FIN les *n* derniers caractères.

@MIN, @MINNUM

@MIN(*liste*) renvoie la plus petite valeur trouvée dans *liste*.

@MINNUM(*liste*) renvoie la plus petite valeur trouvée dans *liste*, sans tenir compte des cellules qui contiennent des libellés.

Arguments

liste peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi [Arguments des fonctions @ statistiques](#).

Exemples

Le champ nommé TEST contient les données suivantes: 5, 7, 9, 11, Janvier.

@MIN(TEST) retourne 0, le libellé Janvier étant compté pour 0, comme plus petite valeur dans le champ TEST.

@MINNUM(TEST) ignore le libellé Janvier et retourne 5.

Fonctions similaires

@MAX et **@MAXNUM** renvoient la plus grande valeur d'une liste. **@BMIN** renvoie la plus petite valeur d'une rubrique de base de données qui répond aux critères de sélection.

@MINUTE

`@MINUTE(nombre-temps)` extrait les minutes (valeur comprise entre 0 et 59) de l'argument *nombre-temps*.

Arguments

nombre-temps est une valeur comprise entre 0,000000 (minuit) et 0,999988 (23:59:59 ou 11:59:59 PM), ou le nom ou l'adresse d'une cellule contenant cette valeur. L'argument *nombre-temps* est généralement généré par une autre fonction @.

Emploi

@MINUTE permet d'extraire uniquement les minutes des nombres produits par les fonctions @ACTUEL, @TEMPS ou @HEURVAL afin d'effectuer des calculs spécifiques (temps écoulé depuis le lancement d'une application, par exemple).

Exemples

@MINUTE(0,333) donne 59, car 0,333 est le nombre-temps correspondant à 7:59:31.

@MINUTE(@TEMPS(11;15;45)) donne 15, car 15 est l'argument *minutes* de la fonction @TEMPS(11;15;45).

Fonctions similaires

@HEURE et @SECONDE extraient respectivement les heures et les secondes d'un nombre temps.

@AMORTMOD

@AMORTMOD(*champ*; *taux-payé*; *taux-perçu*; [*type*]) calcule le taux de rentabilité interne modifié d'une série de mouvements de trésorerie générés par un investissement.

Le taux de rentabilité interne est le pourcentage pour lequel la valeur actuelle d'une série de rentrées de fonds futures est égale à l'investissement initial.

Arguments

champ est le nom ou l'adresse du champ contenant les mouvements de trésorerie (cash-flows). Les nombres négatifs sont considérés comme des sorties de fonds et les nombres positifs comme des rentrées de fonds. *champ* doit contenir au moins une valeur positive et une valeur négative.

Normalement, la première valeur de *champ* doit être négative car elle représente l'investissement. 1-2-3 assigne la valeur 0 aux cellules vierges et aux libellés figurant dans *champ* et les intègre dans ses calculs.

taux-payé est le taux d'intérêt payé sur l'argent emprunté.

taux-perçu est le taux d'intérêt perçu sur l'argent réinvesti.

taux-payé et *taux perçu* peuvent être de quelconques valeurs.

type est un argument facultatif qui permet de préciser si les mouvements de trésorerie ont lieu en fin ou en début de période.

<u>type</u>	<u>Les mouvements de trésorerie interviennent</u>
0	A la fin de chaque période ; valeur par défaut si vous omettez l'argument
1	Au début de chaque période

Remarques

Utilisez @AMORTMOD pour déterminer la rentabilité d'un investissement. Combinez cette fonction avec les autres fonctions financières, telles @VAN, pour évaluer un investissement.

1-2-3 considère que les mouvements de trésorerie sont reçus à intervalles réguliers égaux.

Seules la version 5 de 1-2-3 et ultérieures admettent l'argument facultatif *type*. Si vous sauvegardez une formule @AMORTMOD avec l'argument *type* puis ouvrez le classeur correspondant dans une version précédente de 1-2-3, la formule renvoie ERR.

Exemple

Vous avez acheté un studio et vous avez emprunté pour cette acquisition 100.000 F à 9,5% d'intérêt annuel, il y a cinq et six ans. La liste ci-dessous, enregistrée dans le champ REVENU, contient vos investissements initiaux et les revenus perçus:

-100.000
-100.000
45.500
47.000
48.500
50.000
50.000

Durant ces années, vous avez réinvesti vos revenus à un taux d'intérêt annuel de 11%.

@AMORTMOD(REVENU;0,095;0,11) = 7,96%

Le taux de rentabilité est légèrement inférieur si vos investissements sont effectués en fin d'année.

@AMORTMOD(REVENU;0.095;0,11;1) = 7,70%

Fonction similaire

@AMORT calcule le taux de rentabilité interne. @XIRR renvoie le taux de rentabilité interne pour une série d'entrées et de sorties de liquide.

@MOD, @MODULO

@MOD(x;y) et @MODULO(x;y) calculent le reste (modulo) de la division de x par y.

Arguments

x est une valeur, ou le nom ou l'adresse d'une cellule contenant une valeur. Lorsque x est nul, @MOD et @MODULO retournent 0.

y est une valeur différente de 0.

Remarques

La valeur renvoyée par @MOD est égale à $x - y * @ENT(x/y)$ et est toujours de même signe (+ ou -) que x.

La valeur renvoyée par @MODULO est égale à $x - y * @ARRDEF(x/y)$ et est toujours de même signe (+ ou -) que y.

Exemples

@MOD(9;4) = 1.

@MODULO(9;4) = 1

@MOD(-14;3) = -2.

@MODULO(-14;3) = 1

Fonction similaire

@QUOTIENT calcule x/y et tronque le résultat à un nombre entier.

@MOIS

@MOIS(*nombre-date*) extrait le mois d'un *nombre-date*, sous forme d'une valeur comprise entre 1 et 12.

Argument

nombre-date est un nombre compris entre 1 (1er janvier 1900) et 73050 (31 décembre 2099), ou le nom ou l'adresse d'une cellule contenant ce type de nombre.

Emploi

@MOIS permet d'extraire uniquement le mois d'un nombre date, plutôt que la date complète. Vous l'utiliserez donc pour effectuer le suivi d'événements survenant durant certains mois de l'année (établissement de rapports de ventes trimestriels, par exemple), ou pour calculer le nombre de mois séparant deux événements.

@MOIS peut également servir d'argument *mois* à d'autres fonctions de date reposant sur des dates préalablement calculées.

Remarque

Vous pouvez utiliser une autre fonction de date @ comme *nombre-date*.

Exemples

@MOIS(@DATE(91;3;27)) donne 3, 3 étant l'argument *mois* de @DATE(91;3;27).

@MOIS(20181) donne 4, car 20181 est le nombre date correspondant au 2 avril 1955.

@MOIS(@ACTUEL) génère la valeur correspondant au mois actuel.

Fonctions similaires

@JOUR et @ANNEE extraient respectivement le jour et l'année d'un nombre date.

Exemple de fonction @REGRESSION

Vous vendez des glaces sur un site touristique. Vous supposez que le nombre de glaces vendues est directement lié à la température extérieure à midi, au nombre d'heures de soleil dans la journée et au nombre de bus stationnés sur le site. Pour évaluer approximativement la vente des prochains jours, vous voulez vérifier la validité de cette hypothèse.

Vous saisissez ces différentes informations recueillies sur six jours dans une feuille 1-2-3.

A ---- A ----- B -----C-----D-----

1	Glaces	Soleil	Temp.	Bus
2	250	3	84	10
3	545	5	91	7
4	550	5	89	8
5	450	6	85	10
6	605	6	90	11
7	615	7	88	9

@REGRESSION(B2..D7;A2..A7;3) = 0,977225

Vous vous rendez compte que la valeur de R au carré est proche de 1. Il y a donc une très forte corrélation entre la vente de vos glaces et le nombre d'heures d'ensoleillement, la température à midi et le nombre de bus.

@N

@N(*champ*) convertit en nombre, le contenu de la première cellule de *champ*. S'il s'agit d'un libellé, **@N** génère la valeur 0.

Argument

L'argument *champ* doit être une adresse de cellule ou de champ, ou un nom de champ.

Emploi

@N peut être associé à toute autre fonction de chaîne ou formule pour s'assurer de la présence d'une valeur dans une cellule qui pourrait comporter un libellé. **@N** permet également d'éviter que des formules n'entraînent l'affichage de **ERR**. Cette fonction est particulièrement utile pour vérifier ce qu'entre un utilisateur lors de l'exécution d'une macro.

Exemples

+100+**@N**(B5..F5) = 885 si B5 contient le nombre 785.

@N(A5)+**@N**(B5) donne 785 si A5 contient un libellé et B5 la valeur 785.

Fonctions similaires

@C convertit en libellé le contenu de la première cellule d'un champ. **@ESTNUM** permet de déterminer si une cellule contient une valeur.

@ND

@ND génère la valeur ND (non disponible).

Emploi

La fonction logique @ND se révèle particulièrement utile lorsque vous créez une feuille destinée à contenir des données non encore déterminées. Entrez @ND dans les cellules où se trouveront ces données : les formules se référant à ces cellules généreront la valeur ND tant que vous n'aurez pas fourni les données manquantes.

@ND permet également de déterminer quelles sont les formules qui dépendent d'une cellule donnée.

Remarque

Le libellé ND et la valeur ND ne sont pas équivalents dans les formules : +A2+34=ND si A2 contient @ND, mais cette même formule donnera 34 si A2 comporte le libellé ND.

Exemple

@SI(@CELLULE("type";B14)="";@ND;B14) donne la valeur ND si B14 est vierge.

Fonctions similaires

@ERR produit la valeur ERR. @ESTND vérifie la présence de la valeur ND.

@ACTUEL

@ACTUEL renvoie un nombre correspondant à l'instant actuel (date et heure). La partie entière de cette valeur représente le nombre_date, et la partie décimale, le nombre_temps.

Emploi

La fonction de date et d'heure @ACTUEL s'utilise pour évaluer une durée écoulée à partir d'une date/heure donnée (l'âge à partir de la date de naissance, par exemple). La date renvoyée est remise à jour à chaque recalcul. Pour figer le résultat à un instant précis (horodatage d'une feuille ou évaluation de la durée d'une opération, par exemple), utilisez les touches F2 (MODIFIE) et F9 (CALCUL), ou introduisez le résultat dans une cellule au moyen d'une macro.

Remarques

Vous pouvez affecter au résultat l'un des formats de date ou d'heure de 1-2-3. Si vous optez pour un format de date, @ACTUEL affiche uniquement la partie entière du résultat (ou nombre date), alors que si vous choisissez un format d'heure, seule la partie décimale (ou nombre temps) apparaît. Dans les deux cas, il ne s'agit que d'affichage, 1-2-3 conserve en mémoire le résultat entier.

1-2-3 recalcule @Actuel chaque fois que vous recalculez votre travail.

Exemples

@ACTUEL = 31050,5 (le 3 janvier 1985 à midi).

@ACTUEL = 33418,395 (le 29 juin 1991 à 9:28).

Fonction similaire

@JOURACTUEL renvoie un nombre entier correspondant à la date du jour, sans les décimales de l'heure.

@VAN

@VAN(*taux*;*champ*;*[type]*) calcule la valeur actuelle nette d'une série de mouvements de trésorerie futurs (*champ*), à laquelle est appliqué un *taux* d'intérêt périodique constant.

Arguments

taux est une valeur décimale ou un pourcentage supérieur à -1, ou le nom ou l'adresse d'une cellule comportant l'un de ces deux types de valeurs.

champ est un champ contenant les mouvements de trésorerie.

type est un argument facultatif qui permet de préciser si les mouvements de trésorerie ont lieu en fin ou en début de période.

<u>type</u>	<u>Les mouvements de trésorerie interviennent</u>
0	En fin de période ; valeur par défaut si vous omettez l'argument
1	En début de période

Remarques

La fonction financière @VAN permet d'évaluer un investissement ou de le comparer à d'autres. Elle calcule l'investissement initial nécessaire pour obtenir une certaine sortie de fonds à un taux donné.

@VAN considère que les sorties de fonds ont lieu à intervalles réguliers, la première, à la fin de la première période, et les autres, à la fin des périodes suivantes. Pour déterminer la valeur actuelle nette d'un investissement pour lequel vous avez effectué immédiatement un versement initial, suivi d'une série de rentrées de fonds futures, vous devez mettre en facteur séparément la valeur correspondant à la sortie de fonds initiale. En effet, cette dernière n'est pas concernée par le taux d'actualisation. Par exemple,

+INITIALE+@VAN(TAUX;SERIE) = 1904,07 F. INITIALE est la sortie de fonds initiale, TAUX, le taux d'intérêt et SERIE, la série de rentrées de fonds futures.

@VAN affiche la valeur ERR lorsque *champ* comprend plusieurs lignes ou plusieurs colonnes. C'est le cas notamment pour le *champ* A1..D25, mais pas pour le *champ* A1..D1 (une seule ligne) ou A1..A25 (une seule colonne).

Seules la version 5 de 1-2-3 et ultérieures admettent l'argument facultatif *type*. Si vous sauvegardez une formule @VAN avec l'argument *type* puis ouvrez le classeur correspondant dans une version précédente de 1-2-3, la formule renvoie ERR.

Exemple

Dans l'exemple ci-dessous, @VAN permet d'actualiser une série de versements irréguliers investis au taux annuel de 11,5 %.

La liste des sorties de fonds est enregistrée dans le *champ* VERSEMENTS (une sortie de fonds par mois pendant douze mois).

0,00 F	
0,00 F	
2.5000,00 F	
2.5000,00 F	
3.0000,00 F	
5.0000,00 F	
6.0000,00 F	
9.0000,00 F	
3.0000,00 F	
2.5000,00 F	
0,00 F	
7.5000,00 F	
41.0000,00 F	TOTAL

Pour que @VAN identifie le nombre exact de périodes, les mois ne donnant lieu à aucun versement doivent

apparaître dans *champ*. Les versements étant mensuels, le *taux* d'intérêt (ou taux d'actualisation) doit être exprimé sous forme d'un pourcentage mensuel (cellule TAUX) :

$$0,115/12 = 0,0096$$

$$\text{@VAN(TAUX;VERSEMENTS)} = 38.084,13 \text{ F}$$

Le résultat est différent si les versements interviennent en début de période :

$$\text{@VAN(TAUX;VERSEMENTS;1)} = 38.449,10 \text{ F}$$

Fonctions similaires

@VALACT calcule la valeur actuelle d'un investissement basé sur une série de versements réguliers. @VALFUT calcule sa valeur future.

@XNPV renvoie la valeur nette présente d'une série d'entrées et de sorties de fonds.

@IMPAIR

@IMPAIR(x) arrondit le nombre x au nombre impair supérieur si x est positif ou nul, au nombre impair inférieur si x est négatif. Dans le cas où x est un nombre impair, la fonction retourne x .

Arguments

x est un nombre quelconque.

Exemples

@IMPAIR(3,25) = 5

@IMPAIR(3) = 3

@IMPAIR(-3,25) = -5

Fonctions similaires

@PAIR arrondit un nombre au nombre pair le plus proche. @ARRONDI, @ARRDEF, et @ARREXC arrondissent un nombre aux nombres de décimales spécifiés. @ARRMUL arrondit un nombre au multiple spécifié. @ENT renvoie la partie entière d'un nombre. @TRONQUE tronque un nombre à droite ou à gauche de la virgule.

@POURCENTILE

@POURCENTILE(*x*;*champ*) évalue le *x*ème pourcentile de l'échantillon parmi les valeurs de *champ*.

Arguments

x représente le pourcentile recherché. C'est donc un nombre compris entre 0 et 1.

champ est le nom ou l'adresse d'un champ contenant les valeurs de l'échantillon.

1-2-3 considère les libellés comme des nombres nuls et en tient compte dans les calculs. Par contre, les cellules vides sont ignorées.

Remarques

Par définition, le *x*ème pourcentile d'une liste de valeurs est une valeur telle que *x*% des valeurs lui soient inférieures et (100-*x*)% supérieures.

Pour évaluer un pourcentile, 1-2-3 trie d'abord les valeurs par ordre croissant. La plus petite valeur est le 0ème pourcentile, la plus grande, le 100ème. Chaque valeur intermédiaire correspond au $(x-1)/(n-1)$ ème pourcentile, *x* étant le numéro d'ordre de cette valeur dans la liste triée et *n*, le nombre de valeurs. Ainsi, la troisième valeur d'une liste de dix correspond au $(3-1)/(10-1)$ ème pourcentile, c'est-à-dire au 22ème pourcentile, et la quatrième, au 33ème pourcentile. Un pourcentile qui ne correspond pas exactement à l'une des valeurs est calculé par interpolation linéaire entre les deux valeurs qui l'encadrent.

Exemples

Un champ nommé RESULTATS contient les valeurs suivantes: 87, 85, 90, 80, 82, 92, 79, 85, 95 et 86. Vous voulez en déduire le 90ème pourcentile.

@POURCENTILE(0,9;RESULTATS) = 92,3

Fonction similaire

@PRANG renvoie le pourcentile correspondant à une valeur.

@DECILE renvoie un décile donné, et @QUARTILE renvoie un pourcentile donné.

@ARRANG

@ARRANG($n;p$) retourne le nombre d'arrangements de p objets parmi n . A la différence des combinaisons, l'ordre des p objets dans chaque arrangement n'est pas indifférent.

Arguments

n est un entier positif ou nul.

p est un entier positif ou nul qui ne peut pas être plus grand que n .

Si n et p ne sont pas entiers, 1-2-3 les tronque à leur partie entière.

Exemples

Trois professeurs sont à choisir parmi cinq pour surveiller trois examens prévus à 9:00, 10:00, et 11:00. Combien y-a-t-il de choix possibles?

@ARRANG(5;3) = 60

Fonction similaire

@COMBIN retourne le nombre de combinaisons de p objets parmi n . L'ordre des p objets dans une combinaison étant indifférent.

@PI

@PI produit la valeur π (sous la forme 3,14159265358979), qui est le rapport de la circonférence d'un cercle à son diamètre.

Emploi

La fonction mathématique @PI s'utilise dans les calculs se servant de la valeur π et, plus particulièrement, dans des opérations mettant en oeuvre des fonctions trigonométriques.

Exemples

@PI = 3,1415926536.

@PI*4^2 = 50,2655 (résultat arrondi à 4 décimales), soit la surface d'un cercle de rayon 4.

@REMBOUR, @REMBOURBIS

@REMBOUR(*principal*; *taux*; *périodes*) calcule le montant des versements nécessaires au remboursement d'un emprunt (*principal*), à partir du *taux* d'intérêt et du nombre de *périodes* spécifiés.

@REMBOURBIS(*principal*; *taux*; *périodes*; [*type*]; [*valeur-future*]) calcule le montant des versements nécessaires au remboursement d'un emprunt (*principal*) dont vous pouvez spécifier la *valeur-future*, à partir du *taux* d'intérêt et du nombre de *périodes* spécifiés. Selon la valeur de *type*, @REMBOURBIS prend en compte des versements effectués soit en fin, soit en début de période.

Arguments

principal et *périodes* sont des nombres.

taux est un nombre supérieur à -1.

type est un argument facultatif qui permet d'indiquer si les versements ont lieu en fin ou en début de période:

<u><i>type</i></u>	<u>Les versements interviennent</u>
0	A la fin de chaque période ; valeur par défaut si vous omettez l'argument
1	Au début de chaque période

valeur-future est un argument facultatif qui permet de spécifier la valeur future de la série de versements. *valeur-future* peut être un nombre quelconque. Si vous l'omettez, 1-2-3 utilise 0.

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarques

Les arguments utilisés doivent faire référence à la même unité de temps. Par exemple, si les versements sont mensuels, vous devrez rapporter au mois un taux d'intérêt annuel (généralement en le divisant par 12), et exprimer les périodes en mois en multipliant par 12 un nombre d'années.

Exemples

Vous envisagez d'emprunter 8.000 F sur 3 ans, à un taux annuel de 10,5% sur la base de remboursements mensuels. Si les remboursements interviennent en fin de mois, leur montant sera de :

@REMBOUR(8000;0,105/12;36) = 260,02 F

S'ils interviennent en début de mois, leur montant sera de:

@REMBOURBIS(8000;0,105/12;36;1;0) = 257,76 F

Fonctions similaires

@PMT2 calcule le versement d'un emprunt à un taux *d'intérêt* donné pour un nombre de périodes spécifié, en supposant qu'il s'agit d'un système de règlement par annuité. @PMT1 calcule les intérêts d'un paiement périodique constant. @SPI calcule les intérêts d'un versement périodique où le principal est le même pour chaque période.

@INTVER calcule les intérêts cumulés correspondant au versement périodique d'un investissement. @PRINCVER calcule le principal correspondant au versement périodique d'un investissement. @DUREE calcule le nombre de versements nécessaires pour qu'un investissement atteigne une valeur spécifiée.

@POISSON

@POISSON(*x*;*moyen*;*[cumul]*) renvoie la distribution de Poisson.

Arguments

x, entier positif ou nul, est le nombre d'événements dont on veut étudier la probabilité.

moyen, entier positif ou nul, est le nombre moyen d'événements observés.

Si *x* et *moyen* ne sont pas des nombres entiers, 1-2-3 ne prend pas en compte la partie décimale.

cumul est un argument facultatif qui définit le type de résultat renvoyé par la fonction @POISSON.

<u><i>cumul</i></u>	<u>1-2-3 renvoie</u>
0	La probabilité d'obtenir exactement <i>x</i> événements ; valeur par défaut si vous omettez l'argument
1	La probabilité d'obtenir au plus <i>x</i> événements

Remarques

@POISSON donne un résultat exact à $\pm 3 \cdot 10^{-7}$ près.

La fonction @POISSON trouve une application dans les phénomènes de passages aléatoires tels : le nombre de clients entrant dans un magasin par minute, le nombre d'avions atterrissant dans un aéroport par heure, etc.

Exemples

Vous savez qu'en moyenne 6 voitures franchissent un péage en une heure. La probabilité pour qu'au maximum, 4 voitures franchissent ce péage se calculera ainsi :

@POISSON(4;6;1) = 0,285057, soit 28,51%

Pour déterminer la probabilité d'obtenir exactement 4 passages :

@POISSON(4;6) = 0,133853, soit 13,39%

@PRANG

`@PRANG(x;champ;[ndéc])` renvoie le pourcentile correspondant à x parmi les valeurs de *champ*.

Arguments

x est un nombre quelconque.

champ est le nom ou l'adresse d'un champ contenant les valeurs de l'échantillon.

ndéc est un argument facultatif. S'il est spécifié, 1-2-3 arrondira le résultat renvoyé par @PRANG à *ndéc* chiffres après la virgule. Sinon, 1-2-3 arrondira à 2 décimales.

Remarques

Si x ne fait pas partie des valeurs de *champ*, 1-2-3 considère que la plus petite des valeurs de *champ* représente le pourcentile 0, la plus grande, le 100ème, et effectue une interpolation.

Exemples

Un champ nommé RESULTATS contient les valeurs suivantes : 87, 85, 90, 80, 82, 92, 79, 85, 95 et 86. Vous voulez en déduire le pourcentile correspondant à un résultat de 90 :

`@PRANG(90,RESULTATS)` = 0,78, soit 78%.

Fonction similaire

`@POURCENTILE` évalue le $x^{\text{ème}}$ pourcentile d'un échantillon.

@PRIX

@PRIX(*livraison*; *maturité*; *coupon*; *rendement*; [*remboursement-nominal*]; [*fréquence*]; [*base*]) calcule le prix d'une obligation à intérêt périodique de 100 F de nominal.

Arguments

livraison est la date d'achat de l'obligation, exprimée en nombre date.

maturité est la date d'échéance de l'obligation, exprimée en nombre date. Si *maturité* est inférieure ou égale à *livraison*, @PRIX retourne ERR.

coupon est le taux nominal annuel de l'obligation. Il peut être n'importe quel nombre positif ou nul.

rendement est le rendement annuel. C'est un nombre positif quelconque.

remboursement-nominal est un argument facultatif qui permet de spécifier la valeur de remboursement d'une obligation de 100 F de nominal. Ce peut être n'importe quel nombre positif ou nul ; si vous l'omettez, 1-2-3 utilise 100.

fréquence est un argument facultatif permettant d'indiquer la fréquence de paiement des coupons.

<u>fréquence</u>	<u>Fréquence de paiement des coupons</u>
1	Annuelle
2	Semi-annuelle ; valeur par défaut si vous omettez l'argument
4	Trimestrielle
12	Mensuelle

base est un argument facultatif qui spécifie le type de calendrier jours/mois à prendre en compte.

base est un argument facultatif qui spécifie le type de base de comptage des jours à utiliser. *base* est une valeur appartenant au tableau suivant :

<u>base</u>	<u>Base de comptage des jours</u>
0	30/360 ; valeur par défaut si vous omettez l'argument
1	Réel/réel
2	Réel/360
3	Réel/365
4	Européen 30/360

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Exemple

Un bon est livrable le 1er juillet 1993 et vient à échéance le 1er décembre 1998. Les coupons sont payés semi-annuellement et le taux nominal annuel est de 5,50%. Le rendement annuel est de 5,61%. Le calendrier à prendre en compte est de type 30/360.

La formule ci-dessous permet de déterminer le prix du bon :

@PRIX(@DATE(93;7;1);@DATE(98;12;1);0,055;0,0561;100;2;0) = 99,49

Fonctions similaires

@INTCOUR calcule les intérêts courus pour des titres à intérêt périodique. @RENDEMENT calcule le rendement de titres à intérêt périodique. @DURATION calcule la durée à l'origine et @SENSIBILITETITRE, la sensibilité à l'origine d'un titre de 100 F à intérêt périodique.

@PRIX2 calcule le prix pour une valeur de 100 Yens d'obligations à taux d'intérêt périodique, à l'aide de conventions japonaises.

@NOMPROPRE

@NOMPROPRE(*chaîne*) met en majuscule l'initiale de chaque mot de l'argument *chaîne*, et en minuscules les autres lettres.

Argument

chaîne peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

Emploi

La fonction de chaîne **@NOMPROPRE** permet d'unifier la présentation de données d'origines différentes. Utilisée dans une base de données, **@NOMPROPRE** peut servir à vérifier l'homogénéité d'une liste de noms avant de les trier ou de créer des libellés destinés à un publipostage, par exemple.

Remarque

Si vous utilisez un ordre de tri ASCII (au moyen de la séquence de commandes Fichier - Personnalisation - Paramètres par défaut, sous "Ordre de tri par pays" de l'onglet Général, la mise en majuscules affecte l'ordre de tri des libellés ; deux libellés, identiques par ailleurs, ne seront donc pas considérés comme ex aequo si leur mise en majuscule est différente.

Exemple

@NOMPROPRE(A7&" ; "&G7) donne Robert Dumont ; Toulouse France, si la cellule A7 comporte le libellé ROBERT DUMONT et G7, le libellé Toulouse FRANCE. Remarquez que le point-virgule, placé entre guillemets, est considéré comme une chaîne littérale et non pas comme un séparateur d'arguments.

Fonctions similaires

@MINUSC et **@MAJUSC** convertissent respectivement en minuscules et en majuscules toutes les lettres d'une *chaîne*.

@VALACT, @VALACTBIS

@VALACT(*versements*; *taux*; *périodes*) calcule la valeur actuelle d'un investissement sur la base d'une série de versements égaux, placés à un *taux* d'intérêt périodique, sur un nombre de *périodes* donné.

@VALACTBIS(*versements*; *taux*; *périodes*; [*type*]; [*valeur-future*]) calcule la valeur actuelle d'un investissement dont vous pouvez spécifier la *valeur-future*, sur la base d'une série de versements égaux, placés à un *taux* d'intérêt périodique, sur un nombre de *périodes* donné. Selon le *type* que vous spécifiez, @VALACTBIS calcule la valeur actuelle pour des versements effectués soit en fin, soit en début de période.

Arguments

versements et *périodes* sont des nombres.

taux est un nombre supérieur à -1.

type est un argument facultatif qui permet de préciser si les versements ont lieu en fin ou en début de période :

<u><i>type</i></u>	<u>Les versements interviennent</u>
0	A la fin de chaque période ; valeur par défaut si vous omettez l'argument
1	Au début de chaque période

valeur-future est un argument facultatif qui permet de spécifier la valeur future de la série de versements. *valeur-future* peut être un nombre quelconque. Si vous l'omettez, 1-2-3 utilise 0.

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarques

Les arguments utilisés doivent faire référence à la même unité de temps. Par exemple, si les versements sont mensuels, vous devrez rapporter au mois un taux d'intérêt annuel (généralement en le divisant par 12), et exprimer les périodes en mois en multipliant par 12 un nombre d'années.

@VALACT permet d'évaluer un investissement ou de comparer plusieurs placements (capital retraite versé en une seule fois ou placé sous forme de versements périodiques, par exemple). Pour créer un tableau d'amortissement, vous utiliserez conjointement @VALACT et @REMBOUR.

@VALACT complète @REMBOUR. @VALACT détermine le montant maximal de l'emprunt que vous pouvez contracter, en fonction des versements auxquels vous pouvez faire face. De son côté, @REMBOUR détermine le montant des versements sur la base du montant total de l'emprunt que vous souhaitez contracter.

Exemples

Vous avez gagné un million de francs payable soit en 20 versements annuels de 50.000 F en fin de chaque année, soit en un versement immédiat, supposé équivalent, de 400.000 F. Vous souhaitez connaître l'option la plus avantageuse en francs actuels. Si vous acceptez les paiements annuels de 50.000 F, vous faites l'hypothèse d'investir cette somme à un taux d'intérêt annuel de 8%.

@VALACT(50000;0,08;20) retourne 490.907 F, soit l'équivalent en francs actuels d'un million de francs perçu sur 20 ans,

et si vous recevez les versements en début, et non en fin d'année :

@VALACTBIS(50000;0,08;20;1) = 530.180 F

Fonctions similaires

@VALFUT et @VALFUTBIS calculent la valeur future d'un investissement sur la base d'une série de versements égaux, @VAN calcule sa valeur actuelle nette. @REMBOUR et @REMBOURBIS calculent le montant des versements nécessaires au remboursement d'un emprunt.

@VALACT2 calcule la valeur présente d'un investissement, basée sur une série de versements égaux, avec une remise sur un taux d'intérêt périodique sur le nombre de périodes du terme, en supposant qu'il s'agit d'une convention par annuité. @VALACTMONTANT renvoie la valeur présente d'une somme forfaitaire à recevoir sur un nombre donné de périodes à venir et réduite en fonction d'un taux d'intérêt donné.

@QUOTIENT

@QUOTIENT(x;y) renvoie le résultat de la division de x par y, tronqué à sa valeur entière.

Arguments

x est un nombre. Si x est égal à 0, @QUOTIENT retourne 0.

y est un nombre différent de 0.

Exemples

@QUOTIENT(7;3) = 2

@QUOTIENT(12,25;3,5) = 3

@QUOTIENT(-7;3) = -2

Fonction similaire

@MOD renvoie le reste de la division de x par y.

@RADENDEG

@RADENDEG(*radians*) convertit des *radians* en degrés.

Arguments

radians est un nombre.

Exemples

@RADENDEG(0,523599) = 30 degrés

Fonction similaire

@DEGENRAD convertit des degrés en radians.

@ALEAT

@ALEAT génère un nombre aléatoire compris entre 0 et 1 inclus, avec une précision de 15 chiffres après la virgule. A chaque recalcul de la feuille, la valeur de @ALEAT change.

Emploi

La fonction mathématique @ALEAT s'applique aux cas nécessitant la génération de nombres aléatoires (pour obtenir des données de test dans des analyses de simulation, par exemple, ou sélectionner des valeurs à vérifier).

Remarques

Pour que le nombre généré par @ALEAT devienne une constante, appuyez sur F2 (MODIFIE) puis F9 (CALCUL).

Pour générer des nombres aléatoires compris dans un autre intervalle que l'intervalle 0-1, multipliez @ALEAT par la taille de l'intervalle. Utilisez @ARRONDI ou @ENT pour obtenir des nombres entiers aléatoires.

