

Worksheet Data Types

Cells can contain two types of information - constant values and formulas.

- Constant values are numbers, including dates and times, logical values, error values, and text.
- Formulas are groups of constant values, cell references, names, functions, and operators that result in a new value when calculated or evaluated.

For more information, choose the Contents button or the Index button.

Formula Operators

When creating formulas, Formula One provides a set of operators for specifying the type of calculation or evaluation to be performed on the formula data. The following table lists the formula operators.

<u>Operator Type</u>	<u>Operator</u>	<u>Description</u>
Arithmetic	+	Addition
	-	Subtraction
	/	Division
	*	Multiplication
	%	Percentage
	^	Exponentiation
Text	&	Concatenation
Comparison	=	Equal to
	>	Greater than
	<	Less than
	>=	Greater than or equal to
	<=	Less than or equal to
	<>	Not equal to
Reference	:, .., .	Range - produces a reference that includes all the cells between the two references (e.g., A1:A5 includes cells A1 and A5 and all cells in between).
	,	Union - produces one reference that includes the two references (e.g., A1:A10,C1:C10).

[Operator Precedence](#)

Operator Precedence

When combining operators in a formula, Formula One uses a specific order of precedence to calculate the formula. The following table lists the order of precedence for formula operators.

<u>Operator</u>	<u>Description</u>
()	Parentheses
:, ..., .	Range
,	Union
-	Negation (single operand)
%	Percentage
^	Exponentiation
* and /	Multiplication and Division
+ and -	Addition and Subtraction
&	Text concatenation
= < > <= >= <>	Comparison

Operators of like precedence are evaluated left to right. Parentheses should be used when it is necessary to change the order of evaluation. The following example illustrates how the result of a formula can be altered by adding parentheses to change the order of precedence.

<u>Formula</u>	<u>Result</u>
1+2*37	75
(1+2)*37	111

As illustrated in the previous table, the multiplication operator (*) has higher precedence than the addition operator (+). It is evaluated first unless parentheses are used to force the addition to take place first.

[Formula Operators](#)

Cell References

A reference identifies a cell by referring to the row and column coordinates of the cell. References are based on the row and column headings. For example, A1 refers to the cell at the intersection of row 1 and column A. References can be used in formulas to access data from a worksheet.

A range of cells is specified by placing a colon (:) between two cell references. For example, the reference A1:C3 refers to the range anchored by cells A1 and C3. The range includes all cells in columns A, B, and C of rows 1, 2, and 3.

[Absolute and Relative References](#)

[Automatically Entering Cell References](#)

Absolute and Relative References

There are two types of cell references - relative and absolute.

- Relative references point to a cell based on its relative position to the current cell. When the cell containing the reference is copied, the reference is adjusted to point to a new cell with the same relative offset as the originally referenced cell.
- Absolute references point to a cell at an exact location. When the cell containing the formula is copied, the reference does not change. Absolute references are designated by placing a dollar sign (\$) in front of the row and column that is to be absolute.

References can be part absolute and part relative. These are called mixed references. The following table lists the reference types.

<u>Reference</u>	<u>Type</u>
A1	Relative reference pointing to cell A1.
\$A\$1	Absolute reference pointing to cell A1.
\$A1	Absolute column reference, relative row reference pointing to cell A1.
A\$1	Relative column reference, absolute row reference pointing to cell A1.

The reference operators can be used to specify multiple ranges in the same reference. For example, A1:C1,A10:C10 specifies the three cells A1, B1, and C1 and the three cells A