Exemples

@ALEAT = 0,419501 ou toute valeur comprise entre 0 et 1.

@ALEAT*10 = 6,933674 ou toute valeur comprise entre 0 et 10 inclus.

@ENT(@ALEAT*50)+1 = 49, ou tout nombre entier compris entre 1 et 50.

@RANG

@RANG(*nombre*; *champ*; [*ordre*]) renvoie la position relative de *nombre* par rapport aux autres nombres de *champ* classés dans l'ordre spécifié par *ordre*.

Arguments

nombre est le nombre dont vous voulez connaître la position.

champ est l'adresse ou le nom du champ qui contient l'ensemble des nombres. *nombre* doit être l'une des valeurs de *champ*.

ordre est un argument facultatif qui spécifie si *champ* doit être examiné dans l'ordre décroissant ou croissant ; il peut prendre l'une des valeurs suivantes :

<u>ordre</u>	<u>1-2-3 considère les nombres de <i>champ</i> comme s'ils étaient classés par</u>
0	Ordre décroissant (9 à 1) ; paramètre par défaut si vous omettez cet argument
1	Ordre croissant (1 à 9)

Remarques

1-2-3 assigne le même rang aux nombres trouvés plusieurs fois dans *champ*. La présence de doublons affecte le rang des nombres classés ensuite. Par exemple, dans la suite de nombres 2, 4, 6, 8, 8, 10, 12, le nombre 8 apparaît deux fois et occupe le 4ème rang dans l'ordre croissant; comme deux valeurs se partagent le 4ème rang, aucune n'occupe le 5ème et le nombre suivant dans l'ordre croissant, 10, est classé au 6ème rang.

Exemples

Le champ nommé VENTES (A1..A5) contient les valeurs suivantes :

5.000 F

4.900 F

5.150 F

4.800 F

4.900 F

@RANG(4900;VENTES) = 3 ; 4.900 F est la 3ème plus grande valeur dans le champ VENTES. Il n'y a pas de valeur de rang 4.

@RANG(4900;VENTES;1) = 2 ; 4.900 F est la 2ème plus petite valeur parce que VENTES est examiné dans l'ordre croissant. Il n'y a pas de valeur de rang 3.

@TAUX

@TAUX(*valeur-future*; *valeur-actuelle*; *périodes*) calcule le taux d'intérêt périodique nécessaire pour qu'un investissement (*valeur-actuelle*) atteigne la *valeur-future* spécifiée, sur un nombre de *périodes* donné.

Arguments

Les arguments *valeur-future*, *valeur-actuelle* et *périodes* sont des nombres, ou les noms ou adresses des cellules contenant ces nombres.

Emploi

La fonction financière **@TAUX** permet d'évaluer un investissement ou de comparer plusieurs placements. Il est ainsi possible de calculer le taux d'intérêt composé d'un investissement (la valeur d'une action par exemple), ou d'étudier un placement en connaissant la valeur initiale, la valeur finale et le temps écoulé entre les deux.

Exemples

Vous avez acquis un titre de 10.000 F. Sachant qu'à son échéance, fixée à 5 ans, ce titre vaudra 18.000 F et que les intérêts sont calculés mensuellement, vous souhaitez connaître le taux d'intérêt périodique de votre investissement.

@TAUX(18000;10000;60) donne 0,984 % comme taux d'intérêt mensuel. Pour obtenir le taux annuel, appliquez la formule $((1+\text{@TAUX}(18000;10000;60))^{12})-1$. Le résultat est 12,47 %.

Fonction similaire

@TAUXBIS calcule le taux d'intérêt périodique nécessaire pour qu'un investissement atteigne une valeur future.

@CONVREF

@CONVREF(*référence*) renvoie le nombre entre 1 et 256 correspondant à la lettre de A à IV entrée en *référence* ou, inversement, si un nombre entre 1 et 256 est entré en *référence*, renvoie la lettre désignant la colonne ou la feuille de travail correspondante.

Arguments

référence désigne une colonne ou une feuille de travail et peut être soit une lettre de A à IV, soit un entier entre 1 et 256.

@CONVREF ne distingue pas les majuscules des minuscules ; vous pouvez entrer une référence en lettres majuscules ou minuscules.

Exemples

@CONVREF(10) = J

@CONVREF("J") = 10

Fonctions similaires

@COL compte le nombre de colonnes dans un champ et @FEUILLES compte le nombre de feuilles de travail dans un champ. @COORD renvoie une adresse de cellule à partir des paramètres que vous spécifiez.

@REGRESSION

@REGRESSION(*champ-x*;*champ-y*;*paramètre*;*[origine]*) effectue une régression linéaire multiple et renvoie un des paramètres ainsi calculés.

Arguments

champ-x est un champ d'au plus 75 colonnes sur 8192 lignes qui contient les valeurs des variables indépendantes.

champ-y est un champ mono-colonne représentant les valeurs de la variable dépendante. *champ-y* doit obligatoirement avoir le même nombre de lignes que *champ-x*.

paramètre est un nombre qui détermine l'élément que renvoie la fonction @REGRESSION. Il peut prendre les valeurs suivantes :

<u>paramètre</u>	<u>1-2-3 renvoie</u>
1	Constante
2	Erreur standard concernant l'estimation Y
3	R au carré
4	Nombre d'observations
5	Degrés de liberté
101 to 175	Coefficient de X (pente de la droite de régression) associé à l'une des variables indépendantes spécifiée par <i>paramètre</i>
201 to 275	Erreur standard concernant l'estimation de la pente associée à l'une des variables indépendantes spécifiée par <i>paramètre</i>

Pour déterminer la variable indépendante à prendre en compte lors du calcul de la pente ou de l'erreur standard sur la pente, c'est-à-dire lorsque *paramètre* est compris entre 101 et 175 ou entre 201 et 275, 1-2-3 numérote les variables colonne par colonne. La variable indépendante représentée par la colonne la plus à gauche dans *champ-x* portera le numéro 1, etc.

Si vous souhaitez, par exemple, connaître la pente de la droite de régression associée à la troisième colonne de *champ-x*, vous affecterez la valeur 103 à *paramètre* (100 + 3).

De même, si vous souhaitez déterminer l'erreur standard concernant l'estimation du coefficient de la cinquième variable indépendante (5ème colonne), vous affecterez la valeur 205 à *paramètre* (200 + 5).

origine est un argument facultatif qui précise si la droite de régression passe par l'origine ou, au contraire, si 1-2-3 doit calculer la constante (valeur de Y à l'origine).

<u>Si origine vaut</u>	<u>1-2-3 considère que</u>
0	La droite passe par l'origine
1	La valeur à l'origine doit être calculée ; paramètre par défaut si vous omettez l'argument

Remarques

Pour des valeurs identiques, @REGRESSION et Champ Outils mathématiques Régression renvoient les mêmes résultats.

Exemples

@REGRESSION

Fonctions @ similaires

@PREVISIONT renvoie une valeur prévisionnelle pour x basée sur la tendance linéaire entre les valeurs du *champ-y* et du *champ-x*.

@REPETE

@REPETE(*chaîne*; *n*) répète la *chaîne* le nombre de fois indiqué par *n*.

Arguments

L'argument *chaîne* peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

n est un nombre entier positif, ou le nom ou l'adresse d'une cellule comportant un nombre entier positif.

Emploi

La fonction de chaîne @REPETE permet de répéter un caractère quelconque, y compris les symboles mathématiques et graphiques et les caractères étrangers.

Remarque

La fonction @REPETE reproduit une chaîne autant de fois que vous le demandez (sans que cette répétition ne soit limitée par la taille de la colonne), tandis qu'un libellé répété avec le préfixe de répétition \ (barre oblique inverse) ne peut dépasser la largeur de la colonne actuelle.

Exemples

@REPETE("Bonjour ";3) renvoie Bonjour Bonjour Bonjour

@REPETE("-";10) donne -----.

@SUBSTITUE

@SUBSTITUE(*chaîne-origine*; *position*; *n*; *nouvelle-chaîne*) remplace les *n* caractères de l'argument *chaîne-origine* par ceux figurant dans *nouvelle-chaîne*, en partant de la *position* spécifiée.

Arguments

Les arguments *chaîne-origine* et *nouvelle-chaîne* peuvent être du texte mis entre guillemets, des formules texte, ou le nom ou l'adresse de cellules comportant des libellés ou des formules texte. L'argument *position*, numéro de position d'un caractère dans *chaîne-origine*, est un nombre positif ou nul, ou le nom ou l'adresse d'une cellule contenant un tel nombre. Si *position* est supérieur à la longueur de *chaîne-origine*, @SUBSTITUE ajoute *nouvelle-chaîne* à *chaîne-origine*.

n peut être un nombre entier positif ou nul, ou le nom ou l'adresse d'une cellule contenant un tel nombre. Si *n* est égal à 0, @SUBSTITUE insère *nouvelle-chaîne* à la *position* spécifiée sans supprimer de caractères dans *chaîne-origine*.

Emploi

Utilisez @CHERCHE pour déterminer la *position* d'une chaîne dans une autre et reprenez ce résultat avec @SUBSTITUE pour remplacer cette chaîne par une autre.

@SUBSTITUE permet de remplacer un jeu de caractères par un autre (pour modifier des informations dans une base de données, par exemple).

Remarque

@SUBSTITUE compte tous les caractères d'une chaîne, y compris les espaces et les signes de ponctuation. Aussi, pensez à inclure les espaces nécessaires lorsque vous utilisez @SUBSTITUE pour ajouter ou insérer des chaînes.

Exemple

@SUBSTITUE(SECTION;@CHERCHE("-";SECTION;0);1;"/") transforme 4-24, libellé figurant dans SECTION, en 4/24.

@FIN

@FIN(*chaîne*;n) donne les *n* derniers caractères d'une *chaîne*.

Arguments

L'argument *chaîne* est du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

n peut être un nombre entier positif ou nul, ou le nom ou l'adresse d'une cellule contenant ce type de nombre. Si *n* est égal à 0, le résultat est une chaîne vide. Si *n* est supérieur au nombre de caractères de *chaîne*, @FIN affiche l'intégralité de la chaîne.

Emploi

La fonction de chaîne @FIN permet de ne copier dans une autre cellule qu'une partie d'un libellé (pour extraire des prénoms dans des libellés composés de noms et de prénoms, par exemple). Intégrée à une macro, @FIN peut servir à extraire une partie de libellé pour la placer dans une base de données, l'utiliser dans des appels de sous-routines, ou pour modifier la macro proprement dite. Associez @CHERCHE à @FIN lorsque vous ignorez la valeur exacte de *n*, ou que *n* est variable.

Remarque

@FIN compte les signes de ponctuation et les espaces comme des caractères.

Exemple

@FIN(B3;7) = janvier si B3 contient le libellé Ventes de janvier.

Fonctions similaires

@DEBUT extrait les *n* premiers caractères d'une *chaîne*. @MILIEU extrait des caractères d'une *chaîne*.

@ARRMUL

@ARRMUL(x ; $multiple$; [$direction$]) arrondit le nombre x au $multiple$ supérieur ou inférieur le plus proche.

Arguments

x et $multiple$ peuvent être n'importe quels nombres de même signe.

$direction$ est un argument facultatif qui précise si x doit être arrondi par défaut ou par excès.

<u>$direction$</u>	<u>1-2-3 arrondi x</u>
<u>n</u>	
1	Par excès
0	Au $multiple$ le plus proche ; valeur par défaut si vous omettez l'argument
-1	Par défaut

Exemples

@ARRMUL(25,037;0,05;1) = 25,40

@ARRMUL(25,37;0,5;-1) = 25,35

Fonctions similaires

@ENT tronque un nombre à sa partie entière. @ARRONDI, @ARRDEF et @ARREXC arrondissent un nombre au nombre de décimales spécifié. @PAIR arrondit un nombre au nombre pair supérieur ou inférieur. @IMPAIR arrondit un nombre au nombre impair supérieur ou inférieur. @TRONQUE tronque un nombre au nombre de décimales spécifié.

@ARRONDI

@ARRDEF

@ARREXC

@ARRONDI($x;n$) renvoie le produit de x par 10^n arrondi à l'entier le plus proche, puis divisé par 10^n .

@ARRDEF($x;n$;*direction*) renvoie le produit de x par 10^n arrondi à l'entier le plus proche par défaut, puis divisé par 10^n .

@ARREXC($x;n$;*direction*) renvoie le produit de x par 10^n arrondi à l'entier le plus proche par excès, puis divisé par 10^n .

Arguments

x est un nombre.

n est un nombre entre -100 et 100. Pour @ARRDEF et @ARREXC, 1-2-3 utilise 0 si vous omettez l'argument n .

direction indique comment arrondir les valeurs négatives (argument facultatif). *direction* peut prendre la valeur 0 ou 1.

- Pour @ARREXC : Si *direction* vaut 0, 1-2-3 arrondit une valeur négative par excès. En revanche, si *direction* est égal à 1, 1-2-3 arrondit la valeur par défaut.
- Pour @ARRDEF : Si *direction* vaut 0, 1-2-3 arrondit une valeur négative par défaut. En revanche, si *direction* est égal à 1, 1-2-3 arrondit la valeur par excès.

Si vous omettez l'argument *direction*, 1-2-3 utilise 0. Lorsque x est positif, *direction* est sans effet.

Remarques

Si n est	@ARRONDI
Positif	Affecte la partie décimale de x . Par exemple, si $n = 2$, x sera arrondi au centième le plus proche.
Négatif	Affecte la partie entière de x . Par exemple, si $n = -2$, x sera arrondi à la centaine la plus proche.
0	Arrondit x à l'entier le plus proche.

Remarques

Utilisez le format Fixe pour spécifier le nombre de décimales à afficher tout en laissant 1-2-3 continuer à utiliser les décimales non affichées ; n'utilisez pas @ARRONDI car 1-2-3 travaillerait alors avec les valeurs arrondies.

Exemples

@ARRONDI(134,578;2) = 134,58

@ARRONDI(134,578;0) = 135

@ARRONDI(134,578;-2) = 100

@ARRDEF(134,578;2) = 134,57

@ARRDEF(134,578;0) = 134

@ARRDEF(134,578;-2) = 100

@ARREXC(134,578;2) = 134,58

@ARREXC(134,578;0) = 135

@ARREXC(134,578;-2) = 200

Fonctions similaires

@ARRMUL arrondit un nombre au multiple spécifié. @PAIR arrondit un nombre au nombre pair le plus proche.

@IMPAIR arrondit un nombre au nombre impair le plus proche. @ENT tronque un nombre à sa partie entière.

@TRONQUE tronque un nombre au nombre de décimales spécifié.

@LIGNES

@LIGNES(*champ*) compte le nombre de lignes figurant dans l'argument *champ*.

Argument

L'argument *champ* est un nom ou une adresse de champ.

Emploi

La fonction spéciale @LIGNES permet de déterminer une valeur dépendant du nombre de lignes. Utilisez par exemple @LIGNES comme argument de l'instruction {POUR} pour déterminer le nombre d'exécutions d'une routine qui effectue la même action sur toutes les lignes d'un champ.

Exemples

@LIGNES(A3..B7) = 5 (lignes 3 à 7).

@LIGNES(RESULTAT) = 43 si RESULTAT correspond au champ B3..B45.

Fonctions similaires

@COL et @FEUILLES renvoient respectivement le nombre de colonnes et le nombre de feuilles d'un champ.

Exemple de fonction @STD et @STDE

Le tableau ci-dessous dresse la liste des poids et tailles de dix personnes. Vous voulez déterminer l'écart-type de leur taille.

@STD(A2..A11) = 5,793483

S'il s'agit d'une dizaine de personnes choisies au hasard dans un groupe d'individus, faites appel à la formule suivante :

@STDE(A2..A11) = 6,106868

A	-----	A	-----	B	--
1		TAILLE (cm)		POIDS (kg)	
2		190,50		72,73	
3		187,96		86,36	
4		175,26		68,18	
5		175,26		76,37	
6		180,34		77,27	
7		180,34		72,73	
8		187,96		75,00	
9		172,72		68,18	
10		177,80		70,46	
11		179,07		86,36	

@C

@C(*champ*) convertit en libellé le contenu de la première cellule de *champ*.

Argument

L'argument *champ* est une adresse de cellule ou un nom de champ.

Emploi

La fonction de chaîne @C peut être associée à toute autre fonction de chaîne ou formule texte pour s'assurer de la présence d'un libellé dans une cellule qui pourrait comporter une valeur. @C permet également d'éviter que des formules texte n'entraînent l'affichage de ERR (ainsi, +A1&A2 donne ERR si l'une de ces cellules contient une valeur numérique).

@C sert aussi à vérifier les entrées effectuées par l'utilisateur pendant l'exécution d'une macro. Faites appel à cette fonction dans les routines de recherche d'erreurs pour éviter les erreurs survenant lorsqu'une cellule utilisée dans une formule comporte un type de données incorrect.

Exemple

Dans la macro,

```
{SI @C(B6)=""}{BIP}{INDIQUE "LIBELLE UNIQUEMENT"}
```

@C donne une chaîne vide si la cellule B6 contient une valeur ou est vierge; 1-2-3 émet alors un signal sonore et modifie l'indicateur de mode de façon à ce qu'il affiche la mention LIBELLE UNIQUEMENT.

Fonctions similaires

@N convertit en nombre l'entrée de la première cellule d'un *champ*. @ESTCHAINE permet de vérifier si une cellule contient un libellé.

@SEC

@SEC(z) renvoie la sécante d'un angle z. Dans un triangle rectangle, la sécante d'un angle est le rapport entre l'hypoténuse et le côté adjacent à cet angle. La sécante est la fonction réciproque du cosinus.

Arguments

L'angle x est mesuré en radians. x peut être une valeur comprise entre $-2^{\wedge}63^*$ et $2^{\wedge}63^*$ inclus.

Exemples

@SEC(@DEGENRAD(30)) = 1,154701 qui est la sécante d'un angle de 30 degrés.

Fonctions similaires

@ASEC calcule l'arc sécante et @ASECH, l'arc sécante hyperbolique d'une valeur. @COS renvoie le cosinus d'un angle et @SECH, la sécante hyperbolique.

@SECH

@SECH(z) renvoie la sécante hyperbolique d'un angle z. La sécante hyperbolique est la fonction réciproque du cosinus hyperbolique. La valeur renvoyée par @SECH est supérieure à 0 ou inférieure ou égale à 1.

Arguments

L'angle x, mesuré en radians, est une valeur comprise entre -709.7827 et 709.7827 environ.

Exemples

@SECH(@DEGENRAD(30)) = 0,87701 qui est la sécante hyperbolique d'un angle de 30 degrés.

Fonctions similaires

@ASECH calcule la sécante hyperbolique inverse et @SEC, la sécante d'un angle.

@SECONDE

@SECONDE(*nombre-temps*) extrait les secondes (nombre entier compris entre 0 et 59) de l'argument *nombre-temps*.

Argument

L'argument *nombre-temps* est une valeur comprise entre 0,000000 (minuit) et 0,999988 (23:59:59 ou 11:59:59 PM), ou le nom ou l'adresse d'une cellule contenant cette valeur.

Emploi

La fonction d'heure **@SECONDE** permet d'extraire uniquement les secondes des nombres générés par les fonctions **@ACTUEL**, **@TEMPS** ou **@HEURVAL**. Cette donnée peut être utile dans des applications spécifiques (franchissement de la barre des 20 secondes départ arrêté pour une distance donnée, par exemple).

Exemples

@SECONDE(0,333) = 31.

@SECONDE(**@TEMPS**(11;15;45)) = 45, puisque cette valeur correspond à l'argument *secondes* de **@TEMPS**(11;15;45).

Fonctions similaires

@HEURE extrait l'heure, et **@MINUTE**, les minutes d'un nombre temps.

@SOMMESERIE

@SOMMESERIE($x;n;m;coefficients$) calcule la somme d'une série de puissances.

Arguments

x est la variable de base.

n est le premier exposant.

m est le nombre à ajouter à n pour chacun des exposants suivants.

x , n , et m sont des nombres quelconques.

coefficients est un champ qui contient la liste des coefficients multiplicateurs de la puissance de x . Le nombre de coefficients contenus dans ce champ détermine le nombre de puissances de x à additionner. Par exemple, si le champ *coefficients* contient 10 valeurs, la série sera composée de 10 puissances de x .

Exemples

Un champ nommé COEFF contient les valeurs suivantes : 0,2; 0,7; 1,3.

@SOMMESERIE(3,5;2;1;COEFF) = 227,5438

Ce qui équivaut à écrire : $0,2*3,5^2+0,7*3,5^3+1,3*3,5^4$

@FEUILLES

@FEUILLES(*champ*) renvoie le nombre de feuilles dans *champ*.

Argument

L'argument *champ* est une adresse ou un nom de champ.

Emploi

La fonction spéciale @FEUILLES sert à calculer une valeur dépendant du nombre de feuilles. Utilisez @FEUILLES avec {POUR} dans une macro qui répète la même action sur une série de feuilles, afin de déterminer l'arrêt de la macro.

Exemple

@FEUILLES(T_2) = 4 si T_2 correspond au champ B:B3..E:C45 (feuilles B, C, D et E).

Fonctions similaires

@COL et @LIGNES renvoient respectivement le nombre de colonnes et le nombre de lignes d'un champ.

@CONVREF convertit les lettres de colonne 1-2-3 A à IV (ou les lettres de feuille) en nombres de 1 à 256.

@SIGNE

@SIGNE(x) retourne 1 si x est un nombre positif, 0 si x est nul, -1 si x est un nombre négatif.

Arguments

x est un nombre quelconque.

Exemples

@SIGNE(15) = 1

@SIGNE(15*0) = 0

@SIGNE(-15) = -1

@SIN

@SIN(z) calcule le sinus d'un angle z (dans un triangle rectangle, le sinus d'un angle est le rapport entre le côté opposé à cet angle et l'hypoténuse).

Argument

L'angle x est mesuré en radians. x peut être une valeur comprise entre $-2^{\wedge}63^*$ et $2^{\wedge}63^*$ inclus.

Exemple

@SIN(@DEGENRAD(30)) = 0,5 qui est le sinus d'un angle de 30 degrés.

Fonction similaire

@SINH calcule le sinus hyperbolique d'un angle. @ASINH calcule l'arc sinus hyperbolique d'un angle à partir de son sinus hyperbolique.

@SINH

@SINH(z) calcule le sinus hyperbolique d'un angle z.

Arguments

x peut prendre n'importe quelle valeur entre environ -709,7827 et 709,7827.

Exemples

@SINH(@DEGRAD(30)) = 0,547853, qui est le sinus hyperbolique d'un angle de 30 degrés.

Fonctions similaires

@ASINH renvoie l'arc sinus hyperbolique (sinus hyperbolique inverse) et @SIN, le sinus d'un angle.

@SYMDIST

@SYMDIST(*champ*;*[type]*) évalue la symétrie des valeurs de *champ* par rapport à la moyenne.

Arguments

champ est le nom ou l'adresse d'un champ contenant des valeurs. Si *champ* contient moins de trois valeurs, la fonction @SYMDIST renvoie ERR.

type est un argument facultatif qui précise la nature de *champ* et donc, le type de résultat renvoyé par la fonction @SYMDIST.

<u>type</u>	<u>1-2-3 renvoie</u>
0	Symétrie par rapport à la moyenne d'une population ; valeur par défaut si vous omettez l'argument
1	Symétrie par rapport à la moyenne d'un échantillon

Remarques

Une valeur positive de @SYMDIST indique une tendance des valeurs vers la gauche (en dessous de la moyenne), une valeur négative, une tendance à droite.

Exemples

Le champ DONNEES contient les valeurs suivantes: 2, 5, 5, 9, 1, 2, 4

@SYMDIST(DONNEES) = 0,584816

Fonctions similaires

@KURTOSIS renvoie le Kurtosis des valeurs d'un champ. @STD et @STDNUM calculent l'écart-type d'une liste de valeurs, @VAR et @VARE, sa variance.

@AMORLIN

@AMORLIN(*coût; valeur-récupération; durée*) calcule l'amortissement linéaire d'une immobilisation, en tenant compte du *coût* initial du bien, de sa *durée* présumée d'utilisation et de sa *valeur de récupération* finale.

Arguments

L'argument *coût* représente le montant payé pour l'immobilisation. *coût* peut être une valeur quelconque, ou le nom ou l'adresse d'une cellule contenant cette valeur. L'argument *valeur-récupération* correspond à la valeur de récupération de l'immobilisation à la fin de sa durée d'utilisation. *valeur-récupération* peut être une valeur quelconque, ou le nom ou l'adresse d'une cellule comportant cette valeur. L'argument *durée* indique le nombre de périodes nécessaires pour atteindre la valeur de récupération. *durée* est une valeur différente de 0, ou le nom ou l'adresse d'une cellule contenant cette valeur.

Emploi

La fonction financière **@AMORLIN** sert à calculer l'amortissement d'une immobilisation, lorsqu'il n'est pas nécessaire d'appliquer un amortissement accéléré.

Remarques

La méthode d'amortissement linéaire présente une proportionnalité constante par rapport à l'unité de temps choisie (généralement une année). La dotation est linéaire et il est imputé à chaque année un montant égal au quotient de l'assiette de l'amortissement (la somme à amortir) par le nombre d'années correspondant à la durée probable d'utilisation du bien en cause, ce qui donne une annuité d'amortissement identique d'une année à l'autre.

Exemple

Vous avez acquis un équipement de bureau de 50.000 F. Sachant que sa durée d'utilisation est de quatre ans et que sa valeur de récupération au terme de ces années sera de 5.000 F, vous voulez calculer la charge d'amortissement annuelle, selon la méthode de l'amortissement linéaire. **@AMORLIN**(50000;5000;4) donne une annuité d'amortissement de 11.250 F.

Fonctions similaires

@AMORTDEGTC calcule l'amortissement selon la méthode de l'amortissement dégressif à taux constant.

@AMORDEG calcule l'amortissement selon la méthode de l'amortissement dégressif à taux double. **@AMDEGVAR** met en oeuvre la méthode de l'amortissement dégressif à taux double/à taux variable et **@AMORANN**, celle de l'amortissement proportionnel à l'ordre numérique inversé des années.

@PETIT

`@PETIT(champ;n)` renvoie la *nième* plus petite valeur trouvée dans *champ*.

Arguments

champ est le nom ou l'adresse du champ où se fait la recherche.

n peut être n'importe quel nombre positif entier. Si *n* est supérieur au nombre de valeurs dans *champ*, `@PETIT` renvoie ND.

Exemples

Le champ RESULTATS contient les nombres : 87, 85, 90, 80, 82, 92, 79, 85, 95, 86.

`@PETIT(RESULTATS;3)` renvoie 82, la troisième plus petite valeur de la liste.

Fonctions similaires

`@GRAND` renvoie la *nième* plus grande valeur trouvée dans un champ. `@MAX` et `@MAXNUM` renvoient la plus grande valeur. `@MIN` et `@MINNUM` renvoient la plus petite.

@RACINE

@RACINE(x) calcule la racine carrée de x.

Argument

x est un nombre positif, ou le nom ou l'adresse d'une cellule contenant ce nombre.

Emploi

Utilisez la fonction mathématique @RACINE pour calculer la racine carrée d'un nombre. Associez-la à @ABS pour vous assurer que la valeur absolue de x (éventuellement négatif) sera utilisée en entrant la formule @RACINE(@ABS(x)).

Exemples

@RACINE(B3) = 10, si B3 contient la valeur 100.

@RACINE(-2) = ERR car x est négatif.

@RACINEPI

@RACINEPI(x) calcule la racine carrée de x multiplié par .

Arguments

x est un nombre positif ou nul.

Exemples

@RACINEPI(0,5) = 1,253314

@RACINEPI(2) = 2,506628

Fonctions similaires

@RACINE calcule la racine carrée positive d'un nombre. @PI renvoie le nombre .

@STD, @STDE, @STDNUM, @STDENUM

@STD(*liste*) calcule l'écart-type des valeurs figurant dans *liste*.

@STDE(*liste*) calcule l'écart-type des valeurs d'échantillon figurant dans l'argument *liste*.

@STDNUM(*liste*) calcule l'écart-type des valeurs figurant dans *liste*, sans tenir compte des cellules qui contiennent des libellés.

@STDENUM(*liste*) calcule l'écart-type des valeurs d'échantillon figurant dans l'argument *liste*, sans tenir compte des cellules qui contiennent des libellés.

Arguments

L'argument *liste* peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Remarques

Pour calculer l'écart-type, **@STD** et **@STDNUM** font appel à la méthode n (population), qui suppose que les valeurs de *liste* représentent la totalité de la population. Si seules certaines valeurs ont été sélectionnées, l'écart-type est biaisé, en raison des erreurs induites par l'échantillonnage.

@STDE et **@STDENUM** font appel à la méthode n-1 (échantillon) pour calculer l'écart-type des valeurs d'échantillon. Cette méthode donne un résultat légèrement plus élevé que celui de la méthode n (population) afin de compenser les erreurs inhérentes à l'échantillonnage. Un écart-type plus important n'est pas biaisé par ces erreurs, ce qui garantit une plus grande précision.

L'écart-type est égal à la racine carrée de la variance (ou fluctuation).

Exemples

@STD et @STDE

Fonctions similaires

@BSTD et @BSTDE calculent l'écart-type des valeurs d'une rubrique de table de base de données. @VAR et @VARNUM calculent la variance d'une population. @VARE et @VARENUM calculent la variance d'un échantillon.

@CHAINE

@CHAINE(x;n) convertit le nombre x en un libellé au format défini par n.

Arguments

x est un nombre, ou le nom ou l'adresse d'une cellule contenant un nombre.

n est un nombre entier

<u>Valeur de n</u>	<u>Format renvoyé</u>
0 à 116	<u>Fixe</u> , avec n décimales
1000 à 1116	<u>Milliers</u> , avec n-1000 décimales
-18 à -1	<u>Scientifique</u> , avec des nombres <u>@Absolu(nombre)</u>
10001 à 10512	<u>Libre</u> , jusqu'à n-10000 caractères

Remarques

@CHAINE ne tient pas compte des caractères de formatage éventuellement utilisés pour afficher le nombre x, et notamment des symboles monétaires ou de ponctuation introduits au moyen d'un format des nombres. Par exemple, si la cellule A7 contient 45,23 F, @CHAINE(A7;2) retournera le libellé 45,23.

Exemples

@CHAINE(203;3) = le libellé 203,000

@CHAINE(1,23587;0) = le libellé 1

@CHAINE(20500;1002) = le libellé 20.500,00

@STRING(@PI;-5) = le libellé 3,1416E+000

@STRING(123456,789;10008) = le libellé 12E+005

Fonction similaire

@VALEUR convertit en valeur un nombre entré sous forme de libellé.

@SOUSTOTAL

@SOUSTOTAL(*liste*) fait la somme des valeurs contenues dans *liste*. **@SOUSTOTAL** fonctionne exactement comme **@SOMME** et n'existe que pour indiquer à 1-2-3 quelles cellules **@TOTALGEN** doit prendre en compte.

Arguments

liste peut contenir une combinaison quelconque des arguments suivants : nombres, formules numériques, adresses ou noms de champ contenant des nombres ou des formules numériques. Ces arguments devant être séparés par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

@SOUSTOTAL(B5..B9) renvoie la somme des valeurs en B5..B9.

@SOUSTOTAL(VENTES;M25..R25) renvoie la somme des valeurs dans le champ VENTES et dans le champ M25..R25.

Fonction similaire

@SOMME additionne les valeurs d'une liste. **@SOMMENEGATIFS** n'additionne que les valeurs négatives d'une liste. **@SOMMEPOSITIFS** n'additionne que les valeurs positives d'une liste.

@SOMME

@SOMME(*liste*) additionne les valeurs de *liste*.

Argument

L'argument *liste* peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

@SOMME(B5..B9) effectue la somme des valeurs figurant dans le champ B5..B9.

@SOMME(VENTES;M25..R25) additionne les valeurs des champs VENTES et M25..R25.

Icônes SmartIcons associées



Additionne les valeurs du champ contrasté ou adjacent (à condition de le délimiter par des cellules vides).

Fonction similaire

@BSOMME totalise les valeurs répondant aux critères spécifiés.

@NSOMME ajoute chaque *nième* valeur d'une liste, en commençant au *début*.

@SOUSTOTAL permet de constituer des sous-totaux, destinés à être additionnés au moyen de @TOTALGEN.

@SOMMENEGATIFS n'additionne que les valeurs négatives d'une liste. @SOMMEPOSITIFS n'additionne que les valeurs positives d'une liste.

@SOMMEPROD

@SOMMEPROD(*liste*) multiplie les valeurs situées dans les cellules correspondantes (même colonne, même ligne, même feuille) d'une série de champs, puis additionne les résultats obtenus.

Argument

L'argument *liste* est une série de champs contenant des valeurs, de même taille (même nombre de cellules) et de même forme. Si les champs de *liste* ne satisfont pas cette condition, **@SOMMEPROD** génère la valeur ERR.

Voir aussi Arguments des fonctions @ statistiques.

Remarque

Si les champs de *liste* sont des champs d'une seule colonne, **@SOMMEPROD** multiplie les cellules d'une même ligne. Si ce sont des champs d'une seule ligne, **@SOMMEPROD** multiplie les cellules d'une même colonne.

Exemples

Dans cet exemple, issu d'une table de base de données immobilières, **@SOMMEPROD** totalise les commissions perçues par une agence immobilière sur les logements dans le mois en cours.

liste comprend les deux champs VENDU (D4..D6) et TAUX_COM (E4..E6). Le premier contient les prix de ventes des logements et le deuxième, les commissions perçues (sous forme de pourcentage) par l'agence.

VENDU	TAUX_COM
100000	0,04
200000	0,04
600000	0,05

@SOMMEPROD(VENDU;TAUX_COM) = 42000, total des commissions (4000 + 8000 + 30000) perçues par l'agence.

Fonctions similaires

@SOMMECAR calcule la somme des carrés des valeurs contenues dans une liste. **SOMMEXMY2** soustrait les valeurs contenues dans les cellules correspondantes de deux champs, élève au carré les résultats, puis les additionne.

@SOMMECAR

@SOMMECAR(*liste*) renvoie la somme des carrés des valeurs contenues dans *liste*.

Arguments

L'argument *liste* peut comporter n'importe quelle combinaison des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

@SOMMECAR(2;4;6) = 56

Fonctions similaires

@SOMME additionne les valeurs d'une liste. @SOMMEPROD multiplie les valeurs des cellules correspondantes de plusieurs champs et additionne les résultats obtenus.

@SOMMEXMY2

@SOMMEXMY2(*champX*;*champY*) soustrait les valeurs contenues dans le *champY* des valeurs contenues dans les cellules correspondantes (même colonne, même ligne, même feuille) de *champX*, élève au carré les résultats, puis les additionne.

Arguments

champX et *champY* doivent contenir des valeurs (ou des cellules vierges comptées pour 0), avoir la même taille (même nombre de cellules) et la même forme, sinon **@SOMMEXMY2** retourne ERR.

Remarques

Si *champX* et *champY* sont des champs d'une seule colonne, 1-2-3 soustrait les valeurs d'une même ligne. Si *champX* et *champY* sont des champs d'une seule ligne, 1-2-3 soustrait les valeurs d'une même colonne.

Exemples

Dans l'exemple suivant, *champX* est nommé UN, *champY* est nommé DEUX :

UN	DEUX
5	3
4	4
7	8

@SOMMEXMY2(UN;DEUX) = 5

Fonctions similaires

@SOMMEPROD multiplie le contenu des cellules correspondantes dans plusieurs champs et en effectue la somme.

@SOMMECAR calcule la somme des carrés des valeurs contenues dans une liste.

@AMORANN

@AMORANN(*coût*; *valeur-récupération*; *durée*; *période*) calcule, pour une *période* donnée, la charge d'amortissement d'une immobilisation en appliquant la méthode de l'amortissement proportionnel à l'ordre numérique inversé des années. Ce calcul tient compte de la valeur initiale du bien (*coût*), de sa *durée* probable d'utilisation et de sa *valeur-récupération* finale.

Arguments

L'argument *coût* est le montant payé pour l'immobilisation. *coût* peut être une valeur quelconque. L'argument *valeur-récupération* est la valeur du bien au terme de sa durée d'utilisation. *valeur-récupération* peut être une valeur quelconque. L'argument *durée* indique le nombre de périodes nécessaires pour atteindre la valeur de récupération. *durée* doit être une valeur supérieure ou égale à 1. L'argument *période* correspond à la période pour laquelle vous voulez calculer le montant de l'amortissement. *période* doit être une valeur supérieure ou égale à 1.

Emploi

La méthode appliquée par cette fonction entraîne un amortissement accéléré, plus marqué dans les premières périodes (quoique moins sensible qu'avec la méthode de l'amortissement dégressif à taux double). L'amortissement correspond au coût réel moins la valeur de récupération. La fonction financière **@AMORANN** autorise un amortissement plus marqué dans les premières périodes.

Exemple

Vous venez d'acquérir un équipement de bureau de 50.000 F. Sachant que sa durée d'utilisation est de quatre ans et que sa valeur de récupération au terme de ces quatre années sera de 5.000 F, vous souhaitez connaître le montant de l'annuité d'amortissement pour la quatrième année, en appliquant la méthode de l'amortissement proportionnel à l'ordre numérique inversé des années. **@AMORANN**(50000;5000;4;4) donne 4.500 F comme charge d'amortissement pour la quatrième année.

Fonctions similaires

@AMORDEG calcule l'amortissement selon la méthode de l'amortissement dégressif à taux double. **@AMDEGVAR** applique un amortissement dégressif à taux double ou à taux variable, tandis que **@AMORLIN** fait appel à la méthode linéaire.

Exemple de fonctions @VAR et @VARE

Le tableau ci-dessous dresse la liste des poids et tailles de dix personnes. Vous voulez déterminer la variance de leur poids :

@VAR(B2..B11) = 38,7462

Pour une dizaine de personnes choisies au hasard dans un groupe d'individus, faites appel à la formule :

@VARE(B2..B11) = 43,8292

A	-----	A	-----	B	--
1		TAILLE (cm)		POIDS (kg)	
2		190,50		72,73	
3		187,96		86,36	
4		175,26		68,18	
5		175,26		76,37	
6		180,34		77,27	
7		180,34		72,73	
8		187,96		75,00	
9		172,72		68,18	
10		177,80		70,46	
11		179,07		86,36	

Exemple de fonction @CONSV

Dans l'exemple suivant, la table d'imposition TABLEAU_IMP (A3..E11) donne la liste des prélèvements d'impôts déterminés en fonction du revenu et de la tranche d'imposition. La formule @CONSV(35329;TABLEAU_IMP;1), entrée dans une cellule dotée du format Monnaie sans décimale, donne la valeur 9 351 F, soit le montant de l'impôt à payer pour un revenu de 35 329 F et une tranche d'imposition 1. Ce montant correspond au chiffre qui se rapproche le plus de 35 329 F, sans lui être supérieur.

A ----- A -----	B -----	C -----	D -----	E -----
----- TRANCHES D'IMPOSITION -----				
Revenu (en F) >=	1	2	3	4
35 000,00	9 219,00	7 265,00	11 315,00	5 31,00
35 050,00	9 241,00	7 282,00	11 340,00	8 55,00
35 100,00	9 263,00	7 299,00	11 363,00	8 73,00
35 150,00	9 285,00	7 313,00	11 386,00	8 94,00
35 200,00	9 307,00	7 330,00	11 411,00	8 15,00
35 250,00	9 329,00	7 347,00	11 436,00	8 36,00
35 300,00	9 351,00	7 361,00	11 459,00	8 57,00
35 350,00	9 373,00	7 377,00	11 483,00	8 78,00
35 400,00	9 395,00	7 393,00	11 507,00	8 99,00

@TAN

@TAN(z) donne la tangente de l'angle z. Dans un triangle rectangle, la tangente d'un angle est le rapport entre le côté opposé à cet angle et le côté adjacent.

Argument

x est un angle exprimé en radians. x peut prendre n'importe quelle valeur entre -2^{63} et 2^{63}

Exemple

@TAN(@DEGRAD(35)) = 0,700208 qui est la tangente d'un angle de 35 degrés.

Fonction similaire

@ATAN calcule l'arc tangente à partir de la tangente d'un angle. @TANH renvoie la tangente hyperbolique d'un angle.

@TANH

@TANH(z) renvoie la tangente hyperbolique d'un angle z. La tangente hyperbolique est le rapport du sinus hyperbolique d'un angle à son cosinus hyperbolique. La valeur renvoyée par @TANH est un nombre compris entre -1 et 1.

Arguments

x peut être n'importe quelle valeur entre -709.7827 et 709.7827 environ.

Exemples

@TANH(@DEGENRAD(30)) = 0,480473.

Fonctions similaires

ATANH renvoie l'arc tangente hyperbolique et TAN, la tangente d'un angle.

@DISTT

@DISTT(*x*; *degrés-liberté*; [*type*]; [*sens*]) évalue la distribution de Student (distribution de *t*) appliquée à la variable *x*.

Arguments

L'argument *x* est vu différemment selon la valeur donnée à *type*.

<u>Si <i>type</i> vaut</u>	<u><i>x</i> représente</u>
0 (défaut)	la valeur critique (ou valeur de seuil) de la variable aléatoire de la distribution de Student
1	une probabilité. Sa valeur est alors comprise entre 0 et 1.

degrés-liberté est le nombre de degrés de liberté de l'échantillon. C'est un entier positif.

type est un argument facultatif qui détermine le type du résultat renvoyé par la fonction @DISTT.

<u><i>type</i></u>	<u>1-2-3 renvoie</u>
0	La probabilité de risque correspondant à la valeur de seuil <i>x</i>
1	La valeur de seuil correspondant à la probabilité de risque <i>x</i>

sens est un argument facultatif qui précise le type de test à réaliser.

<u><i>sens</i></u>	<u>1-2-3 exécute</u>
1	Un test de Student mono-directionnel
2	Un double test de Student (sur les deux directions)

Si vous spécifiez un argument facultatif, vous devez obligatoirement spécifier tous les arguments précédents.

Remarques

@DISTT renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$. Si après 100 itérations, 1-2-3 ne peut pas fournir un résultat exact à 0,0000001 près, @DISTT renvoie ERR.

La distribution de Student est le rapport d'une distribution normale standard à la racine carrée d'une distribution de chi-carré divisée par le nombre de ses degrés de liberté.

Exemples

@DISTT(2,228;10) = 0,05

@DISTT(0,05;10;1) = 2,228

Fonctions similaires

@DISTCHI renvoie la fonction Chi-carrée (χ^2), @DISTF, la distribution de *F*. @TESTT évalue la probabilité associée à un test de Student.

@DUREE, @DUREEBIS

@DUREE(*versements*; *taux*; *valeur-future*) calcule le nombre de *versements* égaux pour qu'un investissement atteigne une *valeur-future* spécifiée, à un *taux* d'intérêt périodique donné.

@DUREEBIS(*versements*; *taux*; *valeur-future*; [*type*]; [*valeur-actuelle*]) calcule le nombre de *versements* égaux nécessaires pour atteindre une *valeur-future* spécifiée à un *taux* d'intérêt périodique donné. Selon le *type* spécifié, @DUREEBIS prend en compte des versements effectués soit en fin, soit en début de période.

Arguments

versements est le montant constant des versements. Vous pouvez spécifier n'importe quelle valeur à l'exception de 0.

taux est le taux d'intérêt périodique. Vous pouvez spécifier n'importe quelle valeur supérieure à -1.

valeur-future représente la somme que vous désirez atteindre. Vous pouvez spécifier une valeur quelconque.

type est un argument facultatif qui permet d'indiquer si les versements ont lieu en fin ou en début de période:

<u>type</u>	<u>les versements interviennent</u>
0 (défaut)	A la fin de chaque période
1	Au début de chaque période

valeur-actuelle est un argument facultatif qui permet de spécifier la valeur actuelle de la série de versements futurs. *valeur-actuelle* peut être un nombre quelconque. Si vous l'omettez, 1-2-3 utilise 0.

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarques

Vous pouvez utiliser une *valeur-future* négative pour calculer avec @DUREE le nombre de périodes nécessaires au remboursement d'un emprunt. Ainsi, vous devez rembourser 100.000 F prêtés à un taux annuel de 10% en versant des remboursements annuels de 11.746 F.

Il vous faudra 20 ans, résultat fourni par @ABS(@DUREE(11746;0,1;-100000)).

Exemples

Vous déposez régulièrement 2.000 F sur un compte rémunéré à 7,5 % par an et vous souhaiteriez connaître le nombre d'années nécessaires pour que ce compte atteigne 100.000 F :

Si vous effectuez vos versements en fin d'année, il vous faudra :

@DUREE(2000;0,075;100000) = 21,5 années

Si vous effectuez vos versements en début d'année :

@DUREEBIS(2000;0,075;100000;1) = 20,76 années

Fonction similaire

@DUREECOMP calcule le nombre de périodes nécessaires pour qu'un investissement atteigne une valeur future spécifiée.

@TERM2 calcule le nombre de périodes requises pour une série de versements égaux pour accumuler une valeur future à un taux d'intérêt périodique, en supposant qu'il s'agit d'une convention d'annuités.

@TEMPS

@TEMPS(*heure;minutes;secondes*) calcule le nombre temps correspondant à l'*heure*, aux *minutes* et aux *secondes* spécifiées.

Arguments

L'argument *heure* est un nombre entier compris entre 0 et 23 (23:00 ou 11:00 PM). L'argument *minutes* est un nombre entier compris entre 0 et 59. L'argument *secondes* est un nombre entier compris entre 0 et 59.

Emploi

La fonction **@TEMPS** sert à entrer des temps (heure, minutes, secondes) sous forme de nombres temps afin que 1-2-3 puisse les utiliser dans ses calculs et effectuer le suivi d'une période de temps écoulée (pour une facturation ou des tests, par exemple). **@TEMPS** est utile également pour les feuilles servant de calendrier.

Remarque

Utilisez un format temps pour faire apparaître un nombre temps sous le format date.

Exemple

$(@TEMPS(13;0;0)-@TEMPS(9;15;0))*95*24$ calcule les honoraires à verser à un ingénieur-conseil pour un jour donné, en soustrayant l'heure de fin d'activité de l'ingénieur de son heure de début, en multipliant par 24 le résultat en jours obtenu, puis en multipliant par un taux horaire de 450 F.

Fonction similaire

@HEURVAL convertit des libellés en nombres temps.

@HEURVAL

@HEURVAL(*chaîne*) calcule le nombre temps correspondant à l'heure spécifiée dans *chaîne*.

Argument

L'argument *chaîne* peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte. *chaîne* doit être dans l'un des quatre formats d'heure de 1-2-3.

Emploi

Cette fonction joue le même rôle que @TEMPS, la seule différence étant une entrée d'argument plus simple sous @HEURVAL. Utilisez @HEURVAL pour convertir en nombres temps des heures entrées sous forme de libellés afin de pouvoir les utiliser dans vos calculs. Cette fonction est particulièrement utile avec les données importées d'un autre programme (traitement de texte, par exemple).

Remarque

Utilisez un format temps pour afficher un nombre temps sous un format date.

Exemples

@HEURVAL("08:19:27") donne 0,34684.

@HEURVAL("08:19:27") donne 0,34684 sous la forme 08:19:27, si la cellule a pour format HH:MM:SS.

Fonction similaire

@TEMPS calcule le nombre temps correspondant à l'heure, aux minutes et aux secondes spécifiées.

@JOURACTUEL

@JOURACTUEL calcule le nombre date correspondant à la date actuelle.

Remarques

1-2-3 recalcule @JOURACTUEL chaque fois que vous recalculiez votre travail.

Utilisez un format dateH_GLDTFORM@SS1N60FR.HLP pour faire apparaître le nombre date comme la date qu'il représente.

Exemples

@JOURACTUEL = 31048 pour le 1er janvier 1985.

@JOURACTUEL = 33418 pour le 29 juin 1991.

@SUPPRESSPACE

@SUPPRESSPACE(*chaîne*) supprime de la *chaîne* les espaces de début et de fin ainsi que les espaces consécutifs superflus.

Argument

L'argument *chaîne* peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

Emploi

Utilisez **@SUPPRES**SPACE pour vous assurer que lorsque vous triez un champ, les entrées de base de données ne contiennent pas d'espaces inutiles qui pourraient affecter l'ordre de tri.

Exemple

@SUPPRESSPACE(" 45 3/8") = 45 3/8, après suppression de l'espace précédant 45 et de l'un des deux espaces placés entre 45 et 3/8.

@SUPPRESSPACE(" 500 Bd Carnot") = 500 Bd Carnot, après suppression de l'espace précédant 500, de deux espaces sur les trois entre 500 et Bd, et de l'un des deux espaces séparant Bd de Carnot.

Fonction similaire

@ALIGNCHAINE renvoie un texte aligné sur un nombre de caractères spécifié.

@VRAI

@VRAI donne la valeur logique 1 (vrai).

Emploi

Associez la fonction logique @VRAI à des macros ou à des fonctions utilisant la valeur logique 1 (vrai).

Remarques

Si une instruction logique de type $A1=B1$ est vraie, sa valeur logique est 1 ; si elle est fausse, sa valeur logique est 0. Vous pouvez entrer indifféremment 1 ou @VRAI dans des formules testant des conditions logiques, mais @VRAI rend la formule plus lisible.

Exemple

@SI(A6>500;@VRAI;@FAUX) = 1 si la cellule A6 contient une valeur supérieure à 500.

Fonction similaire

@FAUX donne la valeur logique 0.

@TRONQUE

@TRONQUE(x:[n]) tronque x à n décimales.

Arguments

x est un nombre.

n est un nombre entier entre -100 et +100, cet argument est facultatif.

Si n est @TRONQUE

Positif	Affecte la partie décimale de x. Par exemple, si n = 2, x sera tronqué au centième.
Négatif	Affecte la partie entière de x. Par exemple, si n = -2, x sera tronqué à la centaine.
0	Tronque x à sa partie entière.

Remarques

Utilisez le format des nombres fixé pour spécifier le nombre de décimales à afficher tout en laissant 1-2-3 continuer à utiliser les décimales non affichées ; n'utilisez pas @TRONQUE car 1-2-3 travaillerait alors sur les valeurs tronquées.

Exemples

@TRONQUE(123,45) = 123

@TRONQUE(123,54) = 123

@TRONQUE(-123,45) = -123

@TRONQUE(-123,54) = -123

@TRONQUE(123,45;1) = 123,4

@TRONQUE(-123,45;1) = -123,4

@TRONQUE(123,45;-2) = 100

@TRONQUE(-123,45;-1) = -120

Fonctions similaires

@ARRONDI, @ARRDEF, et @ARREXC arrondissent un nombre aux décimales spécifiées. @ARRMUL arrondit un nombre au multiple spécifié. @PAIR arrondit un nombre positif (négatif) au nombre pair supérieur (inférieur).

@IMPAIR arrondit un nombre positif (négatif) au nombre impair supérieur (inférieur). @ENT tronque un nombre à sa partie entière, en éliminant la partie décimale.

@MAJUSC

@MAJUSC(*chaîne*) convertit en majuscules toutes les lettres de la *chaîne*.

Argument

L'argument *chaîne* peut être du texte mis entre guillemets, une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte.

Emploi

La fonction de chaîne @MAJUSC permet, avec des libellés de diverses provenances, d'assurer la cohérence (majuscules/minuscules) des chaînes de caractères dans tout le fichier de feuilles. Vous pouvez ainsi, avant de trier une série de libellés (liste de pièces détachées, par exemple) convertir en majuscules tous les caractères qui les composent.

Remarque

La mise en majuscules affecte l'ordre de tri des libellés lors du tri d'un champ. Deux libellés, identiques par ailleurs, ne seront donc pas considérés comme ex aequo.

Exemples

@MAJUSC("Jean-Pierre") = JEAN-PIERRE

@MAJUSC(B2) = ATTENTION si B2 contient le libellé attention.

Fonctions similaires

@MINUSC convertit en minuscules toutes les lettres d'une *chaîne*. @NOMPROPRE convertit en majuscule uniquement la première lettre de chaque mot de la *chaîne*.

@VALEUR

@VALEUR(*chaîne*) convertit en nombre le nombre entré sous forme de *chaîne*.

Argument

L'argument *chaîne* peut être du texte mis entre guillemets ou un libellé ne comportant que des nombres. *chaîne* peut se présenter comme un nombre en format standard (456,7) ou scientifique (4,567E2), comme un nombre mixte (45 7/8) ou formaté (32,85%).

Emploi

Faites appel à la fonction de chaîne @VALEUR pour convertir en nombre un texte composé de nombres, afin de les employer dans des opérations mathématiques. Cette fonction est particulièrement utile pour convertir des entrées d'origine externe.

Remarque

@VALEUR ne tient pas compte des espaces de début et de fin de chaîne, mais retourne la valeur ERR lorsque *chaîne* contient des espaces entre les nombres et les symboles (32,85 % par exemple).

@VALEUR génère la valeur 0 lorsque son argument est une chaîne vide *chaîne vide* ou pointe sur une cellule vierge et la valeur ERR lorsqu'il comporte des caractères non numériques.

Appuyez sur F2 (MODIFIE) puis F9 (CALCUL) et validez par ENTREE pour remplacer @VALEUR par son résultat.

Vous ne pouvez pas effectuer de calculs à l'intérieur d'un argument *chaîne* d'une fonction @VALEUR, mais il est par contre possible de créer une formule en combinant plusieurs fonctions @VALEUR. Ainsi, @VALEUR("22"+"20") = 0, mais @VALEUR("22")+@VALEUR("20") = 42.

Exemples

@VALEUR("543") = le nombre 543.

@VALEUR(B3) = 49,75 si la cellule B3 contient le libellé 49 3/4.

@VALEUR("85%") = 0,85.

Fonction similaire

@CHAINE convertit un nombre en libellé.

@VAR, @VARE, @VARNUM, @VARENUM

@VAR(*liste*) calcule la variance des valeurs figurant dans l'argument *liste*, selon la méthode n (population).

@VARE(*liste*) calcule la variance des valeurs figurant dans l'argument *liste*, en utilisant la méthode n-1 (échantillon).

@VARNUM(*liste*) calcule la variance des valeurs figurant dans l'argument *liste*, selon la méthode n (population), sans tenir compte des cellules qui contiennent des libellés.

@VARENUM(*liste*) calcule la variance des valeurs figurant dans l'argument *liste*, en utilisant la méthode n-1 (échantillon), sans tenir compte des cellules qui contiennent des libellés.

Arguments

L'argument *liste* peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Remarques

La variance permet de mesurer la dispersion des différentes données d'une liste par rapport à la moyenne des valeurs de cette liste. Plusieurs tests statistiques de type analyse de variance requièrent ce calcul.

Pour calculer la variance, **@VAR** et **@VARNUM** font appel à la méthode n (population), qui suppose que les valeurs de *liste* représentent la totalité de la population. Si seules certaines valeurs ont été sélectionnées, la variance est biaisée, en raison des erreurs induites par l'échantillonnage.

@VARE et **@VARENUM** font appel à la méthode n-1 (échantillon) pour calculer l'écart type des valeurs d'échantillon. Cette méthode donne un résultat légèrement plus élevé que celui de la méthode n (population) afin de compenser les erreurs inhérentes à l'échantillonnage. Une variance plus importante n'est pas biaisée par ces erreurs, ce qui garantit une plus grande précision.

Exemples

@VAR et @VARE

Fonctions similaires

@BVAR et @BVARE calculent la variance des valeurs répondant aux critères spécifiés.

@AMDEGVAR

@AMDEGVAR(*coût; valeur-récupération; durée; début; fin; [coefficient]; [commute]*) calcule, pour une période déterminée par les arguments *début* et *fin*, la charge d'amortissement d'une immobilisation présentant un *coût* initial, une *durée* de vie et une *valeur-récupération* finale, en appliquant la méthode de l'amortissement dégressif à taux double (taux de l'amortissement linéaire multiplié par 2), à moins que l'argument facultatif *coefficient* ne soit utilisé (il s'agit alors d'un amortissement dégressif à taux variable). La méthode de l'amortissement dégressif à taux double, spécifique au système fiscal américain, peut être adaptée à l'amortissement dégressif français (amortissement dégressif à taux variable) par l'entrée du coefficient adéquat dans l'argument facultatif *coefficient* (automatiquement multiplié au taux de l'amortissement linéaire pour donner le taux de l'amortissement dégressif correspondant.)

Arguments

L'argument *coût*, montant payé pour l'immobilisation, doit être une valeur supérieure à *valeur-récupération*. L'argument *valeur-récupération*, valeur du bien à la fin de sa durée d'utilisation, peut être une valeur quelconque. L'argument *durée* indique le nombre de périodes nécessaires pour que l'immobilisation atteigne la valeur de récupération. *durée* peut être une valeur quelconque, supérieure à 0. L'argument *début* correspond à la période au cours de laquelle vous souhaitez faire démarrer le calcul de l'amortissement. *début* peut être une valeur quelconque, supérieure à 0 et inférieure à *durée*. L'argument *fin* détermine la période au cours de laquelle vous voulez interrompre le calcul de l'amortissement. *fin* est une valeur quelconque, supérieure à *début*. Les arguments *début* et *fin* concernent la durée de vie de l'immobilisation, établie en fonction de l'exercice comptable. Ainsi, pour calculer la charge d'amortissement de la première année d'un bien acquis au début du deuxième trimestre de l'exercice, *début* aura la valeur 0 et *fin* la valeur 0,75 (1 année moins un trimestre). Les calculs d'amortissement effectués à l'aide de @AMDEGVAR peuvent porter sur plusieurs périodes.

@AMDEGVAR autorise la prise en compte d'une période initiale permettant de calculer l'amortissement à partir de la mise en service du bien. Dans ce cas, la partie décimale des périodes *début* ou *fin* sert à déterminer la *période initiale*. Si les périodes de début et de fin ont toutes deux une partie décimale, c'est celle de début qui s'applique. L'argument *coefficient* est un argument facultatif qui permet d'indiquer le coefficient qui sera multiplié au taux de l'amortissement linéaire pour obtenir le taux de l'amortissement dégressif. Si vous n'entrez pas cet argument, 1-2-3 utilise un coefficient de 2 (qui s'applique en France à l'amortissement d'immobilisations dont la durée d'utilisation est de 5 ou 6 ans) ; le résultat obtenu équivaut à celui calculé par la méthode dégressive à taux double (@AMORDEG). *coefficient* peut être une valeur quelconque, supérieure à 0. Les valeurs communément rencontrées sont 1,5, 2 ou 2,5. L'argument facultatif *commute* permet d'empêcher le passage à la méthode de l'amortissement linéaire pour le reste de la durée d'utilisation du bien. Normalement, ce passage s'effectue lorsque les résultats obtenus avec cette méthode sont supérieurs à ceux générés par la méthode de l'amortissement dégressif. L'argument *commute* peut avoir la valeur 0 ou 1.

Si <i>commute</i> est égal	@AMDEGVAR
0 (valeur par défaut)	passe automatiquement à la méthode de l'amortissement linéaire lorsque la charge d'amortissement ainsi obtenue est supérieure à celle de l'amortissement dégressif
1	cette commutation ne peut avoir lieu

Vous ne pouvez pas utiliser un argument facultatif sans ceux qui le précèdent.

Emploi

Utilisez la fonction financière @AMDEGVAR avec l'argument facultatif *coefficient* lorsque vous avez besoin de choisir un taux d'amortissement dégressif particulier, autre que celui appliqué pour des biens dont la durée d'utilisation est de 5 ou 6 ans (servez-vous, dans ce dernier cas, de la fonction @AMORDEG, qui couvre cette situation fiscale française spécifique).

Remarques

Pour la méthode suivie par @AMDEGVAR (lorsque l'argument *commute* est utilisé), le taux d'amortissement reste constant, jusqu'à ce que la valeur de récupération de l'immobilisation soit inférieure à la valeur obtenue par l'équation suivante :

$(\text{valeur comptable} * ((1 - (\text{taux} / \text{durée}))^{\text{durée}}))$

où valeur comptable = coût - (valeur-récupération) - amortissements antérieurs. Dès lors, 1-2-3 applique la méthode d'amortissement linéaire pour le reste de la durée de vie de l'immobilisation, afin d'éviter toute surévaluation de la valeur de récupération. Lors de ce changement, le résultat obtenu avec @AMDEGVAR est ajusté, si besoin est, pour

que l'amortissement total sur la durée de vie de l'immobilisation soit égal au coût de l'immobilisation moins la valeur de récupération.

Exemple

Cet exemple montre l'échéancier d'amortissement d'un équipement de bureau dont le coût initial est de 50.000 F, la durée d'utilisation de 4 ans et la valeur de récupération de 5.000 F. L'amortissement est calculé pour chacune des 4 années, à l'aide de la méthode de l'amortissement dégressif à taux variable, sur la base d'un coefficient de 1,5.

@AMDEGVAR(50000;5000;4;0;1;1,5) = 18 750,00 F
@AMDEGVAR(50000;5000;4;1;2;1,5) = 11 718,75 F
@AMDEGVAR(50000;5000;4;2;3;1,5) = 7 031,25 F
@AMDEGVAR(50000;5000;4;3;4;1,5) = 7 207,03 F
@AMDEGVAR(50000;5000;4;1;2;1,5) = 11 718,75 F
@AMDEGVAR(50000;5000;4;1;2;1,5) = 11 718,75 F
@AMDEGVAR(50000;5000;4;3;4;1,5) = 7 207,03 F
@AMDEGVAR(50000;5000;4;3;4;1,5) = 7 207,03 F
@AMDEGVAR(50000;5000;4;1;2;1,5) = 11 718,75 F
@AMDEGVAR(50000;5000;4;3;4;1,5) = 7 207,03 F
@AMDEGVAR(50000;5000;4;1;2;1,5) = 11 718,75 F
113 203,12 F

Fonctions similaires

@AMORDEG calcule l'amortissement selon la méthode de l'amortissement dégressif à taux double. @AMORLIN fait appel à la méthode linéaire, tandis que @AMORANN applique un amortissement proportionnel à l'ordre numérique inversé des années.

@CONSV

@CONSV(*x*; *champ*; *position-colonne*) renvoie le contenu d'une cellule figurant dans une colonne d'un tableau de consultation verticale.

Arguments

x peut être un nombre ou un libellé, en fonction de la première colonne du tableau de consultation verticale.

<u>Première colonne</u>	<u>x</u>
Nombres	<i>x</i> peut être une valeur quelconque supérieure ou égale à la première valeur de <i>champ</i> . @CONSV génère la valeur <u>ERR</u> lorsque <i>x</i> est inférieur à la première valeur de <i>champ</i> . Si <i>x</i> est supérieur à la dernière valeur de la première colonne de <i>champ</i> , @CONSV s'arrête à la dernière cellule de <i>position-colonne</i> et en donne le contenu.
Libellés	Texte mis entre " " (guillemets), une formule texte, ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule texte. Lorsque <i>x</i> ne correspond pas exactement au contenu d'une cellule de la première colonne de <i>champ</i> , @CONSV génère la valeur ERR.

L'argument *champ*, qui désigne l'emplacement du tableau de consultation verticale, peut être une adresse de cellule ou un nom de champ. Si *champ* est multifeuille, 1-2-3 ne tient compte que de la feuille actuelle.

L'argument *position-colonne* représente le numéro de position de la colonne dans *champ*.

Remarques

@CONSV compare la valeur *x* à chacune des cellules de la première colonne du tableau. Lorsque 1-2-3 atteint dans cette première colonne une cellule contenant *x* (ou, si *x* est une valeur, la valeur la plus proche mais pas supérieure), il se déplace horizontalement du nombre de colonnes spécifié dans l'argument *position-colonne* puis affiche le contenu de la cellule ainsi pépérée.

Exemple

@CONSV

Fonctions similaires

@CONSH recherche une entrée dans un tableau de consultation horizontale. @INDEX retourne le contenu de la cellule située à l'intersection de la colonne et de la ligne identifiées par leur numéro de position dans le champ. @XINDEX renvoie le contenu d'une cellule située à l'intersection d'une ligne et d'une colonne identifiées par leur contenu. @CHOIX est utilisée en remplacement d'un tableau de consultation d'une seule ligne. @CORRESP indique la position relative d'une cellule identifiée par son contenu. @CONSMAX renvoie la référence absolue de la cellule qui contient la valeur la plus élevée de liste-champs. @CONSMIN retourne la référence absolue de la cellule qui contient la valeur la plus basse de liste-champs.

@JOURSEM

@JOURSEM(*date*) renvoie un nombre de 0 à 6 identifiant le jour de la semaine (de 0 pour lundi à 6 pour dimanche) auquel correspond *nombre-date*.

Arguments

date est un nombre date.

Exemples

@JOURSEM(**@DATE**(91;7;3)) = 2 (mercredi)

Fonctions similaires

@MOIS renvoie le numéro de mois entre 1 et 12 correspondant à un nombre date. **@ANNEE** renvoie un numéro entre 0 (pour 1900) et 199 (pour 2099) correspondant à un nombre date.

@ANNEE

@ANNEE(*nombre-date*;*type*) extrait d'un *nombre-date* donné l'année correspondante, sous la forme d'un nombre entier compris entre 0 (1900) et 199 (2099).

Arguments

L'argument *nombre-date* est un nombre entier compris entre 1 (1er janvier 1900) et 73050 (31 décembre 2099), ou le nom ou l'adresse d'une cellule contenant ce nombre.

type est le nombre 0 ou 1. Si *type* est 1, **@ANNEE** renvoie l'année sous forme d'un nombre à quatre chiffres. Si *type* est 0 ou ignoré, **@ANNEE** renvoie la différence de l'année par rapport à 1900 (par exemple, **@ANNEE** renvoie 123 pour l'année 2023).

Remarque

@ANNEE peut également fournir l'argument *année* à d'autres fonctions **@** de date qui requièrent des dates calculées au préalable.

Exemples

@ANNEE(20181) = 55, car le nombre date 20181 correspond au 02-Avr-55.

@ANNEE(@JOURACTUEL) = l'année en cours.

@ANNEE(@DATVAL("14-Fév-91")) = 91.

Fonctions similaires

@JOUR extrait du nombre date le jour du mois (1 à 31) et **@MOIS** en extrait le mois (1 à 12).

@RENDEMENT

@RENDEMENT(*livraison*; *maturité*; *coupon*; *prix*; [*remboursement-nominal*]; [*fréquence*]; [*base*]) calcule le rendement d'obligations à intérêt périodique.

Arguments

livraison est la date d'achat de l'obligation, exprimée en nombre date.

maturité est la date d'échéance de l'obligation, exprimée en nombre date. Si *maturité* est inférieure ou égale à *livraison*, @RENDEMENT retourne ERR.

coupon est le taux nominal annuel de l'obligation. Ce peut être n'importe quel nombre positif ou nul.

prix est le prix payé pour une obligation de 100 F de nominal. Ce peut être n'importe quel nombre positif.

remboursement-nominal est un argument facultatif qui permet de spécifier la valeur de remboursement d'une obligation de 100 F de nominal. Ce peut être n'importe quel nombre positif ou nul ; si vous l'omettez, 1-2-3 utilise 100.

fréquence est un argument facultatif permettant d'indiquer la fréquence de paiement des coupons.

<u>fréquence</u>	<u>Fréquence des coupons de paiement</u>
1	Annuelle
2 (défaut)	Semi-annuelle
3	Trimestrielle
12	Mensuelle

base est un argument facultatif qui spécifie le type de calendrier jours/mois à prendre en compte.

<u>base</u>	<u>Base de comptage des jours</u>
0	30/360
1	Réel/réel
2	Réel/360
3	Réel/365
4	Européen 30/360

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Exemples

Un bon a pour date de livraison le 1er juillet 1993 et vient à échéance le 1er décembre 1998. Les coupons sont payés semi-annuellement et ont un taux nominal de 5,50%. Le prix du bon est de 99,50 F et sa valeur de remboursement est de 100. Le calendrier utilisé est de type 30/360. La formule ci-dessous vous permet de déterminer le rendement de ce bon :

@RENDEMENT(@DATE(93;7;1);@DATE(98;12;1);0,055;99,5;100;2;0) = 0,056072

Fonctions similaires

@INTCOUR calcule les intérêts courus pour des titres à intérêt périodique. @PRIX calcule le prix de titres à intérêt périodique d'un nominal de 100. @DURATION calcule la durée à l'origine et @SENSIBILITETITRE, la sensibilité à l'origine d'un titre de 100 F à intérêt périodique.

@RENDEMENT2 renvoie le rendement des obligations à intérêt périodique, à l'aide des conventions japonaises.

Exemple de fonction @TESTCHI (test de validité)

@TESTCHI(A2..A9;B2..B9) = 0,996882

A ----- A ----- B -----

	Observé	Prévu
1		
2	25	23
3	34	35
4	87	91
5	91	88
6	34	34
7	23	22
8	56	60
9	70	68

Exemple de fonction @TESTCHI (test d'indépendance)

@TESTCHI(B3..C5) = 0,080809

A	A	B	C
1		Brut	Lissé
2			
3	Elevé	42	51
4	Moyen	25	37
5	Faible	85	68

Exemple de fonction @BMOYENNE

Une table de base de données nommée VENTES recense les ventes de studios à Neuilly, Levallois et Puteaux en avril et mai. Le prix de vente des studios apparaît sous la rubrique PRIX. Vous voulez déterminer le prix de vente moyen d'un studio à Levallois.

@BMOYENNE(VENTES;"PRIX";"VILLE=Levallois") = 365.667

A	-----	A	-----	B	-----	C	-----
1		ADRESSE		VILLE		PRIX	
2		33, place Pompidou		Levallois		720.000	
3		1, rue Blanche		Puteaux		318.000	
4		2, bd Bineau		Neuilly		332.000	
5		Parc de la Planchette		Levallois		278.000	
6		4, av. Wilson		Levallois		160.000	
7		18, av. Wilson		Levallois		227.000	
8		Résidence Offenbach		Puteaux		397.000	
9		Rue Monge		Puteaux		303.000	
10		45, rue Jaurès		Levallois		669.000	
11		16, rue Baudin		Levallois		140.000	

Exemple de fonction @BCOMPTE

Une table de base de données nommée AVR_VENTES recense les ventes d'appartements pour le mois d'avril. Le type de chauffage est indiqué dans le champ CHAUF. Vous voulez compter le nombre d'appartements chauffés au gaz :

@BCOMPTE(AVR_VENTES;"CHAUF";"CHAUF=Gaz") = 4

A	----	A	-----	B	-----	C	-----	D	-
1		VILLE		PCES		CHAUF		COUT	
2		Asnières		5		Gaz		690.000	
3		Clichy		5		Autre		305.000	
4		Puteaux		2		Gaz		335.000	
5		Versailles		4		Elec		328.000	
6		Levallois		2		Gaz		374.000	
7		Courbevoie		3		Gaz		395.000	

Exemple de fonction @BLIT

Une table de base de données nommée VENTES recense les ventes de studios dans trois villes en avril et mai. Les commissions des vendeurs apparaissent dans le champ nommé COMM. Vous voulez déterminer la commission du vendeur du studio situé rue Monge :

@BLIT(VENTES;"COMM";"ADRESSE=Rue Monge") = 12.120

A	-----	A	-----	B	-----	C	---
1		ADRESSE		VENDEUR		COMM	
2		33, place Pompidou		Ferry		28.800	
3		1, rue Blanche		Levine		12.720	
4		2, bd Bineau		Ferry		19.920	
5		Parc de la Planchette		Forget		11.120	
6		4, av. Wilson		Dubard		9.600	
7		18, av. Wilson		Levine		13.620	
8		Résidence Offenbach		Dubard		15.880	
9		Rue Monge		Ferry		12.120	

Exemple de fonction @BMAX

Une table de base de données nommée VENTES recense les ventes de studios à Neuilly, Levallois et Puteaux en avril et mai. Le prix de vente des studios apparaît sous la rubrique PRIX. Vous voulez déterminer le prix de vente maximum d'un studio à Levallois.

@BMAX(VENTES;"PRIX";"VILLE=Levallois") = 720.000

A	----- A -----	B -----	C -----
1	ADRESSE	VILLE	PRIX
2	33, place Pompidou	Levallois	720.000
3	1, rue Blanche	Puteaux	318.000
4	2, bd Bineau	Neuilly	800.000
5	Parc de la Planchette	Levallois	278.000
6	4, av. Wilson	Levallois	160.000
7	18, av. Wilson	Levallois	227.000
8	Résidence Offenbach	Puteaux	397.000
9	Rue Monge	Puteaux	303.000
10	45, rue Jaurès	Levallois	669.000
11	16, rue Baudin	Levallois	140.000

Exemple de fonction @BMIN

Une table de base de données nommée VENTES recense les ventes de studios à Neuilly, Levallois et Puteaux en avril et mai. Le prix de vente des studios apparaît sous la rubrique PRIX. Vous voulez déterminer le prix de vente minimal d'un studio à Levallois.

@BMIN(VENTES;"PRIX";"VILLE=Levallois") = 140.000

A	----- A -----	B -----	C -----
1	ADRESSE	VILLE	PRIX
2	33, place Pompidou	Levallois	720.000
3	1, rue Blanche	Puteaux	318.000
4	2, bd Bineau	Neuilly	332.000
5	Parc de la Planchette	Levallois	278.000
6	4, av. Wilson	Levallois	160.000
7	18, av. Wilson	Levallois	227.000
8	Résidence Offenbach	Puteaux	397.000
9	Rue Monge	Puteaux	130.000
10	45, rue Jaurès	Levallois	669.000
11	16, rue Baudin	Levallois	140.000

Exemple de fonction @BSTD et @BSTDE

Une table de base de données nommée RENOVATION recense les travaux de rénovation effectués sur plusieurs immeubles de 1982 à 1986. Pour économiser de l'espace, les lignes 4 à 37 de la feuille de travail ont été cachées. La taille du lot correspondant aux immeubles rénovés apparaît dans le champ LOT. Vous voulez calculer l'écart-type des tailles de lot pour les travaux dont le coût a dépassé 125.000 F.

@BSTD(RENOVATION;"LOT";"COUT>125000") = 0,47

A	----	A	-----	B	-----	C	-----	D	--
1		VILLE		LOT		TYPE		COUT	
2		Asnières		0,25		Toiture		290000	
3		Clichy		0,40		Façade		105000	
4		Puteaux		0,50		Toiture		135000	
5		Versailles		1,00		Chauffage		128000	
6		Levallois		0,50		Toiture		174000	
7		Courbevoie		0,25		Toiture		195000	
8		Nanterre		0,25		Façade		118000	

Exemple de fonction @BSOMME

Une table de base de données nommée VENTES recense les ventes de studios dans trois villes en avril et mai. Les commissions des vendeurs apparaissent dans le champ nommé COMM. Vous voulez déterminer le total des commissions attribuées au vendeur Dubard :

@BSOMME(VENTES;"COMM";VENDEUR="Dubard") = 25480

A	-----	A	-----	B	-----	C	---
1		ADRESSE		VENDEUR		COMM	
2		33, place Pompidou		Ferry		28.800	
3		1, rue Blanche		Levine		12.720	
4		2, bd Bineau		Ferry		19.920	
5		Parc de la Planchette		Forget		11.120	
6		4, av. Wilson		Dubard		9.600	
7		18, av. Wilson		Levine		13.620	
8		Résidence Offenbach		Dubard		15.880	
9		Rue Monge		Ferry		12.120	

Exemples de fonctions @BVAR et @BVARE

Une table de base de données nommée RENOVATION recense les travaux de rénovation effectués sur plusieurs immeubles de 1982 à 1986. Pour économiser de l'espace, les lignes 4 à 37 de la feuille de travail ont été cachées. La taille du lot correspondant aux immeubles rénovés apparaît dans le champ LOT. Vous voulez calculer la variance des tailles de lot pour les travaux dont le coût a dépassé 125.000 F.

@BVAR(RENOVATION;"LOT";"COUT>125000") = 0,216385

@BVARE(RENOVATION;"LOT";CHAMP_CRIT) = 0,224114

A	----	A	-----	B	----	C	-----	D	--
1		VILLE		LOT		TYPE		COUT	
2		Asnières		0,25		Toiture		290,000	
3		Clichy		0,40		Façade		105,000	
4		Puteaux		0,50		Toiture		135,000	
5		Versailles		1,00		Chauffage		128,000	
6		Levallois		0,50		Toiture		174,000	
7		Courbevoie		0,25		Toiture		195,000	
8		Nanterre		0,25		Façade		118,000	

Exemple de fonction @TESTF

@TESTF(A2..A13;B2..B15) = 0,157348

A	A	B
1	Echantillon1	Echantillon2
2	84,5	1,65
3	80,7	4,58
4	34,5	42,60
5	54,6	4,37
6	50,5	30,80
7	33,7	97,70
8	46,8	87,20
9	47,6	40,70
10	22,8	38,40
11	15,5	10,60
12	60,6	56,30
13	80,5	70,50
14		9,04
15		97,30

Exemple de fonction @TOTALGEN

@TOTALGEN en A10 calcule la somme des cellules A1..A8 contenant @SOUSTOTAL (A4 et A8).

A	---	A	-----	B	-----
1		10			
2		15			
3					
4		25		@SOUSTOTAL (A1 . . A2)	
5					
6		20			
7		25			
8		45		@SOUSTOTAL (A6 . . A7)	
9					
10		70		@TOTALGEN (A1 . . A8)	

Exemple de fonction @CORRESP

Le dosage d'un médicament est fonction du poids du patient. Ce dernier, entré dans une cellule nommée POIDS_PATIENT, est de 60 kg.

@INDEX(A2..C7;2;@CORRESP(POIDS_PATIENT;A2..C7;1)) = 2

A	A	B	C
1	Livres	Kilos	Nombre de pilules
2	50	22,5	1,5
3	100	45,5	2,0
4	150	68,0	2,5
5	200	90,5	3,0
6	250	113,5	3,5
7	300	136,0	4,0

Exemple de fonction @TESTT

@TESTT(A2..A13;B2..B13) = 0,050022

A	A	B
1	Echantillon1	Echantillon
2	84,5	65,1
3	80,7	85,4
4	34,5	62,4
5	54,6	73,4
6	50,5	80,3
7	33,7	66,7
8	46,8	87,2
9	47,6	70,4
10	22,8	30,2
11	15,5	60,1
12	60,6	56,3
13	80,5	70,5

Exemple de fonction @XINDEX

Un tableau nommé TARIF (A2..E7) donne les tarifs d'affranchissement de lettres en fonction de leur poids et du type de recommandation.

@XINDEX(TARIF;"R2";50) = 20,50 (coût d'affranchissement d'une lettre de 50g au taux R2).

@XINDEX(TARIF;"R3";500), = 37,00 (coût d'affranchissement d'une lettre de 500g au taux R3).

A --- A ----- B ----- C ---- D ----- E ---

1 ----- TARIFS RECOMMANDES -----

2 Poids maxi	Sans	R1	R2	R3
3 20	2,50	17,00	19,00	24,50
4 50	4,00	18,50	20,50	26,00
5 100	6,20	20,70	22,70	28,20
6 250	11,00	25,50	27,50	33,00
7 500	15,00	29,50	31,50	37,00

@TESTCHI

@TESTCHI(*champ1*;*[champ2]*) effectue un test de χ^2 (chi-carré) sur les valeurs des champs *champ1* et *champ2* et renvoie la probabilité associée.

Arguments

champ1 et *champ2* sont des champs de même taille et ne contiennent que des valeurs numériques. Dans le cas contraire (tailles différentes, libellés, formules de type chaîne, cellules vides), **@TESTCHI** renvoie ERR.

Remarques

@TESTCHI renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$.

Exemples

@TESTCHI : test d'indépendance

@TESTCHI : test de validité

Fonctions similaires

@DISTCHI évalue la distribution Chi-carrée. **@TESTF** effectue un test de *F*, **@TESTT**, un test de Student et **@TESTZ**, un test de *Z*.

@CRITBINOM

@CRITBINOM(*essais;probabilité;maxprob*) renvoie le plus grand nombre entier tel que la fonction binomiale cumulative soit inférieure ou égale à *maxprob*.

Arguments

essais représente le nombre de fois qu'est répétée une épreuve qui ne comporte que deux éventualités (variable aléatoire de Bernoulli). C'est donc un entier positif ou nul.

probabilité est une valeur comprise entre 0 et 1 inclus et représente la probabilité de succès d'une épreuve de Bernoulli.

maxprob est le critère de probabilité (compris également entre 0 et 1).

Remarques

@CRITBINOM renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$.

Exemple

Vous dirigez une fabrique de filtres à huile. Les filtres sont conditionnés par boîte de 100. Il y a 85% de chances qu'un filtre soit exempt de tout défaut. Une boîte de cent filtres est mise au rebut si plus de 90% des filtres qu'elle contient sont défectueux. Vous voulez déterminer le nombre maximum de filtres défectueux qui peuvent être fabriqués avant qu'une boîte soit écartée.

@CRITBINOM(100;0,85;0,90)

Fonctions similaires

@BINOMIALE calcule la fonction binomiale cumulative. **@COMBIN** renvoie le nombre de combinaisons (autrement dit les coefficients binomiaux). **@ARRANG** renvoie le nombre d'arrangements.

@BMOYENNE

@BMOYENNE(*données;rubrique;critères*) renvoie la moyenne des valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

Arguments

Voir Arguments des fonctions @ des bases de données.

Exemples

@BMOYENNE

Fonctions similaires

@MOYENNE et @MOYENNENUM renvoient la moyenne des valeurs d'une liste.

@BCOMPTE et @BCOMPTENUM

@BCOMPTE(*données;rubrique;critères*) renvoie le nombre de cellules occupées de la *rubrique* de table de base de données qui répondent aux *critères*.

@BCOMPTENUM(*données;rubrique;[critères]*) renvoie le nombre de cellules de la *rubrique* de table de base de données qui répondent aux *critères* et qui contiennent des valeurs numériques.

Arguments

Voir Arguments des fonctions @ des bases de données.

Exemples

@BCOMPTE

Fonctions similaires

@COMPTE et @COMPTENUM renvoient le nombre de cellules d'une liste.

@BLIT

@BLIT(*données;rubrique;[critères]*) retourne le contenu (nombre ou libellé) d'une *rubrique* de table de base de données qui répond aux *critères* spécifiés.

Arguments

Voir Arguments des fonctions @ des bases de données.

Remarques

Si plus d'une entrée répond aux *critères*, @BLIT retourne ERR.

@BLIT est utile pour retrouver une information unique : le numéro de code d'un employé par exemple. @BLIT vous permet de localiser cette information automatiquement, pour utilisation dans une macro, comme argument d'une fonction @, ou comme variable dans une formule.

Exemples

@BLIT

Fonctions similaires

@CONSH et @CONSV renvoient le contenu d'une cellule dans un tableau de consultation horizontale ou verticale.

@CHOIX est utilisée en remplacement d'un tableau de consultation d'une seule ligne. @INDEX renvoie le contenu d'une cellule située à l'intersection de la colonne et de la ligne identifiées par leur numéro de position dans le champ.

@XINDEX restitue le contenu d'une cellule située à l'intersection d'une ligne et d'une colonne identifiées par leur contenu. @@ renvoie le contenu d'une cellule repérée par adressage indirect.

@BMAX

@BMAX(*données;rubrique;critères*) renvoie la plus grande des valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

Arguments

Voir [Arguments des fonctions @ des bases de données.](#)

Remarques

Vous pouvez utiliser **@BMAX** pour déterminer la date ou l'heure la plus récente.

Exemples

[@BMAX](#)

Fonctions similaires

[@MAX](#) et [@MAXNUM](#) renvoient la plus grande valeur d'une liste.

@BMIN

@BMIN(*données;rubrique;critères*) renvoie la plus petite des valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

Arguments

Voir [Arguments des fonctions @ des bases de données.](#)

Remarques

Vous pouvez utiliser **@BMIN** pour déterminer la date ou l'heure la plus ancienne.

Exemples

[@BMIN](#)

Fonctions similaires

[@MIN](#) et [@MINNUM](#) renvoient la plus petite des valeurs d'une liste.

BSTD, @BSTDE

@BSTD(*données;rubrique;[critères]*) renvoie l'écart-type de la population représentée par les valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

@BSTDE(*données;rubrique;[critères]*) renvoie l'écart-type de l'échantillon représenté par les valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

Arguments

Voir Arguments des fonctions @ des bases de données.

Remarques

Pour calculer l'écart-type, @BSTD fait appel à la méthode n (population), qui suppose que les valeurs données représentent la totalité de la population. Si seules certaines valeurs ont été sélectionnées, l'écart-type est biaisé, en raison des erreurs induites par l'échantillonnage.

L'écart-type est égal à la racine carrée de la variance (ou fluctuation).

Exemples

@BSTD et @BSTDE

Fonctions similaires

@STD et @STDNUM renvoient l'écart-type d'une population. @STDE et @STDENUM renvoient l'écart-type d'un échantillon. @BVAR calcule la variance d'une population représentée par les valeurs d'une rubrique d'une base de données.

@BSOMME

@BSOMME(*données;rubrique;critères*) renvoie la somme des valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

Arguments

Voir Arguments des fonctions @ des bases de données.

Exemples

@BSOMME

Fonctions similaires

@SOMME renvoie la somme des valeurs d'une liste. @SOMMENEGATIFS n'additionne que les valeurs négatives d'une liste. @SOMMEPOSITIFS n'additionne que les valeurs positives d'une liste.

@BVAR, @BVARE

@BVAR(*données;rubrique;critères*) renvoie la variance de la population représentée par les valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

@BVARE(*données;rubrique;critères*) renvoie la variance de l'échantillon représenté par les valeurs d'une *rubrique* de table de base de données qui répondent aux *critères*.

Arguments

Voir Arguments des fonctions @ des bases de données.

Remarques

La variance permet de mesurer la dispersion des différentes données d'une liste par rapport à la moyenne des valeurs de cette liste. Plusieurs tests statistiques de type analyse de variance requièrent ce calcul.

Pour calculer la variance, **@BVAR** fait appel à la méthode n (population), qui suppose que les valeurs de *liste* représentent la totalité de la population. Si seules certaines valeurs ont été sélectionnées, la variance est biaisée, en raison des erreurs induites par l'échantillonnage.

La variance est le carré de l'écart-type.

Exemples

@BVAR et **@BVARE**

Fonctions similaires

@VAR et **@VARNUM** renvoient la variance d'une population, **@BSTD**, l'écart-type d'une population représentée par les valeurs d'une rubrique de base de données.

@TESTF

`@TESTF(champ1;champ2)` effectue un test de F sur les valeurs des champs *champ1* et *champ2* et renvoie la probabilité associée.

Arguments

champ1 et *champ2* sont les champs qui contiennent les données à tester. Ils n'ont pas nécessairement le même nombre de cellules.

Remarques

`@TESTF` renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$ près.

`@TESTF` est particulièrement utile pour estimer la différence de variance entre deux échantillons.

Exemples

`@TESTF`

Fonctions similaires

`@DISTF` évalue la fonction de distribution de F . `@TESTCHI` effectue un test de Chi-carré, `@TESTT` un test de Student, et `@TESTZ` un test de Z .

@TOTALGEN

@TOTALGEN(*liste*) renvoie la somme de toutes les cellules de *liste* qui contiennent @SOUSTOTAL dans leur formule.

Arguments

liste peut contenir une combinaison quelconque de champs. Séparez les éléments de *liste* au moyen du séparateur d'arguments.

Voir aussi Arguments des fonctions @ statistiques.

Exemples

@TOTALGEN

Fonctions similaires

@SOMME fait la somme des valeurs d'une liste. @BSOMME fait la somme des valeurs d'une rubrique de base de données qui répondent aux critères spécifiés. @SOMMENEGATIFS n'additionne que les valeurs négatives d'une liste. @SOMMEPOSITIFS n'additionne que les valeurs positives d'une liste.

@ESTFICHIER

`@ESTFICHIER("nom-fichier";[type])` vérifie la présence sur disque ou en mémoire du fichier nommé *nom-fichier*. Si ce fichier existe, `@ESTFICHIER` renvoie 1 (vrai) ; sinon, `@ESTFICHIER` renvoie 0 (faux).

Arguments

nom-fichier est le nom complet, extension comprise, du fichier dont vous vérifiez la présence. Il doit être entré comme du texte. Si vous recherchez le fichier ailleurs que dans le dossier courant, vous devez indiquer le chemin d'accès à ce fichier.

type spécifie où contrôler la présence du fichier (en mémoire ou sur le disque).

type = 0 vérifie la présence en mémoire,

type = 1 vérifie la présence sur disque.

Exemple

`@ESTFICHIER("C:\123WBUDGET\COUTS_93.123";1) = 1`, si le fichier COUTS_93.123 est enregistré dans le dossier C:\123WBUDGET.

@CORRESP

@CORRESP(*contenu*;*champ*;*[type]*) renvoie la position relative dans *champ* de la cellule dont le contenu correspond au *contenu* spécifié. Cette position est renvoyée sous la forme d'un numéro de position.

Arguments

contenu peut être un nombre ou du texte. Si *contenu* est du texte, vous pouvez utiliser des caractères joker.

champ est un champ désigné par son nom ou son adresse.

type est un argument facultatif précisant le type de correspondance à établir entre l'argument *contenu* et le contenu de la cellule recherchée dans le *champ*.

typ @CORRESP renvoie la position relative de e

- | | |
|---|---|
| 0 | La première cellule dont le contenu correspond à <i>contenu</i> . |
| 1 | La première cellule dont le contenu est inférieur ou égal à <i>contenu</i> ; valeur par défaut si vous omettez l'argument. Trie <i>champ</i> par ordre croissant. |
| 2 | La première cellule dont le contenu est plus grand ou égal à <i>contenu</i> . Trie <i>champ</i> par ordre décroissant. |

Remarques

1-2-3 explore le *champ* de haut en bas dans une colonne, puis de gauche à droite. Si *champ*

est multifeuille, 1-2-3 explore d'abord la première feuille de *champ*, puis la seconde, et ainsi de suite jusqu'à ce qu'il localise une correspondance ou qu'il atteigne l'extrémité du champ.

Si aucune correspondance n'est trouvée, @CORRESP renvoie la valeur ERR.

Lorsque *type* prend la valeur 1 et que la première cellule du *champ* contient une valeur supérieure à celle située dans *contenu*, @CORRESP renvoie la valeur ERR. Si *type* est égal à 2 et que la première cellule du *champ* contient une valeur inférieure à celle située dans *contenu*, @CORRESP renvoie également la valeur ERR.

Exemples

@CORRESP

Fonctions similaires

@CONSH et @CONSV renvoient le contenu d'une cellule dans un tableau de consultation horizontale ou verticale.

@INDEX renvoie le contenu d'une cellule située à l'intersection de la colonne et de la ligne identifiées par leur numéro de position dans le champ. @CHOIX est utilisée en remplacement d'un tableau de consultation d'une seule ligne.

@CONSMAX renvoie la référence absolue de la cellule qui contient la valeur la plus élevée de liste-champs.

@CONSMIN renvoie la référence absolue de la cellule qui contient la valeur la plus basse de liste-champs.

@NORMALE

@NORMALE(x;*[moyenne]*;*[écart]*;*[type]*;*[domaine]*) évalue la fonction de distribution normale appliquée à x.

Arguments

x est la valeur critique (ou valeur de seuil) de la variable aléatoire de la distribution normale. x est un nombre quelconque. Toutefois, si x est une valeur négative, 1-2-3 la convertit en sa valeur absolue.

moyenne est un argument facultatif qui spécifie la valeur moyenne de la distribution. C'est une valeur positive ou nulle. 1-2-3 utilise 0 si elle est omise.

écart est un argument facultatif qui spécifie l'écart-type de la distribution. C'est une valeur positive ou nulle. 1-2-3 utilise 1 si elle est omise.

type est un argument facultatif qui détermine le type du résultat renvoyé par la fonction @NORMALE.

type @NORMALE calcule

- | | |
|---|---|
| 0 | la fonction de distribution cumulative ; valeur par défaut si vous omettez l'argument |
| 1 | l'inverse de la fonction de distribution cumulative |
| 2 | la fonction de densité de probabilité |

Si vous spécifiez un des arguments facultatifs, vous devez obligatoirement spécifier tous les arguments précédents.

Remarques

@NORMALE renvoie un résultat exact à $\pm 7,5 \cdot 10^{-8}$ près pour la distribution cumulative et à $\pm 4,5 \cdot 10^{-4}$ pour sa fonction inverse.

Exemples

@NORMALE(1,96) = 0,9750

@NORMALE(0.975;0;1;1) = 1.96

@NORMALE(1.96;0;1;2) = 0.058441

Fonctions similaires

@DISTCHI évalue la distribution de Chi-carré. @DISTF calcule la distribution de F, @POISSON celle de Poisson, et @DISTT celle de Student.

@NOMCHAMP

@NOMCHAMP(*cellule*) renvoie le nom du champ dans lequel se situe *cellule*.

Arguments

cellule est une adresse de cellule ou le nom d'un champ unicellulaire.

Remarques

Si vous spécifiez une cellule commune à plusieurs champs nommés, 1-2-3 renvoie le premier nom de champ rencontré.

Si *cellule* ne se trouve pas dans un champ nommé, @NOMCHAMP renvoie ERR.

La fonction @NOMCHAMP ne peut être utilisée que si *cellule* est située dans un classeur chargé en mémoire.

Exemples

@NOMCHAMP(A:A2) renvoie VENTES si A:A2 se trouve dans le champ nommé VENTES.

@NOMCHAMP(@POINTCELL("adresse")) renvoie le nom du champ contenant la cellule actuelle.

@INFOSCENARIO et @INFOVERSION

@INFOSCENARIO(*option;scénario;[id-utilisateur]*) renvoie des informations sur un groupe de versions.

@INFOVERSION(*option;champ;version;[id-utilisateur]*) renvoie des informations sur une version.

Arguments

option est un argument texte indiquant le type d'information à renvoyer.

<i>option</i>	1-2-3 renvoie
créé par	le nom de l'auteur de la version ou du groupe de versions
modifié par	le nom de la dernière personne ayant modifié la version ou le groupe de versions
créé-le	la date et l'heure de création de la version ou du groupe de versions, sous la forme d'une <u>date</u> et <u>nombre date</u>
modifié-le	la date et l'heure de dernière modification de la version ou du groupe de versions, sous la forme d'un nombre date et temps
note	le commentaire associé à la version ou au groupe de versions ; 1-2-3 le tronque s'il dépasse 512 octets
caché	0 (faux) si la version ou le groupe de versions n'est pas caché ou 1 (vrai) dans le cas contraire
protégé	0 (faux) si la version ou le groupe de versions n'est pas protégé ou 1 (vrai) dans le cas contraire

scénario/version est le nom du groupe de versions ou de la version. Si plusieurs groupes de versions ou versions possèdent le même nom, 1-2-3 choisit le/la plus récent(e).

id-utilisateur est le nom de l'utilisateur ayant créé la version ou le groupe de versions. 1-2-3 se base sur cet argument pour choisir une version ou un groupe de versions.

champ est le nom du champ contenant la version. *champ* doit correspondre à un champ nommé existant.

Exemples

@INFOSCENARIO("note";"Cas optimal";"Antoine Marchand") renvoie le commentaire associé au dernier groupe de versions intitulé Cas optimal et créé par Antoine Marchand.

@INFOSCENARIO("créé-par";"Ventes") renvoie le nom de l'utilisateur ayant créé le dernier groupe de versions intitulé Ventes.

@VERSIONINFO("créé-le";"VENTES";"Cas optimal";"Antoine Marchand") renvoie la date et l'heure à laquelle Antoine Marchand a créé la dernière version du champ VENTES intitulée Cas optimal .

@VERSIONINFO("modifié";"VENTES";"Gadget") renvoie la date et l'heure de la dernière modification de la version Gadget du champ VENTES.

@DERNIERSCENARIO

@DERNIERSCENARIO (*fichier*) renvoie le nom du dernier groupe de versions dans un classeur affiché d'un fichier.

Argument

fichier est un argument texte indiquant le nom du fichier classeur, extension comprise. A moins que le fichier classeur se trouve dans le répertoire actuel, vous devez également spécifier sa voie d'accès.

Remarque

Si aucun groupe de version n'a été affiché dans le *fichier* et la session actuelle, @DERNIERSCENARIO renvoie la valeur ERR.

Exemple

@DERNIERSCENARIO("d:\projet\juillet.wk4") renvoie le nom du dernier groupe de versions consulté dans le fichier classeur juillet.wk4, situé dans le répertoire d:\projet.

@ERRMOY

@ERRMOY(*liste*) calcule l'erreur standard sur la moyenne de l'échantillon représenté par les valeurs de *liste*.

Arguments

liste peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Exemples

@ERRMOY(2;6;8;5;3;9;1;2) = 1,052209

Fonctions similaires

@MOYGEOM renvoie la moyenne géométrique des valeurs d'une liste et @MOYHARM, la moyenne harmonique.

@STD et @STDNUM calculent l'écart-type d'une population.

@TESTT

@TESTT(*champ1*;*champ2*;*[type]*;*[sens]*) effectue un test de *t* (test de Student) sur les valeurs des champs *champ1* et *champ2* et renvoie la probabilité associée.

Arguments

champ1 et *champ2* sont des champs qui ne contiennent que des valeurs numériques. Dans le cas contraire (libellés, formules de type chaîne, cellules vides) @TESTT renvoie ERR.

type est un argument facultatif qui détermine le type de test effectué par 1-2-3.

<u>type</u>	<u>1-2-3 effectue</u>
0	Un test de Student sur l'échantillon établi à partir de deux populations de même variance (<i>champ1</i> et <i>champ2</i> n'ont pas nécessairement le même nombre de cellules) ; valeur par défaut si vous omettez l'argument
1	Un test de Student sur l'échantillon établi à partir de deux populations de variances distinctes (<i>champ1</i> et <i>champ2</i> n'ont pas nécessairement le même nombre de cellules)
2	Un double test de Student sur <i>champ1</i> et <i>champ2</i> , qui doivent nécessairement avoir le même nombre de cellules.

sens est un argument facultatif qui spécifie la direction du test de Student.

<u>sens</u>	<u>1-2-3 exécute</u>
1	Un test de Student mono-directionnel
2	Un double test de Student (sur les deux directions)

Si vous spécifiez un argument facultatif, vous devez obligatoirement spécifier tous les arguments précédents.

Remarques

@TESTT renvoie un résultat exact à $\pm 3 \cdot 10^{-7}$.

Exemples

@TESTT

Fonctions similaires

@DISTT évalue la fonction de distribution de Student. @TESTCHI effectue un test de Chi-carré, @TESTF un test de *F* et @TESTZ un test de *Z*.

@VERSIONACTUELLE

@VERSIONACTUELLE(*champ*) renvoie le nom de la version actuelle de *champ*.

Arguments

champ est le nom ou l'adresse du champ dont vous souhaitez identifier la version.

Remarques

S'il aucune version actuelle n'existe, @VERSIONACTUELLE renvoie ERR.

Exemples

@VERSIONACTUELLE(PROFITS) renvoie le nom de la version actuelle du champ PROFITS.

@DONNEESVERSION

@DONNEESVERSION(*option*;*cellule*;*champ*;*version*;*[id-utilisateur]*) renvoie le contenu de la cellule spécifiée d'une version.

Arguments

option est un argument texte indiquant sous quelle forme 1-2-3 doit renvoyer le contenu de *cellule*.

<u>option</u>	<u>1-2-3 renvoie</u>
formule	une formule, sous forme de libellé, ou <u>ERR</u> si la cellule ne contient pas de formule
valeur	le résultat de la formule si la cellule contient une formule, une valeur ou un libellé si la cellule contient une valeur ou un libellé, une <u>chaîne vide</u> si la cellule est vierge

cellule est le nom ou l'adresse de la cellule dont 1-2-3 renvoie le contenu. *cellule* doit être située dans *champ*.

champ est le nom du champ contenant la version. *champ* doit être un champ nommé existant.

version est le nom de la version. Si plusieurs versions portent le même nom, 1-2-3 choisit la version la plus récente.

id-utilisateur est le nom de la personne ayant créé la version. 1-2-3 se base sur ce nom pour identifier la version à utiliser ou à supprimer. Si *id-utilisateur* a créé plusieurs versions de même nom, 1-2-3 choisit la version la plus récente.

Exemples

@DONNEESVERSION("formule";A:B12;VENTES;"Cas optimal") renvoie la formule contenue dans la cellule A:B12 de la version Cas optimal la plus récente du champ VENTES.

@DONNEESVERSION("valeur";A:B12;VENTES;"Cas optimal";"Armelle Binot") renvoie la valeur ou le libellé de la cellule A:B12 de la version Cas optimal la plus récente créée par Armelle Binot pour le champ VENTES.

@MOYPOND

@MOYPOND(*liste*) multiplie les valeurs des cellules correspondantes dans les divers champs de *liste*, additionne les produits, puis divise la somme par le nombre de valeurs dans *liste*. Le résultat est la moyenne pondérée des valeurs de *liste*.

Arguments

liste est une combinaison quelconque de champs de même taille et forme contenant des valeurs. Si les champs de *liste* n'ont pas le même nombre de lignes ou de colonnes, **@MOYPOND** renvoie ERR.

Remarques

Si les champs de *liste* sont des colonnes, **@MOYPOND** multiplie par lignes. Si les champs sont des lignes, **@MOYPOND** multiplie par colonnes. Si les champs s'étendent sur plus d'une colonne, **@MOYPOND** multiplie par lignes.

Exemples

Dans cet exemple, **@MOYPOND** est utilisé pour calculer la moyenne pondérée des commissions correspondant à trois ventes.

liste contient deux champs : le champ VENTES où sont enregistrés les montants des trois ventes effectuées, et le champ COMM contenant les commissions correspondantes, exprimées en pourcentage du montant de la vente :

VENTES	COMM
25.000	0,04
34.580	0,05
77.325	0,04

@MOYPOND(VENTES;COMM) = 44.784,62

Fonction similaire

@SOMMEPROD calcule la somme des produits des cellules de même position dans plusieurs champs.

@JOUROUV

@JOUROUV(*date-début*;jours;*[champ-congés]*;*[weekends]*) renvoie le nombre date correspondant à la date postérieure ou antérieure du nombre de jours *jours* à la *date-début*, en excluant les week-ends et, facultativement, les jours de congés.

Arguments

date-début est un nombre date.

jours est un entier. Utilisez un entier positif si la date à trouver est postérieure à *date-début*, un entier négatif si elle lui est antérieure.

champ-congés est un argument facultatif qui permet d'indiquer les jours à ne pas prendre en compte dans le calcul des jours ouvrables. Ce champ peut contenir une combinaison quelconque de nombres date, de formules générant des nombres date, et d'adresses ou de noms de champs contenant ce type de valeurs. Ces éléments doivent être séparés par le séparateur d'arguments. Si vous omettez cet argument, 1-2-3 exclut seulement les week-ends.

weekends est un argument facultatif qui permet d'indiquer, au moyen d'un argument texte, les numéros des jours que 1-2-3 doit considérer comme des jours de week-end. Les jours sont numérotés de 0 (pour lundi) à 6 (pour dimanche).

Ainsi "45" indique que les vendredis et samedis sont des jours de week-end. Si vous omettez cet argument, 1-2-3 utilise "56" : le samedi et le dimanche sont alors considérés comme des jours de week-end. Si vous souhaitez qu'aucun jour de week-end ne soit exclu du calcul, indiquez la valeur 7.

Remarques

Si vous souhaitez retenir l'argument *weekends* et exclure l'argument *congés*, indiquez l'adresse d'une cellule vierge pour *congés*.

Exemple

Vous souhaitez déterminer la date 30 jours ouvrés après le lundi 31 octobre 1994. Entrez des nombres date dans un champ appelé CONGES pour préciser que le 1er novembre et le 11 novembre sont des jours non travaillés.

L'argument *week-end* peut être omis du fait que, par défaut, 1-2-3 considère les samedis et dimanches comme des jours de week-end.

@JOUROUV(**@DATE**(94;10;31);30;CONGES) = 34680, ou lundi 12 décembre 1994

Fonctions similaires

@JOURS360 et **@J360** calculent le nombre de jours entre deux nombres date. **@JOURSOUVRES** calcule le nombre de jours entre date-début et date-fin en excluant les week-ends et les jours de congé. **@MOISDECALER** calcule la date en ajoutant ou retranchant un certain nombre de mois à une date spécifiée.

@XINDEX

@XINDEX(*champ*;entête-colonne;entête-ligne;[entête-feuille]) renvoie le contenu de la cellule à l'intersection de la colonne, de la ligne et, facultativement, de la feuille, repérées par leur en-tête.

Arguments

champ est une adresse ou un nom de champ.

entête-colonne est le contenu d'une cellule en première ligne de *champ*.

entête-ligne est le contenu d'une cellule en première colonne de *champ*.

entête-feuille est le contenu d'une cellule en première ligne et première colonne de *champ*. C'est un argument facultatif permettant de repérer une feuille dans un champ multifeuille. Si vous l'omettez, **@XINDEX** explore seulement la première feuille de *champ*.

entête-colonne, *entête-ligne* et *entête-feuille* peuvent être aussi bien des nombres que du texte.

Exemples

@XINDEX

Fonctions similaires

@CHOIX est utilisée en remplacement d'un tableau de consultation d'une seule ligne. **@CONSH** et **@CONSV** renvoient le contenu d'une cellule dans un tableau de consultation horizontale ou verticale. **@CORRESP** indique la position relative d'une cellule identifiée par son contenu. **@CONSMAX** renvoie la référence absolue de la cellule qui contient la valeur la plus élevée de liste-champs. **@CONSMIN** retourne la référence absolue de la cellule qui contient la valeur la plus basse de liste-champs.

@TESTZ

@TESTZ(*champ1*; *moyenne1*; *écart1*; [*sens*]; [*champ2*]; [*moyenne2*]; [*écart2*]) effectue un test de Z sur les populations représentées par les valeurs de *champ1* et de *champ2* et renvoie la probabilité associée.

Arguments

champ1 est un champ qui contient soit le premier ensemble, soit la totalité, des valeurs à tester.

moyenne1 est la moyenne connue de *champ1* et peut être une quelconque valeur.

écart1 est une valeur différente de zéro qui représente l'écart-type connu de la population *champ1*.

sens est un argument facultatif qui précise le type de test à réaliser.

<u>sens</u>	<u>1-2-3 exécute</u>
1	Un test de Z mono-directionnel
2 (défaut)	Un double test de Z (sur les deux directions)

champ2 est un argument facultatif qui contient les valeurs du second ensemble à tester.

moyenne2 est la moyenne connue de *champ2* et peut être une quelconque valeur.

écart2 est une valeur différente de zéro qui représente l'écart-type connu de la population *champ2*. Si cet argument facultatif est omis, 1-2-3 utilise 1.

Si vous spécifiez un argument facultatif, vous devez obligatoirement spécifier tous les arguments facultatifs qui le précèdent.

Remarque

@TESTZ renvoie un résultat exact à $\pm 7,5 \cdot 10^{-8}$ près.

Exemple

Le champ A1..A8 contient les valeurs suivantes : 12, 19, 21, 22, 18, 16, 15, 17. Si la moyenne est de 16 et l'écart type de la population, de 3,041381, alors :

@ZTEST(A1..A8;16;3,041381) = 0,081512

Fonctions similaires

@TESTCHI effectue un test de Chi-carré, @TESTF un test de *F* et @TESTT un test de Student (test de *t*).

@DATECHAINE

@DATECHAINE(*date*) convertit un nombre date en son équivalent date et l'affiche sous forme d'un libellé au format de date international par défaut.

Argument

date est un nombre date.

Remarque

Le format de date international par défaut peut être modifié au moyen des Paramètres régionaux (pays) de votre système d'exploitation.

Exemple

Si le format de date international par défaut est jj/mm/aa, **@DATECHAINE**(34635) renvoie le libellé 28/10/94.

Fonctions similaires

@DATVAL calcule le nombre date d'une date entrée sous forme de libellé. **@DATE** calcule le nombre date correspondant à une date spécifiée.

@DURATION, @SENSIBILITETITRE

@DURATION(*livraison*; *maturité*; *coupon*; *rendement*; [*fréquence*]; [*base*]) calcule la durée à l'origine d'un titre de 100 F à taux d'intérêt fixe et périodique, compte tenu du taux de rendement du marché.

@SENSIBILITETITRE(*livraison*; *maturité*; *coupon*; *rendement*; [*fréquence*]; [*base*]) calcule la sensibilité (ou durée modifiée) à l'origine d'un titre de 100 F à taux d'intérêt fixe et périodique, compte tenu du taux de rendement du marché.

Arguments

livraison est la date d'achat du titre, exprimée en *nombre date*.

maturité est la date d'échéance du titre, exprimée en nombre date. Si elle est inférieure ou égale à *livraison*, **@DURATION** et **@SENSIBILITETITRE** retournent **ERR**.

coupon est le taux d'intérêt nominal du titre et peut être n'importe quel nombre positif ou nul.

rendement est le taux de rendement du marché et peut être n'importe quel nombre positif ou nul.

fréquence est un argument facultatif permettant d'indiquer la fréquence de paiement des coupons :

<u><i>fréquence</i></u>	<u>Fréquence de paiement des coupons</u>
1	Annuelle
2	Semi-annuelle ; valeur par défaut si vous omettez l'argument
4	Trimestrielle
12	Mensuelle

base est un argument facultatif qui spécifie le type de calendrier à prendre en compte :

<u><i>base</i></u>	<u>Base de comptage des jours</u>
0	30/360 ; valeur par défaut si vous omettez l'argument
1	Réel/réel
2	Réel/360
3	Réel/365
4	Européen 30/360

Vous ne pouvez utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarque

La durée d'un titre est la durée moyenne pondérée des mouvements de trésorerie qu'il engendre. Les coefficients de pondération sont obtenus en divisant la valeur actuelle de chaque mouvement de trésorerie par la valeur actuelle de l'ensemble des mouvements de trésorerie.

Exemples

Une obligation de 100 F est livrable le 1er juillet 1993 et vient à échéance le 1er décembre 1998. Son taux d'intérêt nominal est de 5,50 %, et les coupons sont payés semestriellement. Le taux de rendement du marché est de 5,61 %. Le calendrier à prendre en compte est de type 30/360.

Pour déterminer la durée de cette obligation, entrez :

@DURATION(**@DATE**(93;7;1);**@DATE**(98;12;1);0,055;0,0561;2;0) = 4,734591

Pour déterminer la sensibilité de cette obligation, entrez:

@SENSIBILITETITRE(**@DATE**(93;7;1);**@DATE**(98;12;1);0,055;0,0561;2;0) = 4,605410

Fonctions similaires

@INTCOUR calcule les intérêts courus pour des titres à intérêt périodique. **@PRIX** calcule le prix de titres à intérêt périodique, d'une valeur nominale de 100 F. **@RENDEMENT** calcule le rendement de titres à intérêt périodique.

@ESTVIDE

@ESTVIDE(*emplacement*) détermine si *emplacement* est ou non une cellule vierge. La fonction renvoie 1 (vrai) si *emplacement* est une cellule vierge et 0 (faux) dans le cas contraire.

Arguments

emplacement est le nom ou l'adresse d'une cellule. Si *emplacement* est un champ, **@ESTVIDE** renvoie 0 (faux).

Exemples

@ESTVIDE(A1) = 1 si la cellule A1 est une cellule vierge.

@ESTVIDE(A1) = 0 si la cellule A1 contient une valeur numérique.

@ESTVIDE(A1) = 0 si la cellule A1 contient un libellé.

@ESTVIDE(A1) = 0 si la cellule A1 contient un préfixe de libellé.

@CONSMAX, @CONSMIN

@CONSMAX(*liste-champs*) renvoie le nom de fichier classeur et la référence absolue de la cellule qui contient la valeur la plus élevée de *liste-champs*.

@CONSMIN(*liste-champs*) retourne le nom de fichier classeur et la référence absolue de la cellule qui contient la valeur la plus basse de *liste-champs*.

Arguments

liste-champs peut être une quelconque liste de champs dont les noms ou adresses sont séparés au moyen du séparateur d'arguments.

Si *liste-champs* est un champ unicellulaire, veuillez malgré tout à entrer une adresse de champ. Par exemple, n'entrez pas A1, mais A1..A1.

1-2-3 ne tient pas compte des libellés et des cellules vierges de *liste-champs*.

Si aucune des cellules de *liste-champs* ne contient de valeur, **@CONSMAX** et **@CONSMIN** renvoient ND.

Exemple

Vous avez placé dans votre répertoire 1-2-3 les devis de trois entreprises (DEVIS1.WK4, DEVIS2.WK4, et DEVIS3.WK4). Ces devis ont été créés à partir du même modèle si bien que leur montant total figure dans le champ nommé TOTAL.

Pour obtenir l'adresse et le nom de fichier classeur de la cellule contenant le devis le plus élevé, entrez :

@CONSMAX(<<DEVIS1.WK4>>TOTAL;<<DEVIS2.WK4>>TOTAL;<<DEVIS3.WK4>>TOTAL)

Pour obtenir l'adresse et le nom de fichier classeur de la cellule contenant le devis le plus bas, entrez :

@CONSMIN(<<DEVIS1.WK4>>TOTAL;<<DEVIS2.WK4>>TOTAL;<<DEVIS3.WK4>>TOTAL)

Fonctions similaires

@CONSH et **@CONSV** retournent le contenu d'une cellule dans un tableau de consultation horizontal ou vertical.

@INDEX retourne le contenu d'une cellule située à l'intersection de la colonne et de la ligne identifiées par leur numéro de position dans le champ. **@CHOIX** est utilisée en remplacement d'un tableau de consultation d'une seule ligne. **@CORRESP** indique la position relative d'une cellule identifiée par son contenu. **@XINDEX** restitue le contenu d'une cellule située à l'intersection d'une ligne et d'une colonne identifiées par leur contenu.

@JOURSOUVRES

@JOURSOUVRES(*date-début*; *date-fin*; [*champ-congés*]; [*weekends*]) calcule le nombre de jours entre *date-début* et *date-fin* en excluant les week-ends et les jours de congé.

Arguments

date-début et *date-fin* sont des nombres date.

champ-congés est un argument facultatif permettant de préciser les jours à ne pas prendre en compte dans le calcul des jours ouvrés. *champ-congés* est le nom ou l'adresse d'un champ contenant des nombres date.

weekends est un argument texte facultatif qui permet d'indiquer, au moyen de numéros, les jours que 1-2-3 doit considérer comme des jours de week-end. Les jours sont numérotés de 0 (pour lundi) à 6 (pour dimanche).

Ainsi "45" indique que les vendredis et samedis sont des jours de week-end. Si vous omettez cet argument, 1-2-3 utilise "56" : le samedi et le dimanche sont alors considérés comme des jours de week-end. Si vous souhaitez qu'aucun jour de week-end ne soit exclu du calcul, indiquez la valeur 7.

Vous ne pouvez utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Remarques

@JOURSOUVRES intègre *date-début* et *date fin* dans le résultat.

Si vous souhaitez retenir l'argument *weekends* et exclure l'argument *congés*, donnez l'adresse d'une cellule vierge pour ce dernier.

Exemple

Vous souhaitez déterminer le nombre de jours ouvrés que comporte la période comprise entre le mardi 1er novembre 1994 et le jeudi 1er décembre 1994. Entrez des nombres date dans un champ appelé CONGES pour préciser que le 1er novembre et le 11 novembre sont des jours non travaillés. L'argument *week-end* peut être omis du fait que, par défaut, 1-2-3 considère les samedis et dimanches comme des jours de week-end.

@JOURSOUVRES(@DATE(94;11;1);@DATE(94;12;1);CONGES) = 21

Fonctions similaires

@JOURS360 et @J360 calculent le nombre de jours entre deux nombres date. @JOURS calcule le nombre de jours entre deux dates, sur la base du calendrier spécifié. @JOUROUV calcule la date en ajoutant ou retranchant un certain nombre de jours à une date spécifiée, sans tenir compte des week-ends et des jours de congé.

@MOISDECALER calcule la date en ajoutant ou retranchant un certain nombre de mois à une date spécifiée.

@MOISDECALER

@MOISDECALER(*date-départ*;mois;*[jour-du-mois]*;*[base]*) calcule le nombre date de la date qui est antérieure ou postérieure à *date-départ* d'un nombre de *mois* spécifié.

Arguments

date-départ est un nombre date.

mois est un nombre entier. Utilisez un entier positif ou négatif selon que vous souhaitez ajouter ou retrancher des mois à *date-départ*.

jour-du-mois est un argument facultatif permettant de préciser pour quel jour du mois la fonction @MOISDECALER doit calculer un résultat :

<u>jour du mois</u>	<u>@MOISDECALER renvoie</u>
0	Une date qui tombe le même jour du mois que <i>date-départ</i> . Si une telle date n'existe pas (ce qui est notamment le cas si la date cherchée est postérieure d'un mois à une <i>date-départ</i> fixée au 30 janvier 1994, le mois de février ne comportant que 28 jours), @MOISDECALER renvoie la date du dernier jour du mois. Valeur par défaut si vous omettez l'argument.
1	Une date qui est le premier jour du mois.
2	Une date qui est le dernier jour du mois.

base est un argument facultatif qui permet de préciser le type de base de comptage des jours à utiliser :

<u>base</u>	<u>Base de comptage des jours</u>
0	30/360
1	Réel/réel ; valeur par défaut si vous omettez l'argument
2	Réel/360
3	Réel/365
4	Européen 30/360

Vous ne pouvez utiliser un argument facultatif sans utiliser les arguments facultatifs qui le précèdent.

Exemple

Vous souhaitez déterminer la date du dernier jour du mois qui suit le jeudi 7 avril 1994 :

@MOISDECALER(@DATE(94;4;7);1;2) = 34485, soit le mardi 31 mai 1994.

Fonctions similaires

@JOURS360 et @J360 calculent le nombre de jours entre deux nombres date. @JOUROUV calcule la date en ajoutant ou retranchant un certain nombre de jours à une date spécifiée, sans tenir compte des week-ends et des jours de congé. @JOURSOUVRES calcule le nombre de jours entre deux dates, en excluant les week-ends et les jours de congé.

@ALIGNNECHAINE

@ALIGNNECHAINE(*texte*;*longueur*;*[alignement]*) renvoie un libellé dont le nombre de caractères est égal à *longueur*. Ce libellé est composé d'un *texte* et d'un nombre suffisant d'espaces pour que l'alignement de *texte* soit conforme à l'argument *alignement*.

Arguments

texte peut être un quelconque argument texte.

longueur peut être un nombre entier compris entre 1 et 512. Si l'argument *longueur* est inférieur au nombre de caractères de *texte*, @ALIGNNECHAINE renvoie *texte*.

alignement est un argument facultatif qui précise l'alignement du texte :

<u><i>alignement</i></u>	<u>1-2-3 aligne <i>texte</i></u>
0	A gauche du nombre total de caractères défini par <i>longueur</i> ; valeur par défaut si vous omettez l'argument.
1	Au centre du nombre total de caractères défini par <i>longueur</i> . Si le nombre de caractères espace est impair, 1-2-3 place l'espace supplémentaire sur la gauche de <i>texte</i> .
2	A droite du nombre total de caractères défini par <i>longueur</i> .

La plupart des polices utilisées sous Windows sont des polices proportionnelles dans lesquelles le caractère espace est généralement plus étroit que les lettres.

Exemples

Dans les exemples ci-dessous, • représente le caractère espace.

@ALIGNNECHAINE("Compte de résultats";26) = Compte de résultats••••••••

@ALIGNNECHAINE("Compte de résultats";26;1) = ••••Compte de résultats••••

@ALIGNNECHAINE("Compte de résultats";26;2) = ••••••••Compte de résultats

Fonction similaire

@SUPPRESPACE supprime de la chaîne les espaces de début et de fin ainsi que les espaces superflus.

@SOMMENEUTIFIS, @SOMMEPOSITIFIS

@SOMMENEUTIFIS(*liste*) additionne uniquement les valeurs négatives de *liste*.

@SOMMEPOSITIFIS(*liste*) additionne uniquement les valeurs positives de *liste*.

Argument

L'argument *liste* peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques, et adresses ou noms de champ contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par le séparateur d'arguments.

Exemples

@SOMMENEUTIFIS(-2;21;5;12;-2;-7) = -11

@SOMMEPOSITIFIS(-2;21;5;12;-2;-7) = 38

Fonctions similaires

@SOMME additionne une liste de valeurs. @BSOMME totalise les valeurs répondant aux critères spécifiés.

@SOUSTOTAL permet de constituer des sous-totaux destinés à être additionnés au moyen de @TOTALGEN.

@VALFUTMONTANT

@VALFUTMONTANT(*principal;taux;périodes;[fréquence]*) renvoie la valeur capitalisée d'un montant forfaitaire investi à un taux donné pour un nombre de périodes donné.

Arguments

principal représente le montant initialement investi et peut être une valeur quelconque.

taux représente le taux d'intérêt périodique et peut être n'importe quelle valeur décimale ou pourcentage.

périodes représente le nombre de périodes d'investissement et peut être une valeur quelconque.

fréquence est un argument facultatif qui permet d'indiquer la fréquence de versement par période. Les nombres non entiers sont tronqués. La fréquence par défaut est 1.

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments qui le précèdent.

Remarques

Si *périodes* et *fréquence* sont égaux à 1, **@VALFUTMONTANT** calcule un taux simple.

Exemple

Supposons que pour les 10 ans de votre enfant, vous déposez 1 000 F sur un compte rémunéré à 10 % par an. En utilisant **@VALFUTMONTANT**, vous pouvez connaître le montant du compte lorsque votre enfant aura atteint l'âge de 18 ans.

@VALFUTMONTANT(1000;10%;8) = 2143,60 F

Fonctions @ similaires

@VALFUTMONTANT ne concerne qu'un seul versement. **@VALFUT** et **@VALFUTBIS** renvoient la valeur capitalisée d'une annuité, c'est à dire, une série d'un ou plusieurs versements. **@VALACTMONTANT** renvoie la valeur actuelle d'un montant forfaitaire placé à un taux d'intérêt donné pour un nombre de périodes données à recevoir dans le futur.

@VALACTMONTANT

@VALACTMONTANT(*valeur-future*;taux;*périodes*;*[fréquence]*) renvoie la valeur actuelle d'un montant forfaitaire placé à un taux donné pour un nombre de périodes données à recevoir dans le futur.

Arguments

valeur-future représente le montant à percevoir dans le futur et peut être une valeur quelconque.

taux représente le taux périodique et peut être une valeur décimale ou un pourcentage quelconque.

périodes représente le nombre de périodes dans le futur une fois le principal perçu et peut être une valeur quelconque.

fréquence est un argument facultatif qui permet d'indiquer la fréquence de versement par période. Les nombres non entiers sont tronqués. La fréquence par défaut est 1.

Remarque

Si *périodes* et *fréquence* sont égaux à 1, **@VALACTMONTANT** calcule un intérêt simple.

Exemples

Supposons que vous avez besoin de 10 000 francs pour frais scolaires, à payer dans les dix années à venir et que vous pouvez recevoir 10 % d'intérêt sur cet argent. Vous pouvez utiliser **@VALACTMONTANT** pour déterminer le montant forfaitaire à investir aujourd'hui :

@VALACTMONTANT(10000,10%,10) = 3 855,43 francs

Fonctions @ similaires

@VALACT et **@VALACTBIS** renvoient la valeur actuelle d'une annuité, c'est à dire, une série d'un ou plusieurs versements. **@VALFUTMONTANT** renvoie la valeur capitalisée d'un montant forfaitaire investi à un taux donné pour un nombre de périodes donné.

@XAMORT

@XAMORT(*estimation;cashflow;dates*) renvoie le taux de rentabilité interne d'une série de mouvements de trésorerie.

Arguments

estimation est une valeur décimale ou un pourcentage représentant le taux de rentabilité prévu. *estimation* est généralement un pourcentage compris entre 0 (0 %) et 1 (100 %). 10 % est souvent une bonne estimation. Lorsque les calculs portent sur des fonds importants, la valeur d'*estimation* doit être la plus précise possible.

cashflow est le nom ou l'adresse d'un champ contenant les mouvements de trésorerie. Les nombres négatifs sont considérés comme des sorties de fonds et les nombres positifs comme des rentrées de fonds. En général, le premier montant de trésorerie du champ doit être un nombre négatif (sortie de fonds), car il représente l'investissement. 1-2-3 affecte la valeur 0 aux cellules vierges et aux libellés figurant dans le champ et les intègre dans ses calculs.

dates est l'adresse ou le nom du champ contenant les dates auxquelles le mouvement de trésorerie est intervenu. Chaque date peut correspondre à n'importe quel nombre date et à la date du mouvement de trésorerie correspondant dans le champ *cashflow*. Les dates doivent être dans l'ordre croissant.

Les champs des arguments *cashflow* et *dates* doivent être de la même taille, c'est à dire avoir le même nombre de cellules. Les cellules des deux champs sont associées deux à deux par ordre dans le champ. Les champs sont classés du haut vers le bas (par lignes), puis de gauche à droite (par colonnes), puis par feuilles. Si les champs de *cashflow* et *dates* n'ont pas la même taille, @XAMORT renvoie ERR.

Remarques

Les mouvements de trésorerie peuvent être réalisés à intervalles irréguliers. Chaque mouvement de trésorerie de *cashflow* est associé à une date dans *dates*. Le premier mouvement de trésorerie et la première date déterminent le début de l'estimation. Le premier mouvement de trésorerie n'est pas escompté. Les mouvements de trésorerie suivants sont escomptés sur la base du taux d'escompte annuel et leur périodicité, comme indiqué par la date correspondante.

@AMORT escompte le premier mouvement de trésorerie. @XAMORT ne le fait pas et utilise la première date comme début du programme. Utilisez la valeur 0 pour le premier mouvement de trésorerie dans @XAMORT simuler la convention d'escompte de @AMORT.

@XAMORT permet des mouvements de trésorerie à intervalles irréguliers. @AMORT considère que les intervalles sont réguliers.

Exemples

Vous investissez la somme de 50 000 francs le 13 septembre 1996 et recevez les versements suivants : 1 000 francs le 31 janvier 1997 et 53 000 le 14 juin 1997. Le champ ci-dessous affiche ces données :

ESTIMATION	FONDS	DATES
0.10	-50000	13/09/96
	1000	31/01/97
	53000	14/06/97

@XAMORT(ESTIMATION;FONDS;DATES) = 10,90 %

Fonctions @ similaires

@XVAN renvoie la valeur actuelle nette d'une série de mouvements de trésorerie. @AMORT calcule le taux de rentabilité interne d'une série de mouvements de trésorerie générés par un investissement. @AMORTMOD calcule le taux de rentabilité interne modifié.

@XVAN

@XVAN(*taux;cashflow;dates*) renvoie la valeur actuelle nette d'une série de mouvements de trésorerie.

Arguments

taux peut être n'importe quelle valeur supérieure à -1 et représente le taux d'escompte.

cashflow est le nom ou l'adresse du champ contenant les mouvements de trésorerie. Les nombres négatifs sont considérés comme des sorties de fonds et les nombres positifs comme des rentrées de fonds. En général, le premier montant de trésorerie du champ doit être un nombre négatif (sortie de fonds), car il représente l'investissement. 1-2-3 affecte la valeur 0 aux cellules vierges et aux libellés figurant dans le champ et les intègre dans ses calculs.

dates est l'adresse ou le nom du champ contenant les dates auxquelles le mouvement de trésorerie est intervenu. Chaque date peut correspondre à n'importe quel nombre date et à la date du mouvement de trésorerie correspondant dans le champ *cashflow*. Les dates doivent être dans l'ordre croissant.

Les champs des arguments *cashflow* et *dates* doivent être de la même taille, c'est à dire avoir le même nombre de cellules. Les cellules des deux champs sont associées deux à deux par ordre dans le champ. Les champs sont classés du haut vers le bas (par lignes), puis de gauche à droite (par colonnes), puis par feuilles. Si les champs de *cashflow* et *dates* n'ont pas la même taille, @XVAN renvoie ERR.

Remarques

Les mouvements de trésorerie peuvent être réalisés à intervalles irréguliers. Chaque mouvement de trésorerie de *cashflow* est associé à une date dans *dates*. Le premier mouvement de trésorerie et la première date déterminent le début de l'estimation. Le premier mouvement de trésorerie n'est pas escompté. Les mouvements de trésorerie suivants sont escomptés sur la base du taux d'escompte annuel et leur périodicité, comme indiqué par la date correspondante.

Exemples

Cet exemple utilise la fonction @XVAN pour escompter au cours du jour (1er janvier 1996) une série de distributions irrégulières investies à un taux annuel de 11,5 %. Cet exemple utilise les données suivantes, dans lesquelles *cashflow* représente une liste de mouvements de trésorerie dans un champ nommé DISTRIBUTIONS et *dates* la liste des dates auxquels les mouvements interviennent, enregistrée dans un champ nommé DATES :

DISTRIBUTIONS	DATES
0,00 F	01/01/96
250000,00 F	04/01/96
250000,00 F	05/01/96
300000,00 F	06/01/96
500000,00 F	07/01/96
600000,00 F	08/01/96
900000,00 F	09/01/96
300000,00 F	10/01/96
250000,00 F	11/01/96
750000,00 F	01/01/97

@XVAN(0?115;DISTRIBUTIONS;DATES) = 3 821 809,20 F

Fonctions @ similaires

@VAN calcule la valeur actuelle nette d'un investissement, en déduisant la valeur actuelle de la valeur capitalisée.

@XAMORT renvoie le taux de rentabilité interne d'une série de mouvements de trésorerie.

@COMPRIENTRE

@COMPRIENTRE(*valeur*;*limite1*;*limite2*;*[inclusion]*) vérifie si l'argument *valeur* est compris entre les arguments *limite1* et *limite2*. @COMPRIENTRE renvoie 1 (vrai) si *valeur* est compris entre *limite1* et *limite2* et 0 (faux) dans le cas contraire.

Arguments

valeur, *limite1* et *limite2* peuvent être des valeurs numériques quelconques ou du texte.

inclusion est un argument facultatif et peut être l'un des éléments suivants :

<u><i>n</i></u>	<u>Description</u>	<u>Algorithme</u>
0	Inclut <i>limite1</i> et <i>limite2</i> (par défaut)	@MIN(<i>limite1</i> , <i>limite2</i>) <= <i>valeur</i> <= @MAX(<i>obligation1</i> , <i>limite2</i>)
1	Inclut la plus petite valeur de <i>limite1</i> et <i>limite2</i> et exclut la partie maximale	@MIN(<i>limite1</i> , <i>limite2</i>) <= <i>valeur</i> < @MAX(<i>limite1</i> , <i>limite2</i>)
2	Exclut la partie minimale de <i>limite1</i> et <i>limite2</i> et inclut la plus grande	@MIN(<i>limite1</i> , <i>limite2</i>) < <i>valeur</i> <= @MAX(<i>limite1</i> , <i>limite2</i>)
3	Exclut <i>limite1</i> et <i>limite2</i>	@MIN(<i>limite1</i> , <i>limite2</i>) < <i>valeur</i> < @MAX(<i>limite1</i> , <i>limite2</i>)

Remarques

Si le type de données de *limite1* est différent du type de données de *limite2*, @COMPRIENTRE renvoie ERR.

Si le type de données de *limite1* ou *limite2* est différent du type de données de *valeur*, @COMPRIENTRE renvoie 0.

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments qui le précèdent.

Exemples

@COMPRIENTRE(123;100;200) = 1

@COMPRIENTRE("ABC";"AAA";"BBB") = 1

@COMPRIENTRE(100;100;200;0) = 1

@DECILE

@DECILE(*x*;*champ*) renvoie un décile donné.

Arguments

x est un nombre entier compris entre 0 et 10 inclus.

<u>x</u>	<u>Renvoi</u>	<u>Correspondant à</u>
0	Minimum	@POURCENTILE(0,0; <i>champ</i>) ou @MIN(<i>champ</i>)
1	Premier décile	@POURCENTILE(0,1; <i>champ</i>)
2	Deuxième décile	@POURCENTILE(0,2; <i>champ</i>)
3	Troisième décile	@POURCENTILE(0,3; <i>champ</i>)
4	Quatrième décile	@POURCENTILE(0,4; <i>champ</i>)
5	Médiane	@POURCENTILE(0,5; <i>champ</i>) ou @MEDIANE(<i>champ</i>)
6	Sixième décile	@POURCENTILE(0,6; <i>champ</i>)
7	Septième décile	@POURCENTILE(0,7; <i>champ</i>)
8	Huitième décile	@POURCENTILE(0,8; <i>champ</i>)
9	Neuvième décile	@POURCENTILE(0,9; <i>champ</i>)
10	Maximum	@POURCENTILE(1,0; <i>champ</i>) ou @MAX(<i>champ</i>)

champ est l'adresse ou le nom du champ contenant les valeurs.

Remarques

@DECILE calcule des cellules vierges et des cellules contenant des libellés comme égales à 0.

Exemples

Un champ nommé DONNEES contient les éléments suivants :

34
12
60
128
67
350
206

@DECILE(0;DONNEES) = 12,0

@DECILE(1;DONNEES) = 25,2

@DECILE(2;DONNEES) = 39,2

Fonctions @ similaires

@POURCENTILE calcule le *x*ème pourcentile échantillon sur les valeurs de *champ*. @QUARTILE renvoie un quartile donné.

@MIN détermine la valeur minimale dans une liste. @MAX détermine la valeur maximale dans une liste.

@MEDIANE renvoie la valeur médiane dans une liste.

@PREVISION

@PREVISION (*x;champ-y;champ-x*) renvoie une valeur prévisionnelle pour *x*, sur la base d'une tendance linéaire entre des valeurs dans *champ-y* et *champ-x*.

Arguments

x peut être n'importe quelle valeur et représente la valeur à prévoir.

champ-y est l'adresse ou le nom d'un champ et représente l'ensemble des valeurs dépendantes. Si *champ-y* contient des données non numériques, @PREVISION renvoie ERR.

champ-x est l'adresse ou le nom d'un champ et représente l'ensemble des valeurs indépendantes. Si *champ-x* contient des données non numériques, @PREVISION renvoie ERR.

champ-y et *champ-x* doivent être de la même taille ; c'est à dire, contenir le même nombre de cellules. Les cellules des deux champs sont associées par 2 en fonction de leur ordre dans le champ. Dans les champs , l'ordre s'étend du haut vers le bas (par lignes), puis de gauche à droite (par colonnes), puis par feuilles. Si *champ-y* et *champ-x* ne sont pas de la même taille, @PREVISION renvoie ERR.

Remarques

Notez que ces modes de calcul sont aussi utilisés par la fonction @REGRESSION.

Exemples

Pour les données suivantes :

CHAMPY	CHAMPX
250	3
545	5
550	5
450	6
605	6
615	7

@PREVISION(-1;CHAMPY;CHAMPX) = -26,786

@PREVISION(0;CHAMPY;CHAMPX) = 56,786

Fonctions @ similaires

@REGRESSION effectue une régression linéaire multiple et renvoie le paramètre ainsi calculé.

@NSOMME

@NSOMME(*position*; *n*; *liste*) ajoute chaque nième valeur dans *liste*, à partir de *position*.

@NSOMME ignore ERR et ND dans les cellules/valeurs ignorées.

Arguments

liste peut contenir une combinaison quelconque des arguments suivants : nombres, formules numériques et adresses ou noms de champs contenant des nombres ou formules. Ces arguments doivent être séparés par des séparateurs d'arguments.

position peut être n'importe quel nombre entier positif.

n peut être n'importe quel nombre entier positif.

Remarques

@NSOMME renvoie la somme de (*position*), (*position* + *n*), (*position* + 2*n*), (*position* + 3*n*), ...dans *liste*.

@NSOMME(0;1;*liste*) renvoie le même résultats que **@SOMME**(*liste*).

Exemple

@NSOMME(1;3;B5..B15) renvoie la somme des valeurs figurant dans B6, B9, B12 et B15.

Fonctions @ similaires

@SOMME additionne les valeurs de *liste*. **@BSOMME** totalise les valeurs répondant aux critères spécifiés.

@SOUSTOTAL ajoute les valeurs d'une liste et permet d'indiquer quelles valeurs **@TOTALGEN** doit additionner.

@MEDIANENUM

@MEDIANENUM(*liste*) renvoie la valeur médiane de *liste*, en ignorant les cellules vides, libellés et formules générant des libellés.

Arguments

liste peut comporter un ou plusieurs des éléments suivants : nombres, formules numériques et adresses ou noms de champs contenant des nombres ou des formules numériques. Séparez les différents éléments de *liste* par les séparateurs d'arguments.

Remarques

La médiane est la valeur centrale dans *liste* telle qu'il existe un nombre égal de valeurs supérieures et inférieures à la médiane. Si *liste* contient un nombre impair de valeurs, **@MEDIANENUM** trie les valeurs et renvoie la valeur centrale. Si *liste* contient un nombre pair de valeurs, **@MEDIANENUM** trie les valeurs et renvoie la moyenne arithmétique des deux valeurs centrales.

@MEDIANENUM renvoie ERR lorsque le champ est vide ou lorsque le champ ne comporte que des libellés ou des formules générant des libellés.

Exemples

@MEDIANENUM(A2..A6) = 12, lorsque A2..A6 contient les valeurs 5, 12, 65, 82 et 9.

@MEDIANENUM(A1..A6) = 12, lorsque A1..A6 contient les valeurs 5, 12, 65, 82 et 9 et le libellé "Janvier".

(**@MEDIANENUM** ignore le libellé). **@MEDIANE** traite le libellé comme 0 et renvoie la valeur 10,5 pour ces données.

Fonctions @ similaires

@MEDIANE renvoie la valeur médiane dans une liste.

@QUARTILE

@QUARTILE(*x*;*champ*) renvoie un quartile donné.

Arguments

x est un nombre entier compris entre 0 et 4 inclus. Pour *x*, utilisez une des valeurs du tableau ci-dessous :

<i>x</i>	Renvoi	Correspondant à
0	Minimum	@POURCENTILE(0,0; <i>champ</i>) ou @MIN(<i>champ</i>)
1	Premier quartile	@POURCENTILE(0,25; <i>champ</i>)
2	Médiane	@POURCENTILE(0,50; <i>champ</i>) ou @MEDIANE(<i>champ</i>)
3	Troisième quartile	@POURCENTILE(0,75; <i>champ</i>)
4	Maximum	@POURCENTILE(1,00; <i>champ</i>) ou @MAX(<i>champ</i>)

champ est l'adresse ou le nom du champ contenant les valeurs.

Remarques

@QUARTILE compte les cellules vides et les cellules contenant des libellés comme 0.

Exemples

Données correspond à un champ avec le contenu suivant :

34
12
60
128
67
350
206

@QUARTILE(0;DONNEES) = 12

@QUARTILE(1;DONNEES) = 47

@QUARTILE(2;DONNEES) = 67

@QUARTILE(3;DONNEES) = 167

@QUARTILE(4;DONNEES) = 350

Fonctions @ similaires

@POURCENTILE calcule le xième pourcentile échantillon sur les valeurs de *champ*. @DECILE renvoie un décile donné.

@CHERCHEO

@CHERCHEO(*chaîne1*; *chaîne2*; *n*) recherche la position (en octets) de la première occurrence de *chaîne1* dans *chaîne2*, à partir de la position (en octets) du même caractère.

Arguments

chaîne1 et *chaîne2* peuvent être du texte mis en guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

n est un numéro de position en octets.

Remarques

La valeur ERR s'affiche lorsque @CHERCHEO ne trouve pas *chaîne1* dans *chaîne2*, ou quand *n* dépasse le nombre d'octets de *chaîne2* ou correspond à une valeur négative.

@CHERCHEO distingue les majuscules des minuscules et les caractères accentués de ceux qui ne le sont pas. Ainsi @CHERCHEO ne trouve pas l'argument *chaîne1* "e" dans le *chaîne2* "TOULOUSE".

@CHERCHEO est aussi utile, lorsque associé à @MILIEUO ou @SUBSTITUEO, il localise, extrait ou remplace du texte.

Exemples

@CHERCHEO("F";"Comptes Fournisseurs";0) = 8 car *chaîne1* F se trouve en 8ème position dans la chaîne Comptes Fournisseurs (*chaîne2*).

Fonctions @ similaires

@CHERCHE permet de déterminer la position calculée en caractères.

@MILIEUO

@MILIEUO(*chaîne*; *position-début*; *n*) extrait *n* octets de *chaîne*, à partir des données indiquées par l'argument *position-début* en octets.

Arguments

chaîne peut être du texte mis entre guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

position-début est un numéro de position en octets. Si *position-début* est supérieur à la longueur de *chaîne*, @MILIEUO génère une chaîne vide.

n est un entier positif quelconque ou nul. Si *n* est égal à 0, @MILIEUO génère une chaîne vide. Si *n* est supérieur à la longueur de *chaîne*, 1-2-3 renvoie tous les octets compris entre *position-début* et la fin de *chaîne*.

Remarques

@MILIEUO est utilisé pour les jeux de caractères à double octets (DBCS) tels que le japonais.

@MILIEUO compte les signes de ponctuation et les espaces comme des caractères.

Si vous ignorez la longueur de *texte*, indiquez un nombre *n* élevé. 1-2-3 ne tient pas compte des espaces supplémentaires et renvoie la totalité de *texte* en partant de *position-début*.

Si le premier octet du libellé renvoyé correspond à la dernière partie d'un caractère à double octets ou si le dernier octet du libellé renvoyé correspond à la première partie d'un caractère à double octets, @MILIEUO remplace le caractère incomplet par un espace.

Exemple

@MILIEUO("Compte courant crédité";7;7) = "courant"

Fonctions @ similaires

@MILIEU extrait *n* caractères de *chaîne*, à partir du caractère indiqué par l'argument *position-début*.

@DEBUTQ renvoie les *n* premiers octets de *chaîne* et @FINQ renvoie les *n* derniers octets de *chaîne*.

@DEBUTO

`@DEBUTO(chaîne;n)` renvoie les *n* premiers octets de *chaîne*.

Arguments

chaîne peut être du texte mis entre guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

n peut être un nombre entier positif ou nul. Si *n* est égal à 0, le résultat est une chaîne vide. Si *n* est supérieur à la longueur en octets de *chaîne*, `@DEBUTO` renvoie l'intégralité de *chaîne*.

Remarques

`@DEBUTO` compte les signes de ponctuation et les espaces comme des caractères.

`@DEBUTO` est utilisé pour les jeux de caractères à double octets (DBCS) tels que le japonais. Par exemple, si vous souhaitez insérer un libellé dans une rubrique d'une base de données externe avec 1-2-3 et que la rubrique a une longueur donnée en octets, vous pouvez extraire des parties du libellé pour ne pas dépasser la longueur autorisée.

Si le dernier octet du libellé renvoyé correspond à la première partie d'un caractère à double octets, `@DEBUTO` remplace le caractère incomplet par un espace.

Si *chaîne* ne comporte que des caractères un octet, `@DEBUTO` renvoie le même résultat que la fonction `@DEBUT`.

Exemple

`@DEBUTO("un octet";4) = "un o"`

Fonctions @ similaires

`@DEBUT` renvoie les *n* premiers caractères de *chaîne*. `@MILIEUQ` renvoie des octets de *chaîne*. `@FINQ` renvoie les *n* derniers octets de *chaîne*.

@FINO

@FINO(*chaîne*;n) renvoie les *n* derniers octets dans *chaîne*.

Arguments

chaîne peut être du texte mis en guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

n peut être un nombre entier positif ou nul. Si *n* est égal à 0, le résultat est une chaîne vide. Si *n* est supérieur à la longueur de *chaîne*, @FINO renvoie l'intégralité de *chaîne*.

Remarques

@FINO est utilisé pour un jeu de caractères à double octets (DBCS) tel que le japonais.

@FINO compte les signes de ponctuation et les espaces comme des caractères.

Si le premier octet du libellé renvoyé correspond à la dernière partie d'un caractère à double octets, @FINO remplace le caractère incomplet par un espace.

Utilisez conjointement @FINO et @CHERCHEO lorsque vous ne connaissez pas la valeur exacte de *n* ou lorsque *n* peut varier.

Exemple

@FINO("Un octet",5) = "octet"

Fonctions @ similaires

@FIN renvoie les *n* derniers caractères dans *chaîne*. @DEBUTO renvoie les *n* premiers octets dans *chaîne*.

@MILIEUQ renvoie des octets de *chaîne*.

@LONGO

@@LONGO(*chaîne*) compte les octets de *chaîne*.

Arguments

chaîne peut être du texte mis entre guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

Remarques

@LONGO compte les signes de ponctuation et les espaces comme des caractères.

@LONGO est utilisé pour les jeux de caractères à double octets (DBCS) tels que le japonais. Par exemple, si vous souhaitez insérer du texte dans une rubrique d'une base de données externe avec 1-2-3 et que la rubrique a une longueur donnée en octets, vous pouvez vérifier si votre texte dépasse cette longueur.

Exemple

@LONGO(A5&G12) = calcule le nombre total d'octets présents dans les cellules A5 et G12.

Fonctions @ similaires

@LONG compte les caractères de *chaîne*.

@SUBSTITUEO

@SUBSTITUEO(*chaîne-origine*;position;n;*nouvelle-chaîne*) remplace *n* octets de *chaîne-origine* par *nouvelle-chaîne*, à partir de *position*.

Arguments

chaîne-origine et *nouvelle-chaîne* peuvent être du texte mis entre guillemets, des formules texte ou les adresses ou les noms de cellules contenant des libellés ou des formules générant des libellés.

position est le numéro de position des données de *chaîne-origine*. Il peut être une valeur chaîne positive ou nulle. Si *position* est supérieur à la longueur en octets de *chaîne-origine*, **@SUBSTITUEO** ajoute *nouvelle-chaîne* à *chaîne-origine*

n peut être un nombre entier positif ou nul. Si *n* est égal à 0, **@SUBSTITUEO** ajoute *nouvelle-chaîne* à *position* sans supprimer les données de *chaîne-origine*

Remarques

@SUBSTITUEO traite les positions et les longueurs de chaîne en octets.

@SUBSTITUEO compte les signes de ponctuation et les espaces comme des caractères. Si vous utilisez **@SUBSTITUEO** pour ajouter ou insérer du texte, vous ne devez pas oublier d'inclure les espaces nécessaires.

Si les données remplacées dans *chaîne-origine* commencent et finissent au centre d'un caractère double octets, **@SUBSTITUEO** remplace chaque caractère double octets incomplet dans le texte renvoyé par un espace.

Utilisez conjointement @CHERCHEO et **@SUBSTITUEO** pour rechercher et remplacer un libellé ou pour calculer un argument *position* inconnu.

@SUBSTITUEO est utile lorsque vous devez remplacer un jeu de caractères par un autre. Par exemple, pour modifier les indicatifs de numéros de téléphone dans une base de données.

Exemple

@SUBSTITUEO(CELLULE;@CHERCHEO("-";CELLULE;0),1,"/") transforme le libellé dans CELLULE, 4-24 en 4/24.

Fonctions @ similaires

@SUBSTITUE remplace *n* caractères dans *chaîne-origine* par *nouvelle-chaîne*, à partir de *position*.

@CONVDATE

@CONVDATE(*date;type-données;type-sortie*) convertit une date Hijri (PAYS arabeS), Parsi (Iran) ou Hébreux (Israël) en une date grégorienne ou inversement.

Remarques

La fonction @CONVDATE est utilisée pour les langues bi-directionnelles et ne fonctionne que sur des ordinateurs prenant en charge des versions bi-directionnelles de 1-2-3 (par exemple : hébreux, parsi).

@TCHERCHE

TCHERCHE(*chaîne1*;*chaîne2*;*position-colonne*) recherche la position du caractère thaï logique dans *chaîne2* à laquelle 1-2-3 cherche la première occurrence de *chaîne1*, à partir du caractère thaï de *position-colonne*. Cette fonction utilise des caractères thaï logiques, constitués de 1 à 3 octets.

Arguments

chaîne1 et *chaîne2* sont du texte mis entre guillemets, des formules texte ou les noms ou les adresses de cellules contenant des libellés ou des formules générant des libellés.

position-colonne est un numéro de position.

Remarques

TCHERCHE est également utile combiné avec @TMLIEU ou @TSUBSTITUE pour localiser et extraire ou remplacer du texte.

Exemples

@TCHERCHE("ไทย","บริษัท ไทยซอฟต์แวร์ จำกัด",0) = 5

Fonctions @ similaires

@CHERCHE exécute les mêmes fonctions que @TCHERCHE mais est utilisé pour les chaînes de texte non thaï.

@TMILIEU

@TMILIEU(*chaîne*; *position*; *n*) extrait *n* caractères thaï logiques de *chaîne*, à partir du caractère thaï indiqué par l'argument *position*. Cette fonction utilise des caractères thaï logiques, constitués de 1 à 3 octets.

Arguments

chaîne est du texte mis entre guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

position est un numéro de position. Si *position* est supérieur à la longueur de *chaîne*, le résultat est une chaîne vide.

n peut être un nombre entier positif ou nul. Si *n* est égal à 0, le résultat est une chaîne vide. Si *n* est supérieur à la longueur de *chaîne*, @TMILIEU renvoie l'intégralité des caractères de *position* à la fin de *chaîne*.

Exemples

@TMILIEU("บริษัท วิทยุการบิน จำกัด",5,7) = วิทยุการบิน

Fonctions @ similaires

@MILIEU exécute les mêmes fonctions que @TMILIEU, mais est utilisé pour les chaînes de texte non thaï. @TFIN renvoie les *n* derniers caractères thaï de *chaîne*. @TDEBUT renvoie les *n* premiers caractères thaï logiques de *chaîne*.

@TDEBUT

@TDEBUT(*chaîne*; *n*) renvoie les *n* premiers caractères logiques Thaï de l'argument *chaîne*. Cette fonction utilise des caractères logiques Thaï, constitués de 1 à 3 octets.

Arguments

chaîne est du texte mis entre guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

n peut être un nombre entier positif ou nul. Si *n* est égal à 0, le résultat est une chaîne vide. Si *n* est supérieur à la longueur de *chaîne*, @TDEBUT renvoie l'intégralité de *chaîne*.

Exemples

@TDEBUT("บริษัท วิทยุคอมพ์ จำกัด",4) = บริษัท

Fonctions @ similaires

@DEBUT exécute les mêmes fonctions que @TDEBUT, mais est utilisé pour les chaînes de texte non Thaï.

@TMILIEU renvoie les caractères thaï de *chaîne*. @TFIN renvoie les *n* derniers caractères thaï de *chaîne*.

@TFIN

@TFIN(*chaîne*;n) renvoie les *n* derniers caractères thaï logiques dand *chaîne*. Cette fonction utilise des caractères thaï logiques , constitués de 1 à 3 octets.

Arguments

chaîne est du texte mis entre guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

n peut être un nombre entier positif ou nul. Si *n* est égal à 0, le résultat est une chaîne vide. Si *n* est supérieur à la longueur de *chaîne*, **@TFIN** renvoie l'intégralité de *chaîne*.

Exemples

@TFIN("บริษัท ไทยซอฟต์แวร์ จำกัด",4) = จำกัด

Fonctions @ similaires

@FIN exécute les mêmes fonctions que **@TFIN**, mais est utilisé pour les chaînes de texte non Thaï. **@TMILIEU** renvoie les caractères thaï de *chaîne*. **@TDEBUT** renvoie les *n* premiers caractères thaï logiques de *chaîne*

@TLONG

@TLONG(*chaîne*) compte les caractères thaï logiques de *chaîne*. Cette fonction utilise des caractères thaï logiques, constitués de 1 à 3 octets.

Arguments

chaîne est du texte mis entre guillemets, une formule texte ou le nom ou l'adresse d'une cellule contenant un libellé ou une formule générant un libellé.

Exemples

@TLONG("บริษัท วิทยุสื่อสาร จำกัด") = 17

Fonctions @ similaires

@LONG exécute les mêmes fonctions que @TLONG, mais est utilisé pour les chaînes de texte non thaï.

@TSUBSTITUE

TSUBSTITUE(*chaîne-origine*; *position*; *n*; *nouvelle-chaîne*) remplace les *n* caractères thaï logiques de l'argument *chaîne-origine* par ceux figurant dans *nouvelle-chaîne*, en partant de l'argument *position*. Cette fonction utilise des caractères logiques Thaï, constitués de 1 à 3 octets.

Arguments

chaîne-origine et *nouvelle-chaîne* peuvent être du texte mis entre guillemets, des formules texte ou le nom ou l'adresse de cellules comportant des libellés ou des formules générant des libellés.

position, numéro de position d'une colonne dans *chaîne-origine*, peut être une valeur positive ou nulle. Si *position* est supérieur à la longueur de *chaîne-origine*, TSUBSTITUE ajoute *nouvelle-chaîne* à *chaîne-origine*.

n peut être un nombre entier positif ou nul. Si *n* est égal à 0, TSUBSTITUE insère *nouvelle-chaîne* à *position* sans supprimer de colonnes dans *chaîne-origine*.

Exemples

@TSUBSTITUE("บริษัท จำกัด",5,0,"ไทยซอฟท์") = บริษัท ไทยซอฟท์ จำกัด

Fonctions @ similaires

@SUBSTITUE exécute les mêmes fonctions que @TSUBSTITUE, mais est utilisé pour les chaînes de texte non thaï.

@TNOMDATE

@TNOMDATE(*valeur-date*) convertit *valeur-date* au jour de la semaine thaï.

Arguments

valeur-date est un nombre date.

Exemples

@TNOMDATE(@DATE(91,8,16)) = ศุกร์

@DISTT

@DISTT(*chaîne-nombre*) convertit un argument *chaîne-nombre* constitué de chiffres arabes en une chaîne de caractères numériques thaï.

Arguments

chaîne-nombre est une chaîne contenant des chiffres arabes.

Exemples

@DISTT ("16 ๓๓ . 34") = ๑๖ ๓๓ . ๓๔

Fonctions @ similaires

@ENOMBRE convertit *chaîne-nombre* constitué de caractères numériques thaï en chaîne numérique arabe équivalente .

@ENOMBRE

@ENOMBRE(*chaîne-nombre*) convertit l'argument *chaîne-nombre* composé de caractères numériques thaï en une chaîne numérique arabe.

Arguments

chaîne-nombre est une chaîne contenant des caractères numériques thaï .

Remarques

Utilisez conjointement **@ENOMBRE** et **@VALEUR** pour convertir des données entrées sous la forme de chaînes numériques thaï en valeurs numériques pouvant être utilisées dans des calculs.

Exemples

@ENOMBRE ("๑๖ ๓๔ . ๓๔") = 16 ๓๔ . 34

Fonctions @ similaires

@TNOMBRE convertit *chaîne-nombre* composé des chiffres arabes en une chaîne numérique thaï.

@VALEUR convertit un nombre entré sous forme de texte en sa valeur correspondante.

@TNBRECHaine

@TNBRECHaine (*nombre*) convertit *nombre* en chaîne de nombres thaï.

Arguments

nombre est un nombre entier ou une virgule flottante

Exemples

@TNBRECHaine(1000000) = หนึ่งล้านบาท

Fonctions @ similaires

@NBRECHaine convertit *nombre* en un nombre indiqué (enregistré en tant que un libellé) dans un format non thaï.

@TDATECHAINE

@TDATECHAINE(*valeur-date*) convertit *valeur-date* en une chaîne de date thaï de format court.

Arguments

valeur-date est un nombre date.

Exemples

@TDATECHAINE (@DATE(91,8,16)) = 16 สค. 34

Fonctions @ similaires

@TDATECHAINE convertit *valeur-date* en une chaîne de date thaï de format long.

@TLDATECHAINE

@TLDATECHAINE(*valeur-date*) convertit un argument *valeur-date* en chaîne date thaï de format long.

Arguments

valeur-date est un nombre date.

Exemples

@TLDATECHAINE (@DATE(91,8,16)) = 16 สิงหาคม 2534

Fonctions @ similaires

@TDATECHAINE convertit *valeur-date* en une chaîne date thaï de format court.

@SINT

@SINT(*principal;taux;périodes;période*) calcule la part d'intérêt d'un remboursement périodique dans lequel la part principale est la même pour chaque période.

Arguments

principal représente la valeur de l'emprunt et peut être une valeur quelconque.

taux représente le taux d'intérêt de l'emprunt et peut être n'importe quelle valeur décimale ou pourcentage supérieur à -1.

périodes représente le nombre de périodes de remboursement et peut être n'importe quel nombre entier positif. Les nombres non entiers sont tronqués.

période représente la période de remboursement pour laquelle vous souhaitez déterminer la part d'intérêt. *période* peut être n'importe quel nombre entier positif inférieur ou égal à *périodes* . Les nombres non entiers sont tronqués. *période* commence à 0 (zéro).

Remarques

La période utilisée pour calculer *taux* doit être la même que celle utilisée pour *périodes*. Par exemple, si les remboursements sont mensuels, entrez le taux d'intérêt et la période par incréments mensuels. Généralement, vous devez diviser le taux d'intérêt par 12 et multiplier le nombre d'années dans *périodes* par 12.

Exemples

Vous avez emprunté 8 000 francs sur 3 ans, à un taux d'intérêt annuel de 10,5 % sur la base de remboursements mensuels. Pour déterminer la part d'intérêt du premier remboursement :

@SINT(8000;0,105/12;36;0) = 70 francs

Pour déterminer le montant du premier remboursement :

@SINT(8000;0,105/12;36;0) + 8000/36 = 292,22 francs

Fonctions @ similaires

@REMBOURINT calcule la part d'intérêt d'un remboursement périodique lorsque le versement est le même pour chaque période, mais que la part principale augmente tandis que l'intérêt diminue.

@REMBOURINT

@REMBOURINT(*principal*; *taux*; *périodes*; *période*) calcule les parts d'intérêts d'un versement périodique constant.

Arguments

principal représente la valeur de l'emprunt et peut être une valeur quelconque.

taux représente le taux d'intérêt de l'emprunt et peut être une valeur décimale ou un pourcentage supérieur à -1.

périodes représente le nombre de périodes de versements. *périodes* peut être n'importe quel nombre entier positif. Les nombres non entiers sont tronqués.

période représente la période de versement pour laquelle vous souhaitez déterminer les intérêts. *période* peut être n'importe quel nombre entier positif inférieur ou égal à *périodes*. Les nombres non entiers sont tronqués.

Remarques

La période utilisée pour calculer *taux* doit être la même que celle utilisée pour *périodes*. Par exemple, si les versements sont mensuels, entrez le taux d'intérêt et la période par incréments mensuels. Généralement, vous devez diviser le taux d'intérêt par 12 et multiplier le nombre d'années dans *périodes* par 12.

@INTVER(*principal*; *taux*; *périodes*; *période*; *période*; 0) renvoie le même résultat que

@REMBOURINT(*principal*; *taux*; *périodes*; *période*).

Exemples

Vous avez emprunté 8 000 francs sur 3 ans, à un taux d'intérêt annuel de 10,5 % sur la base de remboursements mensuels. Vos remboursements mensuels sont de 260,02 francs. Pour déterminer la part d'intérêt du premier versement :

@REMBOURINT(8000;0,105/12;36;1) = 70 francs

Fonctions @ similaires

@REMBOUR et @REMBOURBIS calculent le remboursement d'un emprunt (*principal*) à un taux d'intérêt donné pour un nombre de périodes de versement spécifié, selon une annuité ordinaire. @REMBOUR2 calcule le remboursement d'un emprunt à un taux d'intérêt donné pour un nombre de périodes de versement défini, selon une convention d'annuité due.

@SINT calcule la part d'intérêt d'un remboursement périodique d'un emprunt pour lequel le versement principal est le même pour chaque période de remboursement.

@INTVER calcule les intérêts cumulés correspondant au remboursement périodique d'un emprunt à un taux d'intérêt donné pour un nombre de périodes défini.

@DEMICAR

@DEMICAR(*libellé*) convertit les caractères japonais double octets dans *libellé* en caractères (ASCII) un octet correspondants.

Arguments

libellé est une chaîne qui contient des caractères japonais double octets à convertir. Si *libellé* n'est pas une chaîne, @DEMICAR renvoie ERR.

Remarques

@DEMICAR convertit les caractères, les nombres et les signes (tels que +, -, :) d'un jeu de caractères double octets (DBCS) en caractères ASCII correspondants. Les caractères ASCII et autres caractères auxquels ne correspondent pas de valeurs ASCII, ne sont pas convertis. @DEMICAR les renvoie sous leur forme d'origine. @DEMICAR ne convertit pas le jeu de caractères double octets Katakana en jeu de caractères un octet Katakana.

Fonctions @ similaires

@CARENTIER convertit des caractères un octet en caractères japonais double octets.

@CARENTIER

@CARENTIER(*libellé*) convertit des caractères (ASCII) un octet dans *libellé* en caractères double octets japonais correspondants.

Arguments

libellé est une chaîne qui contient les caractères ASCII à convertir. Si *libellé* n'est pas une chaîne, @CARENTIER renvoie ERR.

Remarques

@CARENTIER convertit des caractères ASCII dans *libellé* en caractères double octets correspondants. Les caractères appartenant au jeu de caractères double octets (DBCS) ne sont pas convertis et @CARENTIER les renvoie sous leur forme d'origine. @CARENTIER ne convertit pas le jeu de caractères un octet Katakana en jeu de caractères double octets Katakana.

Fonctions @ similaires

@DEMICAR convertit des caractères double octets japonais en caractères un octet.

@NBRECHaine

@NBRECHaine(*nombre*;*type*) convertit l'argument *nombre* dans le texte japonais du nombre, en utilisant le format spécifié dans *type*.

Arguments

nombre peut être un nombre arrondi quelconque compris entre 0 et 9 999 999 999 999. 1-2-3 arrondit *nombre* au nombre entier le plus proche. Si vous indiquez un nombre no compris entre ces deux chiffres, @NBRECHaine renvoie ERR.

type indique le format. Utilisez l'une des valeurs du tableau ci-dessous :

<u>typ</u>	<u>Renvoi</u>
1	Notation régulière de nombres Kanji
2	Notation formelle de nombres Kanji
3	Conversion des nombres en nombres Kanji un par un

Exemples

@TNBRECHaine(1234567890,1) = 十二億三千四百五十六万七千八百九十

@TNBRECHaine(1234567890,2) = 拾貳億參千四百五拾六万七千八百九拾

@TNBRECHaine(1234567890,3) = 一 二 三 四 五 六 七 八 九 〇

@INTCOUR2

@INTCOUR2(*règlement*; *maturité*; *coupon*; [*valeur-nominale*]; [*fréquence*]; [*émission*]; [*premier-intérêt*]; [*type*]) calcule les intérêts courus des obligations à versements d'intérêt périodique selon les conventions japonaises.

Arguments

règlement est la date de règlement de l'obligation, exprimée en nombre date.

maturité est la date de maturité de l'obligation, exprimée en nombre date. Si *maturité* est inférieur ou égal à *règlement*, @INTCOUR2 renvoie ERR.

coupon est le taux nominal annuel de l'obligation. *coupon* peut être n'importe quelle valeur positive ou nulle, exprimée sous forme de nombre décimal.

valeur-nominale est un argument facultatif qui permet de spécifier la valeur nominale de l'obligation, c'est-à-dire, le principal à payer à maturité. *valeur-nominale* est une valeur positive. Si vous l'omettez, 1-2-3 utilise 100.

fréquence est un argument facultatif qui permet d'indiquer la fréquence de versement des coupons par année. *fréquence* est l'une des valeurs du tableau ci-dessous :

<u>fréquence</u>	<u>Fréquence des versements de coupons</u>
1	Annuelle
2	Semi-annuelle (par défaut)

émission est un argument facultatif qui représente la date d'émission de l'obligation, exprimée en nombre date. Si *émission* est supérieur à *règlement*, @INTCOUR2 renvoie ERR.

premier-intérêt est un argument facultatif qui représente la première date d'intérêt de l'obligation, exprimée en nombre date. Si *premier-intérêt* est inférieur ou égal à *émission* ou supérieur à *maturité*, @INTCOUR2 renvoie ERR.

type est un argument facultatif qui indique le type d'obligation. *type* est l'une des valeurs du tableau ci-dessous :

<u>type</u>	<u>Description</u>
0	Obligation du gouvernement japonais ; par défaut si vous omettez l'argument
1	Obligation publique ou émise par une société privée

Vous ne pouvez pas utiliser un argument facultatif sans utiliser les arguments qui le précèdent.

Exemples

Vous avez acheté une obligation du gouvernement japonais de 100 000 Yens le 14 juin 1996. La date d'émission est le 1er mai 1990 et la date de maturité, le 20 juin 2000. Le taux du coupon est de 9 %, les coupons sont payés semi-annuellement, et la date des premiers intérêts était le 21 décembre 1990.

L'intérêt couru est le suivant :

@INTCOUR2(@DATE(96,6,14),@DATE(2000,6,20),0.09,100000,2,@DATE(90,5,1),@DATE(90,12,21)) = 4 339 Yens

Fonctions @ similaires

@INTCOUR calcule le taux d'intérêt couru des obligations à versements d'intérêt périodique. @PRIX2 calcule le prix d'obligations à intérêt périodique d'une valeur nominale de 100 Yens, selon les conventions japonaises.

@RENDEMENT2 renvoie le taux de rendement pour des obligations à intérêt périodique selon les conventions japonaises.

@PRIX calcule le prix d'une obligation rapporté à la valeur nominale 100 francs. @RENDEMENT calcule le taux de rendement d'obligations à intérêt périodique. @DURATION calcule la durée annuelle et @SENSIBILITETITRE calcule la durée annuelle modifiée des obligations à intérêt périodique.

@VALFUT2

@VALFUT2(*versements;taux;périodes*) calcule la valeur capitalisé d'un investissement, sur la base d'une série de *versements* égaux, placés à un *taux d'intérêt* périodique, sur un nombre de *périodes* de versement donné, selon une convention d'annuité due.

Arguments

versements représente le montant payé à chaque période et peut être une valeur quelconque.

périodes représente le nombre de périodes de versement. *périodes* peut être un nombre entier positif quelconque. Les nombres non entiers sont tronqués.

taux représente le taux d'intérêt périodique et peut être une valeur décimale ou un pourcentage supérieur à -1.

Remarques

La période utilisé pour calculer l'argument *taux* doit être la même période que celle utilisée pour *périodes*. Par exemple, si les versements sont mensuels, vous devez rapporter au mois un taux d'intérêt annuel (généralement en le divisant par 12) et exprimer les *périodes* en mois en multipliant par 12 un nombre d'années.

Exemples

Vous projetez de déposer 500 000 francs par an sur un compte épargne retraite pendant les 20 années à venir. Le compte est rémunéré en fin d'année (le dernier jour de chaque année), à un taux annuel de 7,5 %. Pour calculer la valeur de votre compte dans 20 ans, si vous effectuez vos versement en début de période (le premier jour de chaque année), entrez la formule suivante :

@VALFUT2(500 000;0,75;20) = 23 276 266 francs qui correspond à la valeur de votre compte après 20 ans.

Vous pouvez obtenir le même résultat avec la fonction **@VALFUTBIS**(500 000;0,075;20;1).

Fonctions @ similaires

@VALFUT calcule la valeur capitalisée d'un investissement, sur la base d'une série de versements égaux, placés à un *taux d'intérêt* périodique, sur un nombre de *périodes* de versement donné, selon une annuité ordinaire. **@VALFUTBIS** calcule la valeur capitalisée d'un investissement dont la valeur actuelle est spécifiée, selon une annuité ordinaire ou due.

@VALACT et **@VALACTBISL** déterminent la valeur actuelle d'un investissement. **@VAN** calcule la valeur actuelle nette d'un investissement, par rapport à sa valeur capitalisée.

@REMBOUR2

@REMBOUR2(*principal;taux;périodes*) calcule le montant du remboursement d'un emprunt (*principal*) à un *taux* d'intérêt donné pour un nombre de périodes de versement défini (*périodes*), selon une convention d'annuité due.

Arguments

principal représente la valeur de l'emprunt et peut être une valeur quelconque.

taux représente le taux d'intérêt de l'emprunt et peut être une valeur décimale ou un pourcentage supérieur à -1.

périodes représente le nombre de périodes de versement. *périodes* peut être n'importe quel nombre entier positif. Les nombres non entiers sont tronqués.

Remarques

La période utilisée pour *taux* doit être la même que celle utilisée pour *périodes*. Par exemple, si les versements sont mensuels, entrez le taux d'intérêt et la période par incréments mensuels. Généralement, vous devez diviser le taux d'intérêt par 12 et multiplier le nombre d'années dans *périodes* par 12.

Contrairement à **@REMBOUR**, **@REMBOUR2** utilise pour le calcul, la fin de la période plutôt que le début.

Exemples

Vous envisagez d'emprunter 8 000 francs sur 3 ans, à un taux d'intérêt annuel de 10,5 % sur la base de remboursements mensuels. Les remboursements interviennent le dernier jour du mois. Vous souhaitez connaître le montant de vos remboursements mensuels :

@REMBOUR2(8000;0,105/12;36) = 257,76 francs

@REMBOURBIS(8000;0,105/12;36;1) renvoie le même résultat.

Fonctions @ similaires

@REMBOUR et **@REMBOURBIS** calculent le remboursement d'un emprunt (*principal*) à un taux d'intérêt donné pour un nombre de périodes de versement spécifié, selon une annuité ordinaire. **@REMBOURINT** calcule les parts d'intérêts d'un versement périodique constant.

@INTVER calcule es part d'intérêts cumulées correspondant au versement périodique pour un investissement.

@PRINCVER calcule le principal correspondant au versement périodique pour un investissement. **@DUREE** calcule le nombre de périodes de versements pour un investissement. **@DUREE2** calcule le nombre de périodes nécessaires sur la base d'une série de versements égaux pour accumuler une valeur capitalisée à un taux d'intérêt périodique, selon une convention d'annuité due.

@PRIX2

@PRIX2(*règlement*; *échéance*; *coupon*; *rendement*; [*remboursement-nominal*]; [*base*]) calcule le prix d'une obligation à intérêt périodique d'une valeur nominale de 100 yens, selon les conventions japonaises.

Arguments

règlement est la date de règlement de l'obligation, exprimée en nombre date.

échéance est la date d'échéance de l'obligation, exprimée en nombre date. Si *échéance* est inférieur ou égal à *règlement*, @PRIX2 renvoie ERR.

coupon est le taux d'intérêt annuel de l'obligation, indiqué par une valeur décimale. *coupon* est une valeur positive quelconque ou nulle.

rendement est le rendement annuel, indiqué par une valeur décimale. *rendement* est une valeur positive quelconque.

remboursement-nominal est un argument facultatif qui indique la valeur de remboursement d'une valeur nominale de 100 yens. *remboursement-nominal* est une valeur positive quelconque ou nulle ; si vous l'omettez, 1-2-3 utilise 100.

base est un argument facultatif qui indique le type de base de comptage des jours à utiliser.

<u>base</u>	<u>Base de comptage des jours</u>
0	30/360
1	Réel/réel
2	Réel/360
3	Réel/365
4	Européen 30/360

Si vous insérez *base*, 1-2-3 utilise @PRIX pour calculer la valeur.

Exemples

Un bon est livrable le 18 juillet 1996 et vient à échéance le 20 juin 2000. Le taux du coupon est de 5,50 % et le rendement annuel de 5,61 %. La formule ci-dessous permet de déterminer le prix du bon :

@PRIX2(@DATE(96,7,18),@DATE(100,6,20),0.055,0.0561,100) = 99,65 yens

Fonctions @ similaires

@PRIX calcule le prix d'une obligation à intérêt périodique d'une valeur nominale de 100 yensl.

@INTCOUR calcule les intérêts courus des obligations à intérêt périodique. @RENDEMENT calcule le rendement d'obligations à intérêt périodique. @DURATION calcule la duration annuelle et @SENSIBILITETITRE calcule la duration annuelle modifiée des obligations à intérêt périodique.

@INTCOUR2 calcule les intérêts courus pour des obligations à intérêt périodique, selon les conventions japonaises.

@RENDEMENT2 renvoie le rendement pour des obligations à intérêt périodique, selon les conventions japonaises.

@VALACT2

@VALACT2 (*versements;taux;périodes*) calcule la valeur actuelle d'un investissement sur la base d'une série de versements égaux, placés à un taux d'intérêt périodique, sur un nombre de *périodes* donné, selon une convention d'annuité due.

Arguments

versements représente le taux de versement et peut être une valeur quelconque.

intérêt représente le taux d'intérêt de l'investissement et peut être une valeur décimale ou un pourcentage supérieur à -1.

périodes représente le nombre de périodes de versement. *périodes* peut être n'importe quel nombre entier positif. Les nombres non entiers sont tronqués.

Remarques

La période utilisée pour calculer *intérêt* doit être la même période utilisée pour *périodes*. Par exemple, si les versements sont mensuels, entrez l'intérêt et la période dans les incréments mensuels. Généralement, vous devez diviser le taux d'intérêt par 12 et multiplier le nombre des années dans *périodes* par 12.

Contrairement à **@VALACT**, **@VALACT2** utilise pour le mode de calcul, la fin de la période plutôt que le début.

@VALACT2 permet l'évaluation d'un investissement ou sa comparaison avec d'autres investissements. **@VALACT2** permet de comparer plusieurs types d'investissements. Par exemple, capital retraite versé en une seule fois ou placé sous forme de versements périodiques. Pour créer un tableau d'amortissement, vous utilisez conjointement **@VALACT2** et **@REMBOUR2**.

@VALACT2 complète **@REMBOUR**. **@VALACT2** détermine le montant maximal de l'emprunt que vous pouvez contracter, en fonction des versements auxquels vous pouvez faire face. De son côté, **@REMBOUR** détermine le montant des versements sur la base du montant total de l'emprunt que vous souhaitez contracter.

Exemples

Vous avez gagné un million de francs payable soit en 20 versements annuels de 50 000 francs à la fin de chaque année, soit en un versement immédiat de 400 000 francs plutôt que l'annuité de un million de francs. Vous souhaitez connaître l'option la plus avantageuse en francs actuels.

Si vous acceptez les versements annuels de 50 000 francs, vous faites l'hypothèse d'investir cette somme à un taux d'intérêt annuel de 8 %.

@VALACT2 (50000;0,08;20) renvoie 530 180 francs, ce qui signifie que les un million de francs perçus sur 20 ans sont plus avantageux que les 530 180 francs en francs actuels.

Vous obtenez le même résultat avec la fonction **@VALACTBIS**(50000;0,08;20;1)

Fonctions @ similaires

@VALACT calcule la valeur actuelle d'un investissement sur la base d'une série de versements égaux, placés à un taux d'intérêt périodique sur un nombre de périodes donné, selon une annuité ordinaire. **@VALACTBIS** calcule la valeur actuelle d'un investissement avec une valeur capitalisée spécifique, pour une annuité ordinaire ou due.

@VALFUT et **@VALFUTBISL** calculent la valeur capitalisée d'un investissement sur la base de versements égaux.

@VALFUT2 calcule la valeur capitalisée d'un investissement sur la base de versements égaux, placés à un taux d'intérêt périodique, sur un nombre de périodes donné, selon une convention d'annuité due. **@VAN** calcule la valeur actuelle nette d'un investissement en déduisant la valeur actuelle de la valeur capitalisée.

@REMBOUR et **@REMBOURBIS** calculent le remboursement d'un emprunt à un taux d'intérêt donné pour un nombre de périodes de versement spécifié. **@REMBOUR2** calcule le remboursement d'un emprunt à un taux d'intérêt donné pour un nombre de périodes de versement spécifié, selon une convention d'annuité due.

@DUREE2

@DUREE2(*versements*; *taux*; *valeur-capitalisée*) calcule le nombre de périodes nécessaires à une série de *versements* égaux pour qu'un investissement atteigne une *valeur-capitalisée* à un *taux* d'intérêt périodique, selon une convention d'annuité due.

Arguments

versements représente la valeur constante des versements. Vous pouvez indiquer n'importe quelle valeur à l'exception de 0.

taux représente le taux d'intérêt périodique. Vous pouvez indiquer une valeur décimale ou un pourcentage supérieur à -1.

valeur-capitalisée représente la somme que vous désirez atteindre. Vous pouvez indiquer une valeur quelconque.

Remarques

Vous pouvez utiliser une *valeur-capitalisée* négative pour calculer avec **@DUREE2** le nombre de périodes nécessaires au remboursement d'un emprunt. Ainsi, si vous devez rembourser 10 000 francs prêtés à un taux annuel de 10 % en versant des remboursements annuels de 1 174 francs, il vous faudra 16 ans, résultat fourni par **@ABS(@DUREE2(1174,6;0,1;-10000))**.

Contrairement à **@DUREE**, **@DUREE2** utilise la fin de la période, plutôt que le début pour le calcul.

Exemples

Vous déposez en fin d'année 2.000 francs sur un compte rémunéré à 7,5 % par an et vous souhaitez connaître le nombre d'années nécessaires pour que ce compte atteigne 100 000 francs. Il vous faut :

@DUREE2(2000;0,075;100000) = 20,76 années

Vous pouvez obtenir le même résultat avec la fonction **@DUREEBIS(2000;0,075;100000;1)**

Fonctions @ similaires

@DUREE calcule le nombre de périodes nécessaires à une série de *versements* égaux à un taux d'intérêt périodique pour qu'un investissement atteigne une *valeur-capitalisée*, selon une annuité ordinaire. **@DUREEBIS** calcule le nombre de périodes nécessaires à une série de *versements* égaux à un taux d'intérêt périodique pour qu'un investissement atteigne une *valeur-capitalisée*, selon une annuité ordinaire ou due.

@DUREECOMP calcule le nombre de périodes nécessaires pour qu'un investissement à versement unique atteigne une *valeur-capitalisée*.

@RENDEMENT2

@RENDEMENT2(*règlement*; *échéance*; *coupon*; *prix*; [*remboursement-nominal*]; [*base*]) renvoie le rendement d'obligations à intérêt périodique, selon les conventions japonaises.

Arguments

règlement est la date de règlement de l'obligation, exprimée en nombre date.

échéance est la date d'échéance de l'obligation, exprimée en nombre date. Si *échéance* est inférieur ou égal à *règlement*, @RENDEMENT2 renvoie ERR.

coupon est le taux d'intérêt annuel de l'obligation, exprimé en nombre décimal. *coupon* est n'importe quelle valeur positive ou nulle.

prix est le prix en yens à payer pour une obligation d'une valeur nominale de 100 yens. Ce peut être n'importe quelle valeur positive.

remboursement-nominal est un argument facultatif qui permet de spécifier la valeur de remboursement d'une obligation d'une valeur nominale de 100 yens. *remboursement-nominal* est une valeur positive quelconque ou nulle ; si vous l'omettez, 1-2-3 utilise 100.

base est un argument facultatif qui indique le type de base de comptage des jours à utiliser.

<u>base</u>	<u>Base de comptage des jours</u>
0	30/360
1	Actuel/actuel
2	Actuel/360
3	Actuel/365
4	Européen 30/360

Si vous insérez *base*, 1-2-3 utilise @RENDEMENT pour calculer la valeur.

Exemples

Un bon a pour date de règlement le 1er juillet 1993 et vient à échéance le 1er décembre 1998. Les coupons sont payés semi-annuellement et ont un taux nominal de 5,50 %. Le prix du bon est de 99,50 yens et sa valeur de remboursement est de 100 yens. La formule ci-dessous vous permet de déterminer le rendement de ce bon :

@RENDEMENT2(@DATE(93;7;1),@DATE(98;12;1),0.055,99.5,100) = 0,056072

Fonctions @ similaires

@RENDEMENT renvoie le rendement d'obligations à intérêt périodique.

@INTCOUR2 calcule les intérêts courus pour des obligations à intérêt périodique, selon les conventions japonaises.

@PRIX2 calcule le prix d'obligations à intérêt périodique d'une valeur nominale de 100 yens, selon les conventions japonaises.

@INTCOUR calcule les intérêts courus pour des obligations à intérêt périodique. @PRIX calcule le prix d'obligations à intérêt périodique d'une valeur nominale de 100 francs. @DURATION calcule la durée annuelle des obligations à intérêt périodique. @SENSIBILITETITRE calcule la durée annuelle modifiée des obligations à intérêt périodique.

@LIAISONOLE

@LIAISONOLE(*nom-application*;*nom-sujet*;*nom-élément*;*[format]*;*[lignes-max]*;*[col-max]*;*[feuilles-max]*) crée une liaison OLE vers des données.

La liaison peut être modifiée par modification des arguments **@LIAISONOLE**.

Arguments

nom-application est un texte indiquant le nom d'une application Windows qui prend en charge la liaison serveur OLE. Bien que la version 97 de 1-2-3 n'utilise pas cet argument, il est fourni pour une compatibilité avec les versions précédentes de 1-2-3.

nom-sujet est un texte indiquant le nom du fichier d'application vers lequel établir la liaison.

nom-élément est un texte indiquant le nom de l'élément de l'application serveur vers lequel établir la liaison. Il s'agit de l'élément du fichier d'application serveur dont vous voulez transférer les données au moyen de la liaison.

format est un texte indiquant l'un des formats du Presse-papiers : Texte, WK1 ou WK3.

Si vous omettez l'argument *format*, 1-2-3 utilise le format du Presse-papiers Texte.

lignes-max, *col-max* et *feuilles-max* indiquent le nombre maximal de lignes, de colonnes et de feuilles du champ cible. Si vous n'incluez pas l'un de ces arguments, 1-2-3 utilise le nombre de lignes, colonnes ou feuilles requis par le champ cible.

Remarques

Lorsque vous créez une liaison en utilisant la séquence de commandes Edition - Collage spécial et en choisissant "Texte", WK1 ou WK3 comme type de format, **@LIAISONOLE** apparaît dans la première cellule du champ cible de la liaison.

Les liaisons créées sont automatiquement nommées "LiaisonOLE 1", "LiaisonOLE2" et ainsi de suite.

Pour en savoir plus, reportez-vous à la rubrique [Création d'une liaison OLE dans 1-2-3](#).

Exemples

La formule **@LIAISONOLE** ci-dessous crée une liaison OLE au fichier Word Pro PRET.LWP.

@LIAISONOLE("WordPro";"C:\LOTUS\TRAVAIL\WORDPRO\PRET.LWP";"!Liaison_Signet1")

@PRODUIT

@PRODUIT(*liste*) multiplie les valeurs contenues dans *liste*.

Arguments

liste peut contenir une combinaison quelconque des éléments suivants : nombres, formules numériques et adresses ou noms de champs contenant des nombres ou formules numériques. Ces arguments doivent être séparés par les séparateurs d'arguments.

Reportez-vous également à la rubrique Arguments de fonctions @ statistiques.

Exemple

@PRODUIT(2;4;6;8) = 384

Fonctions @ similaires

@FACT calcule la factorielle d'une valeur. @SOMME additionne les valeurs d'une liste. @SOMMEPROD calcule la somme des produits des valeurs correspondantes dans plusieurs champs.

Fonctions @ de A à Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

@@(*emplacement*)

Permet d'obtenir indirectement le contenu de la cellule spécifiée dans *emplacement*.

@ABS(*x*)

Calcule la valeur absolue (positive) d'*x*.

@ACOSEC(*x*)

Calcule la réciproque de l'arc cosécante en utilisant la cosécante *x* d'un angle.

@ACOS(*x*)

Calcule la réciproque de l'arc cosinus à l'aide du cosinus *x* d'un angle.

@ACOSH(*x*)

Calcule la réciproque de l'arc cosinus hyperbolique à l'aide du cosinus hyperbolique *x* d'un angle.

@ACOTAN(*x*)

Calcule la réciproque de l'arc cotangente à l'aide de la cotangente *x* d'un angle.

@ACOTANH(*x*)

Calcule la réciproque de l'arc cotangente hyperbolique en utilisant la cotangente hyperbolique *x* d'un angle.

@ACSECH(*x*)

Calcule la réciproque de l'arc cosécante hyperbolique en utilisant la cosécante hyperbolique *x* d'un angle.

@ACTUEL

Calcule le nombre date (part du nombre entier) et nombre temps (part décimale) qui correspond à la date du jour et l'heure actuelle (fournie par l'horloge système).

@ALEAT

Génère une valeur aléatoire des nombres compris entre 0 et 15. Chaque fois qu'1-2-3 recalcule votre travail, la fonction @ALEAT génère une nouvelle valeur aléatoire.

@ALIGNCHAINE(*texte;longueur;[alignement]*)

Renvoie un libellé qui a la *longueur* de caractère nécessaire.

@AMDEGVAR(*coût;valeur récupération;durée;période début, période fin;[coefficient d'amortissement]; [commute]*)

Calcule la provision de l'amortissement d'une immobilisation selon la méthode de l'amortissement dégressif à taux variable.

@AMORANN(*coût; valeur récupération; durée; période*)

Calcule le montant de la dotation à l'amortissement des nombres d'années d'une immobilisation pour une *période* spécifiée.

@AMORDEG(*coût; valeur récupération; durée; période*)

Calcule l'amortissement d'une immobilisation selon la méthode de l'amortissement dégressif à taux double.

@AMORDEGTC(*coût; valeur récupération; durée; période*)

Calcule l'amortissement d'une immobilisation selon la méthode de l'amortissement dégressif.

@AMORLIN(*coût; valeur récupération; durée*)

Calcule la provision sur l'amortissement linéaire d'une immobilisation avec un *coût* initial, une durée normale d'utilisation et une valeur de récupération finale sur une période donnée.

@AMORT(*estimation; champ*)

Calcule le taux de rentabilité interne d'une série de rentrées de fonds générée par un investissement.

@AMORTMOD(*champ; taux-payé; taux-perçu; [type]*)

Calcule le taux de rentabilité interne modifié d'une série de rentrées de fonds générée par un investissement.

@ANNEE(*nombre-date; type*)

Extrait l'année d'un nombre entier compris entre 1 (1er janvier 1900) et 2958465 (31 décembre 9999), de *nombre-date*.

@ARRANG(*n; p*)

Calcule le nombre de séquences ordonnées (permutations) d'objets *p* qui peuvent être sélectionnés à partir d'un total d'objets *n*.

@ARRDEF(*x; [n]; [direction]*)

Arrondit la valeur *x* au multiple inférieur le plus proche de la puissance de 10 spécifiée par *n*.

@ARREXC(*x; [n]; [direction]*)

Arrondit la valeur *x* par excès au multiple le plus proche de la puissance de 10 spécifiée par *n*.

@ARRMUL(*x; multiple; [direction]*)

Arrondit la valeur *x* au *multiple* le plus proche.

@ARRONDI(*x; n*)

Arrondit la valeur *x* au multiple le plus proche de la puissance de 10 spécifiée par *n*.

@ASEC(*x*)

Calcule la réciproque de l'arc sécante en utilisant la sécante *x* d'un angle.

@ASECH(*x*)

Calcule la réciproque de l'arc sécante hyperbolique en utilisant la sécante hyperbolique *x* d'un angle.

@ASIN(*x*)

Calcule la réciproque de l'arc sinus en utilisant le sinus *x* d'un angle.

@ASINH(*x*)

Calcule la réciproque de l'arc sinus hyperbolique en utilisant le sinus hyperbolique *x* d'un angle.

@ATAN(*x*)

Calcule la réciproque de l'arc tangente en utilisant la tangente *x* d'un angle. Le résultat de la fonction @ATAN est un angle en radians de $-\pi/2$ à $\pi/2$.

@ATAN2(*x; y*)

Calcule l'arc tangente en utilisant la tangente *y/x* d'un angle. Le résultat de la fonction @ATAN2 est un angle en radians de $-\pi$ à π , en fonction des signes *x* et *y*.

@ATANH(*x*)

Calcule la réciproque de l'arc tangente hyperbolique en utilisant la tangente hyperbolique *x*.

B

@BCOMPTE(*données; rubrique; [critère]*)

Compte les cellules non vides dans une *rubrique* d'une table de base de données correspondant aux *critères* spécifiés.

@BCOMPTENUM(*données; rubrique; [critère]*)

Compte les cellules contenant des valeurs dans la *rubrique* d'une table de base de données qui correspond aux *critères* spécifiés.

@BESSELI(*x; n*)

Calcule la fonction Bessel modifiée en suivant l'ordre du nombre entier $I_n(x)$.

@BESSELJ(*x;n*)

Calcule la fonction Bessel en suivant l'ordre du nombre entier $I_n(x)$.

@BESSELK(*x;n*)

Calcule la fonction Bessel modifiée en suivant l'ordre du nombre entier $K_n(x)$.

@BESSELY(*x;n*)

Calcule la fonction Bessel en suivant l'ordre du nombre entier $Y_n(x)$, également connu en tant que fonction Neumann.

@BETA(*z;w*)

Calcule la fonction Bêta.

@BETAI(*a;b;x*)

Renvoie la fonction Bêta incomplète.

@BINOMIALE(*essais;succès;probabilité;[type]*)

Calcule la fonction de masse de la probabilité binomiale ou la loi binomiale cumulative.

@BLIT(*données;rubrique;[critère]*)

Extrait une valeur ou un libellé de la *rubrique* d'une table de base de données qui correspond aux *critères* spécifiés.

@BMAX(*données;rubrique;[critère]*)

Recherche la valeur la plus élevée dans la *rubrique* d'une table de base de données qui correspond aux *critères* spécifiés.

@BMIN(*données;rubrique;[critère]*)

Recherche la valeur la plus petite dans la *rubrique* d'une table de base de données qui correspond aux *critères* spécifiés.

@BMOYENNE(*données;rubrique;[critères]*)

Calcule la moyenne des valeurs dans la *rubrique* d'une table de base de données correspondant aux *critères* spécifiés.

@BSOMME(*données;rubrique;[critère]*)

Totalise les valeurs dans la *rubrique* d'une table de base de données répondant aux *critères* spécifiés.

@BSTDE(*données;rubrique;[critère]*)

Calcule l'écart type d'échantillon des valeurs d'une *rubrique* d'une table de base de données qui correspond aux *critères* spécifiés.

@BSTD(*données;rubrique;[critère]*)

Calcule l'écart type des valeurs d'une *rubrique* (population) d'une table de base de données qui correspond aux *critères* spécifiés.

@BVAR(*données;rubrique;[critère]*)

Calcule la variance des valeurs d'une *rubrique* (population) d'une table de base de données répondant aux *critères* spécifiés.

@BVARE(*données;rubrique;[critère]*)

Calcule la variance d'échantillon des valeurs d'une *rubrique* d'une table de base de données répondant aux *critères* spécifiés.

C

@C(*champ*)

Renvoie l'entrée dans la première cellule du *champ* sous un libellé.

@CAR(*x*)

Renvoie le caractère du code du jeu de caractères LMBCS (Lotus Multibyte Character Set) qui correspond au nombre *x*.

@CARENTIER(*libellé*)

Convertit les caractères (ASCII) un octet dans un *libellé* en caractères double octets japonais correspondants.

@CELLULE(*attribut;emplacement*)

Renvoie les données relatives à la première cellule dans un *emplacement*.

@CHAINE(*x;n*)

Convertit la valeur *x* en libellé en utilisant le format spécifié par *n*.

@CHERCHE(*chaîne1;texte;chaîne2*)

Calcule la position dans *texte* à laquelle 1-2-3 cherche la première occurrence de *chaîne1*, à partir de *chaîne2*.

@CHERCHEQ(*chaîne1*; *texte*; *chaîne2*)

Calcule la position en octets dans *texte* à laquelle 1-2-3 cherche la première occurrence de *chaîne1*, à partir de la position en octets indiquée par *chaîne2*.

@CHOIX(*x*; *liste*)

Renvoie le nième élément de la *liste* dans laquelle *n* correspond à la valeur de *x*.

@CODE(*texte*)

Renvoie le code du jeu de caractères LMBCS (Lotus Multibyte Character Set) qui correspond au premier caractère dans un code *texte*.

@COL(*champ*)

Compte les colonnes d'un champ.

@COMBIN(*n*; *p*)

Calcule le coefficient du binôme pour *n* et *p*, qui correspond en nombre aux différentes manières pour sélectionner *p* à partir de *n*, sans prendre en compte l'ordre utilisé.

@COMPRIENTRE(*valeur*; *obligation1*; *obligation2*; [*inclusion*])

Effectue un test pour savoir si la *valeur* est comprise entre *obligation1* et *obligation2* et renvoie 1 (vrai) ou 0 (faux).

@COMPTENUM(*liste*)

Compte les cellule dans une *liste* de champ, en excluant les cellules contenant des libellés.

@COMPTET(*liste*)

Compte les cellules non vides dans une *liste* de champs.

@CONSEV(*x*; *champ*; *position-colonne*)

Indique le contenu de la cellule figurant dans une colonne d'un tableau de consultation vertical.

@CONSH(*x*; *champ*; *position ligne*)

Détermine le contenu d'une cellule dans une ligne spécifiée d'un tableau de consultation horizontal dans la colonne spécifiée par *x*.

@CONSMAX(*champ-liste*)

Renvoie une référence absolue à la cellule qui contient la valeur la plus élevée dans une liste de champs.

@CONSMINP(*champ-liste*)

Renvoie une référence absolue à la cellule qui contient la valeur minimale dans une liste de champs.

@CONVDATE(*date*; *type-données*; *type-sortie*)

Convertit une date en Hijri (arabe), Farsi (iranienne) ou Hébreu (israélienne) en une date grégorienne ou inversement. Uniquement pour des versions bi-directionnelles d'1-2-3.

@CONVREF(*référence*)

Convertit les lettres de colonne ou feuille A à IV en nombre de 1 à 256 et inversement.

@COORD(*feuille*; *colonne*; *ligne*; *absolu*)

Crée une référence de cellule à partir des valeurs qui correspondent à *feuille*, *colonne* et *ligne*.

@CORREL(*champ1*; *champ2*)

Calcule le coefficient de corrélation des valeurs dans le *champ1* et *champ2*.

@CORRESP(*contenu-cellule*; *champ*; [*type*])

Indique la position relative d'une cellule dans *champ* dont le contenu correspond à *contenu-cellules*.

@COS(*x*)

Calcule le cosinus de l'angle *x*. L'angle doit être mesuré en radians.

@COSEC(*x*)

Calcule la cosécante de l'angle *x*. L'angle doit être mesuré en radians.

@COSECH(*x*)

Calcule la cosécante hyperbolique de l'angle *x*. L'angle doit être mesuré en radians.

@COSH(*x*)

Calcule le cosinus hyperbolique de l'angle *x*. Le résultat de la fonction @COSH est une valeur supérieure ou égale à 1. L'angle doit être mesuré en radians.

@COTAN(*x*)

Calcule la cotangente de l'angle *x*. L'angle doit être mesuré en radians.

@COTANH(*x*)

Calcule la cotangente hyperbolique de l'angle x . L'angle doit être mesuré en radians.

@COVAR(*champ1;champ2;*[*type*])

Calcule la covariance (population ou échantillon) des valeurs dans le *champ1* et *champ2*.

@CRITBINOM(*essais;probabilité;maxprod*)

Renvoie le plus grand nombre entier pour lequel la distribution binomiale cumulative est inférieure ou égale à *alpha*.

D

@DATE(*année;mois;jour*)

Calcule le nombre date pour l'*année*, le *mois* et le *jour* indiqués.

@DATECHaine(*date*)

Convertit un nombre date en son équivalent date et l'affiche sous la forme d'un libellé en utilisant le format date international par défaut.

@DATEVAL(*texte*)

Calcule le nombre date pour la date spécifiée dans un *texte*.

@DEBUT(*texte;n*)

Renvoie les *n* premiers caractères d'un *texte*.

@DEBUTO(*texte;n*)

Renvoie les *n* premiers octets d'un *texte*.

@DECILE(*mosaïque;champ*)

Renvoie un décile donné.

@DECIMAL(*hexadécimale*)

Convertit une valeur *hexadécimale* en son équivalent en valeur décimale signée.

@DEGENRAD(*degrés*)

Convertit des *degrés* en radians.

@DEMICAL(*libellé*)

Convertit les caractères double octets japonais dans *libellé* en caractères (ASCII) un octet correspondants.

@DERFERR(*x*)

Calcule la dérivée de la fonction d'erreur.

@DERNIERSCENARIOT(*nom-fichier*)

Renvoie le nom du dernier groupe de versionS affiché dans un classeur lors de la session 1-2-3 actuelle.

@DIFDATE(*date initiale;date finale;format*)

Calcule le nombre d'années, de mois ou de jours compris entre la date initiale et la date finale.

@DISTCHI(*x;degrés-liberté;*[*type*])

Calcule la distribution de Khi deux.

@DISTF(*x;degrés-liberté1;degrés-liberté2;*[*type*])

Calcule les distributions *F*.

@DISTT(*x;degrés-liberté;*[*type*];[*sens*])

Calcule la distribution *t* de Student.

@DONNESVERSIONS

(*option;cellule;champ-version;nom;*[*Auteur*])

Renvoie le contenu d'une cellule spécifiée dans une version.

@DURATION

(*livraison;maturité;coupon;rendement;*[*fréquence*];[*base*])

Calcule la durée annuelle d'une obligation à intérêt périodique.

@DUREE(*versements;intérêt;valeur capitalisée*)

Calcule le nombre de périodes requises pour une série de *versements* égaux afin d'accumuler une *valeur capitalisée* à un taux d'*intérêt* périodique, prenant en compte une annuité ordinaire.

@DUREE2(*versements;intérêt;valeur capitalisée*)

Calcule le nombre de périodes requises pour une série de *versements* égaux afin d'accumuler une *valeur capitalisée* à un taux d'*intérêt* périodique, prenant en compte une convention d'une annuité due.

@DUREEBIS(*versements;intérêt;valeur capitalisée;*[*type*];[*valeur actualisée*])

Calcule le nombre de périodes requises pour une série de *versements* égaux afin d'accumuler une *valeur*

capitalisée à un taux d'intérêt périodique.

@DUREECOMP(*intérêt; valeur capitalisée; valeur actualisée*)

Calcule le nombre de périodes nécessaires pour qu'un investissement (*valeur actualisée*) atteigne une *valeur capitalisée* à taux d'intérêt fixe.

E

@ECARTMED(*liste*)

Calcule la moyenne des écarts type absolus des valeurs dans une *liste*.

@EGAL(*texte1; texte2*)

Renvoie 1 (vrai) si *texte1* et *texte2* sont identiques ; 0 (faux) dans le cas contraire.

@ENOMBRE(*nombre-chaîne*)

Convertit *nombre-chaîne* des caractères numériques Thai en chaîne numérique arabe.

@ENT(*x*)

Renvoie une part du nombre entier d'*x*.

@EPURE(*texte*)

Retire les caractères non imprimables d'un *texte*.

@ERR

Renvoie la valeur ERR.

@ERRMOY(*champ*)

Calcule l'erreur type de la méthode échantillon pour les valeurs dans un *champ*.

@ESTCHAINE(*x*)

Effectue un test pour savoir si *x* correspond à du texte ou à une cellule qui contient un libellé ou une formule dont le résultat est un libellé et renvoie 1 (vrai) ou 0 (faux).

@ESTCHAMP(*champ*)

Effectue un test pour savoir si *champ* correspond à un nom de champ défini ou une adresse de champ valide et renvoie 1 (vrai) ou 0 (faux).

@ESTERR(*x*)

Effectue un test pour savoir si *x* correspond à la valeur ERR et renvoie 1 (vrai) ou 0 (faux).

@ESTFICHER(*nom-fichier; [type]*)

Effectue un test pour savoir si *nom-fichier* correspond à un fichier sur disque ou en mémoire et renvoie 1 (vrai) ou 0 (faux).

@ESTMACRO(*nom*)

Effectue un test pour savoir si *nom* correspond à une sous-routine LotusScript complémentaire définie et renvoie 1 (vrai) ou 0 (faux).

@ESTND(*x*)

Effectue un test pour savoir si *x* correspond à la valeur ND et renvoie 1 (vrai) ou 0 (faux).

@ESTNUM(*x*)

Effectue un test pour savoir si *x* correspond à une valeur, ND, ERR ou une cellule vierge et renvoie 1 (vrai) ou 0 (faux).

@ESTVIDE(*emplacement*)

Effectue un test pour savoir si *emplacement* correspond à une cellule vierge et renvoie 1 (vrai) ou 0 (faux).

@EXP(*x*)

Calcule la valeur de la constante e (approximativement 2,718282) atteignant la puissance *x*.

@EXP2(*x*)

Calcule la valeur de la constante e (approximativement 2,718282) atteignant la puissance ($-x^2$).

F

@FACT(*n*)

Calcule la factorielle de *n*.

@FAUX

Renvoie la valeur logique 0 (faux).

@FERR(*limite inf; [limite sup]*)

Calcule la fonction d'erreur comprise entre *limite inf* et *limite sup*.

@FERRC(*x*)

Calcule la fonction d'erreur complémentaire comprise entre x et ∞ (l'infini).

@FEUILLES(*champ*)

Compte les feuilles dans un *champ*.

@FIN(*texte;n*)

Renvoie les derniers n caractères dans un *texte*.

@FINQ(*texte;n*)

Renvoie les derniers n octets dans un *texte*.

G

@GAMMA(x)

Calcule la fonction gamma.

@GAMMAI($a;x$;*[complément]*)

Calcule la fonction gamma incomplète.

@GRAND(*champ;n*)

Indique la nième plus grande valeur dans un *champ*.

H

@HEURE(*nombre-temps*)

Détermine l'heure, valeur comprise entre 0 (minuit) et 23 (23:00) de *nombre-temps*.

@HEUREVAL(*texte*)

Calcule le nombre temps correspondant à l'heure spécifiée dans *texte*.

@HEX(x)

Convertit un nombre décimal en valeur hexadécimale.

I

@IMPAIR(x)

Arrondit la valeur x éloignée de 0 au nombre entier impair le plus proche.

@INDEX(*champ;colonne;ligne*;*[feuille]*)

Renvoie le contenu d'une cellule dans *colonne*, *ligne* et *feuille* d'un *champ*.

@INFO(*attribut*)

Renvoie les données relatives à la session 1-2-3 actuelle.

@INFODATE(*date;attribut*)

Renvoie les données relatives à la *date*.

@INFOSCENARIO(*option;nom*;*[auteur]*)

Renvoie les données relatives au groupe de versions.

@INFOVERSION(*option;champ-version;nom*;*[auteur]*)

Renvoie les données relatives à une version.

@INTCOUR(*règlement;émission;premier-intérêt;coupon*;*[valeur nominale]*;*[fréquence]*;*[base]*)

Calcule l'intérêt couru des obligations avec des règlements d'intérêt périodique.

@INTCOUR2(*règlement;maturité;coupon*;*[valeur nominale]*;*[fréquence]*;*[émission]*;*[premier-intérêt]*;*[type]*)

Calcule l'intérêt couru des obligations avec des règlements d'intérêt périodique, selon les conventions japonaises.

@INTVER

(*principal;taux;périodes;période-début*;*[période-fin]*;*[type]*;*[valeur capitalisée]*)

Calcule la part d'intérêt cumulée correspondant à un versement périodique au titre d'un crédit pour un nombre de périodes de versements spécifiés (*taux*).

@ISAAF(*nom*)

Effectue un test pour savoir si *nom* correspond à une fonction LotusScript complémentaire définie et renvoie 1 (vrai) ou 0 (faux).

@ISAPP(*nom*)

Effectue un test pour savoir si *nom* correspond à une application complémentaire actuellement chargée et renvoie 1 (vrai) ou 0 (faux).

J

@J360(*date initiale;date finale*)

Calcule le nombre de jours compris entre les deux nombres dates, sur la base d'une année comportant 360 jours.

@JOUR(*nombre-date*)

Extrait le jour du mois, une valeur de 1 à 31 du *nombre date*.

@JOURACTUEL

Calcule le nombre date correspondant à la date du jour (fournie par votre ordinateur).

@JOUROUV(*date initiale;jours;[champ-congés];[weekends]*)

Calcule le nombre date pour la date qui correspond au nombre de *jours* spécifié avant ou après *date-initiale*.

@JOURS(*date initiale;date finale;[base]*)

Calcule le nombre de jours compris entre la *date initiale* et la *date finale* sur une base cde comptage des jours.

@JOURS360(*date initiale;date finale*)

Calcule le nombre de jours compris entre la *date initiale* et la *date finale*, sur la base d'une année comportant 360 jours, selon les normes du secteur des obligations aux Etats-Unis.

@JOURSEM(*nombre-date*)

Extrait le jour de la semaine d'un nombre entier compris entre 0 (lundi) et 6 (dimanche) de *nombre date*.

@JOURSOUVRES(*date début;date fin;[champ-congés];[weekends]*)

Calcule le nombre de jours compris entre *date début* et *date fin*, en excluant les weekends et les jours de congés.

K

@KURTOSIS(*champ;[type]*)

Calcule le kurtosis des valeurs dans un *champ*.

L

@LIAISONOLE(*nom-app;nom-rubrique;nom-élément;[format];[lignes-max];[colonnes-max];[feuilles-max]*)

Crée une liaison OLE à des données.

@LIGNES(*champ*)

Compte les lignes dans un *champ*.

@LN(*x*)

Calcule le logarithme népérien (base e) de la valeur *x*.

@LNFACT(*n*)

Calcule le logarithme népérien de la factorielle *n*.

@LNGAMMA(*x*)

Calcule le logarithme népérien de la fonction gamma.

@LOG(*x*)

Calcule le logarithme décimal (base 10) de la valeur *x*.

@LONG(*texte*)

Compte les caractères dans un *texte*.

@LONGQ(*texte*)

Compte le nombre d'octets dans un *texte*.

M

@MAJUSC(*texte*)

Met en majuscules toutes les lettres d'un *texte*.

@MAX(*liste*)

Détermine la valeur maximale dans *liste*.

@MAXNUM(*liste*)

Détermine la valeur maximale dans *liste*, en excluant les cellules contenant des libellés.

@MEDIANE(*liste*)

Renvoie la valeur médiane dans *liste*.

@MEDIANENUM(*liste*)

Renvoie la valeur médiane dans *liste*, en ignorant les cellules vierges, les libellés et les formules générant des libellés.

@MILIEU(*texte;nombre-début;n*)

Extrait *n* caractères d'un *texte*, à partir du caractère spécifié dans *nombre-début*.

@MLIEUO(*texte;snombre-début;n*)

Extrait *n* octets d'un *texte*, à partir des données en octets dans *nombre-début*.

@MIN(*liste*)

Détermine la valeur minimale dans *liste*.

@MINNUM(*liste*)

Détermine la valeur minimale dans *liste*, en excluant les cellules contenant des libellés.

@MINUSC(*texte*)

Met en minuscules toutes les lettres d'un *texte*.

@MINUTE(*nombre-temps*)

Détermine les minutes, une valeur comprise entre 0 et 59 de *nombre-temps*.

@MOD(*x;y*)

Calcule le reste (modulo) de *x/y*. Le signe du résultat correspond au signe *x*.

@MODULO(*x;y*)

Calcule le reste (modulo) de *x/y*. Le signe du résultat correspond au signe *y*.

@MOIS(*nombre-date*)

Extrait le mois, une valeur comprise entre 1 et 12 du nombre date.

@MOISDECALER(*date début;mois;[jour du mois];[base]*)

Calcule le nombre date pour la date qui correspond à un nombre spécifié de *mois* avant ou après *date début*.

@MOYENNE(*liste*)

Calcule la moyenne des valeurs d'une *liste*.

@MOYENNENUM(*liste*)

Calcule la moyenne des valeurs dans *liste*, en excluant toutes les cellules contenant des libellés.

@MOYGEOM(*liste*)

Calcule la moyenne géométrique des valeurs dans *liste*.

@MOYHARM(*liste*)

Calcule la moyenne harmonique des valeurs dans *liste*.

@MOYPOND(*champ-données;champ-pondération;[type]*)

Calcule la moyenne pondérée des valeurs dans *champ-données*.

N

@N(*champ*)

Renvoie l'entrée dans la première cellule d'un *champ* en tant que valeur. Si la cellule contient un libellé, la fonction @N renvoie la valeur 0.

@NBRECHAINE(*nombre;type*)

Convertit *nombre* en texte japonais correspondant au nombre, en utilisant le format spécifié dans *type*.

@ND

Renvoie la valeur ND (non disponible).

@NOMCHAMP(*cellule*)

Renvoie le nom du champ dans lequel une *cellule* est placée.

@NOMPROPRE(*texte*)

Met en majuscules la première lettre de tous les mots de *texte* et en minuscules les autres lettres.

@NORMALE(*x;[moyen];[std];[type]*)

Calcule la fonction de distribution normale pour *x*.

@NSOMME(*position;n;liste*)

Ajoute toutes les nièmes valeurs dans *liste*, débutant par *position*.

O

Aucune entrée

P

@PAIR(*x*)

Arrondit la valeur *x* éloignée de 0 au nombre entier pair le plus proche.

@PETIT(*champ;n*)

Détermine la *nième* valeur minimale dans un *champ*.

@PI

Renvoie la valeur π (pour π égal à 3,14159265358979).

@POINTCELL(*attribut*)

Renvoie des données relatives à la cellule actuelle.

@POISSON(*x;moyenne:[cumul]*)

Calcule la distribution de Poisson.

@POURCENTILE(*x;champ*)

Calcule le *xième* pourcentile d'échantillon des valeurs d'un *champ*.

@PRANG(*x;champ:[ndéc]*)

Détermine le pourcentile d'*x* sur les valeurs dans un *champ*.

@PREVISION(*champ-x;y;champ-x*)

Renvoie une valeur prévisionnelle pour *x*, sur la base des tendances linéaires des valeurs entre *champ-y* et *champ-x*.

@PRINCVER(*principal;intérêt;taux;période début:[période fin];[type];[valeur capitalisée]*)

Calcule le principal correspondant au versement périodique sur un emprunt (*principal*) à un taux d'*intérêt* donné pour un nombre spécifié de périodes de versement (*période*).

@PRIX(*règlement;échéance;coupon;prix;[remboursement nominal];[fréquence];[base]*)

Calcule le prix d'une obligation à intérêt périodique de 100 francs de nominal.

@PRIX2(*règlement;échéance;coupon;prix;[remboursement];[base]*)

Calcule le prix d'une obligation à intérêt périodique de 100 yens de nominal selon les conventions japonaises.

@PRODUIT(*liste*)

Multiplie les valeurs dans *liste*.

Q

@QUARTILE(*titre;champ*)

Renvoie un quartile donné.

@QUOTIENT(*x;y*)

Divise *x* par *y* et réduit le quotient à un nombre entier.

R

@RACINE(*x*)

Renvoie la racine carrée positive de *x*.

@RACINEPI(*x*)

Calcule la racine carrée de $x^* \pi$.

@RADENDEG(*radians*)

Convertit des *radians* en degrés.

@RANG(*nombre;champ:[ordre]*)

Calcule la position relative d'une valeur dans un *champ* en relation avec d'autres valeurs du champ.

@REGRESSION(*champ-x;champ-y;attribut:[origine]*)

Effectue une analyse de régression linéaire multiple et renvoie les statistiques spécifiées.

@REMBOUR(*principal;intérêt;période*)

Calcule le versement d'un emprunt (*principal*) à un taux d'*intérêt* donné pour un nombre spécifié de périodes de versement (*période*), en prenant en compte une annuité ordinaire.

@REMBOUR2(*principal;intérêt;période*)

Calcule le versement d'un emprunt (*principal*) à un taux d'*intérêt* donné pour un nombre spécifié de périodes de versement (*période*), en prenant en compte une convention d'annuité due.

@REMBOURBIS(*principal;intérêt;période:[type];[valeur capitalisée]*)

Calcule le versement d'un emprunt (*principal*) à un taux d'*intérêt* donné pour un nombre spécifié de périodes de versement (*période*).

@REMBOURC(*principal;intérêt;période*)

Calcule le versement d'un emprunt (*principal*) à un taux d'*intérêt* donné pour un nombre spécifié de périodes de

versement (*période*). Reconnaît les conventions d'emprunt canadiennes.

@REBOURINTI(*principal;intérêt;taux;période*)

Calcule la part d'*intérêt* d'un versement périodique constant.

@RENDEMENT(*règlement;échéance;coupon;prix;[remboursement nominal];[fréquence];[base]*)

Renvoie le taux de rendement pour des obligations à intérêt périodique.

@RENDEMENT2(*règlement;échéance;coupon;prix;[remboursement nominal];[base]*)

Renvoie le taux de rendement pour des obligations à intérêt périodique selon les conventions japonaises.

@REPETE(*texte;n*)

Répète du *texte* autant de fois qu'il est indiqué par *n*.

S

@SEC(*x*)

Calcule la sécante de l'angle *x*.

@SECH(*x*)

Calcule la sécante hyperbolique de l'angle *x*.

@SECONDE(*nombre-temps*)

Extrait le seconde d'un nombre entier compris entre 0 et 59 du *nombre-temps*.

@SENSIBILITETITRE

(*règlement;maturité;coupon;sens;[fréquence];[base]*)

Calcule la durée annuelle modifiée d'une obligation à intérêt périodique.

@SI(*condition;x;y*)

Effectue une *condition* et renvoie *x* si la *condition* est remplie ou *y* dans le cas contraire.

@SIGNE(*x*)

Renvoie 1 si *x* est une valeur positive, 0 si *x* est une valeur nulle et -1 si *x* est une valeur négative.

@SIN(*x*)

Calcule le sinus de l'angle *x*.

@SINH(*x*)

Calcule le sinus hyperbolique de l'angle *x*.

@SINTI(*principal;intérêt;périodes;période*)

Calcule la part d'*intérêt* d'un versement périodique dans lequel la part principale est la même pour chaque période.

@SOMECART(*liste*)

Calcule la somme des carrés des écarts pour les valeurs d'une *liste* à partir de leur moyenne.

@SOMME(*liste*)

Ajoute des valeurs dans *liste*.

@SOMMECAR(*liste*)

Calcule la somme des carrés des valeurs dans *liste*.

@SOMMENEGATIFS(*liste*)

Additionne uniquement les valeurs négatives dans *liste*.

@SOMMEPOSITIFS(*liste*)

Additionne uniquement les valeurs positives dans *liste*.

@SOMMEPROD(*liste*)

Multiplie les valeurs des cellules correspondantes dans une *liste* de champs, puis additionne les produits.

@SOMMESERIE(*x;n;m;coefficients*)

Calcule la somme d'une série de puissances.

@SOMMEXMY2(*champ1;champ2*)

Soustrait les valeurs du *champ2* des cellules correspondantes dans le *champ1*, calcule le carré des différences, puis additionne les résultats.

@STD(*liste*)

Calcule l'écart type de population des valeurs dans *liste*.

@STDE(*liste*)

Calcule l'écart type échantillon des valeurs dans *liste*.

@STDENUM(*liste*)

Calcule l'écart type échantillon des valeurs dans une *liste*, en excluant les cellules contenant des libellés.

@STDNUM(*liste*)

Calcule l'écart type de population des valeurs dans *liste*, en excluant les cellules contenant des libellés.

@SUBSTITUE(*texte-origine, nombre-début; n; nouveau-texte*)

Substitue *n* caractères dans *texte-origine* à *nouveau-texte*, commençant par *nombre-début*.

@SUBSTITUEQ(*texte-origine, nombre-début; n; texte-nouveau*)

Substitue *n* octets dans *texte-origine* à *nouveau-texte*, commençant par *nombre-début*.

@SUPPRESPEACE(*texte*)

Supprime d'un *texte* les espaces de début, de fin et en double.

@SOUSTOTAL(*liste*)

Additionne les valeurs de *liste*. Utilisez cette fonction pour indiquer à la fonction @TOTALGEN les cellules à additionner.

@SYMDIST(*champ; [type]*)

Calcule l'asymétrie des valeurs dans un *champ*.

T

@TAN(*x*)

Calcule la tangente de l'angle *x*.

@TANH(*x*)

Calcule la tangente hyperbolique de l'angle *x*.

@TAUX(*valeur capitalisée; valeur actualisée; taux*)

Calcule le taux d'intérêt périodique nécessaire pour qu'un investissement atteigne une *valeur capitalisée* sur des périodes données.

@TAUXBIS(*période; versement; valeur actualisée; [type]; [valeur capitalisée]; [estimation]*)

Calcule le taux d'intérêt périodique nécessaire pour qu'une annuité (*valeur actualisée*) atteigne une *valeur capitalisée* sur le nombre de périodes de composition en *taux*.

@TCHERCHE(*texte-recherche; texte; colonne-début*)

Calcule la position logique des caractères thaï dans un *texte* dans lequel 1-2-3 détermine la première occurrence de *texte-recherche*, commençant par le caractère thaï indiqué par *colonne-début*. Utilisé avec les chaînes de caractères thaï.

@TDATECHAINE(*valeur-date*)

Convertit *valeur-date* en une chaîne de date thaï dans un format réduit.

@TDATENOM(*valeur-date*)

Convertit *valeur-date* en jour de la semaine correspondant en thaï.

@TDEBUT(*texte; n*)

Renvoie les premiers *n* caractères logiques thaï dans un *texte*. Utilisé avec les chaînes de caractères thaï.

@TEMPS(*heure; minutes; secondes*)

Calcule le nombre temps correspondant à l'*heure*, aux *minutes* et aux *secondes spécifiées*.

@TESTCHI(*champ1; [champ2]*)

Effectue un test de Khi deux sur l'autonomie des données du *champ1* ou un test de Khi deux sur le bon emplacement des données dans le *champ1* et *champ2*.

@TESTF(*champ1; champ2*)

Effectue le test F et renvoie la probabilité associée.

@TESTT(*champ1; champ2; [type]; [sens]*)

Effectue le test *t* de Student sur les données du *champ1* et du *champ2* et renvoie la probabilité associée.

@TESTZ(*champ1; moyenne1; écart1; [sens]; [champ2]; [moyenne2]; [écart2]*)

Exécute un test *z* sur une ou deux populations et renvoie la probabilité associée.

@TFIN(*texte; n*)

Renvoie les derniers *n* caractères logiques thaï dans un *texte*. Utilisé avec les chaînes de caractères thaï.

@TLDATECHAINE(*valeur-date*)

Convertit *valeur-date* en une chaîne de date thaï dans un format long.

@TLONG(*texte*)

Compte le nombre de caractères logiques thaï dans un *texte*. Utilisé avec les chaînes de caractères thaï.

@TMILIEU(*texte;nombre-début;n*)

Extrait des *n* caractères logiques du *texte*, commençant par le caractère thaï dans *nombre-début*. Utilisé avec les chaînes de caractères thaï.

@TNBRECHaine(*nombre*)

Convertit *nombre* en une chaîne de nombres thaï indiquée.

@TNOMBRE(*chaîne-nombre*)

Convertit *chaîne-nombre* utilisant des nombres arabes en sa chaîne équivalente de caractères numériques thaï.

@TOTALGEN(*liste*)

Calcule la somme de toutes les cellules dans *liste* dont les formules contiennent la fonction.

@TRONQUE(*x;[n]*)

Tronque *x* au nombre de positions de décimales spécifiées par *n*.

@TSUBSTITUE(*texte-original;nombre-début;n;texte-nouveau*)

Remplace des *n* caractères logiques thaï dans *texte-original* avec *texte-nouveau*, commençant par *nombre-début*. Utilisé avec les chaînes de caractères thaï.

U

Aucune entrée

V

@VALACT(*versements;taux;périodes*)

Calcule la valeur actuelle d'un investissement, sur la base d'une série de *versements* égaux, escompté à un *taux* d'intérêt périodique sur la période indiquée par *périodes*, prenant en compte une annuité ordinaire.

@VALACT2(*versements;taux;périodes*)

Calcule la valeur actuelle d'un investissement, sur la base d'une série de *versements* égaux, escompté à un *taux* d'intérêt périodique sur le nombre de périodes indiqué par *périodes*, prenant en compte une convention d'une annuité due.

@VALACTBIS(*versements;taux;périodes;[type];[valeur actualisée]*)

Calcule la valeur actuelle d'un investissement avec une *valeur capitalisée* spécifiée, pour une annuité ordinaire ou une annuité due.

@VALACTMONTANT(*valeur actualisée;taux;périodes;[fréquence]*)

Renvoie la valeur actualisée d'une somme forfaitaire à recevoir sur un nombre donné de périodes dans le futur (*périodes*) et escompté à un *taux* d'intérêt donné.

@VALEUR(*texte*)

Convertit un nombre entré sous *texte* en sa valeur correspondante.

@VALFUT(*versements;taux;périodes*)

Calcule la valeur capitalisée d'un investissement sur la base d'une série de *versements* égaux, atteignant un *taux* d'intérêt périodique avec le nombre de *périodes* de versements, prenant en compte une annuité ordinaire.

@VALFUT2(*versements;taux;périodes*)

Calcule la valeur capitalisée d'un investissement sur la base d'une série de *versements* égaux, atteignant un *taux* d'intérêt périodique avec le nombre de *périodes* de versements, prenant en compte une convention d'annuité due.

@VALFUTBIS(*versements;taux;périodes;[type];[valeur actualisée]*)

Calcule la valeur capitalisée d'un investissement avec une valeur actualisée spécifiée pour une annuité ordinaire ou due.

@VALFUTMONTANT(*principal;taux;périodes;[fréquence]*)

Renvoie la valeur capitalisée d'un montant forfaitaire investi à un *taux* donné pour un nombre de périodes données.

@VAN(*intérêt;champ;[type]*)

Calcule la valeur actualisée nette d'une série de mouvements de trésorerie futurs(*champ*), escompté à un *taux d'intérêt* périodique fixe.

@VAR(*liste*)

Calcule la variance de population des valeurs dans *liste*.

@VARE(*liste*)

Calcule la variance de population échantillon des valeurs dans *liste*.

@VARENUM(*liste*)

Calcule l'écart type de population échantillon des valeurs dans *liste*, en excluant les cellules contenant des libellés.

@VAR(*liste*)

Calcule la variance de population des valeurs dans *liste*.

@VARNUM(*liste*)

Calcule la variance de population des valeurs dans *liste*, en excluant les cellules contenant des libellés.

@VERSIONACTUELLE(*champ*)

Renvoie le nom de la version actuelle dans *champ*.

@VRAI

Renvoie la valeur logique 1 (vrai).

W

Aucune entrée

X

@XAMORT(*estimation;fonds;dates*)

Renvoie le taux de rentabilité interne d'une série de mouvements de trésorerie.

@XINDEX(*champ;en-tête-colonne;en-tête-ligne;[en-tête-feuille]*)

Renvoie le contenu d'une cellule placée à l'intersection spécifiée par *en-tête-colonne*, *en-tête-ligne*, et *en-tête-feuille*.

@XVAN(*taux;fonds;dates*)

Renvoie la valeur actualisée nette d'une série de mouvements de trésorerie.

Y

Aucune entrée

Z

Aucune entrée

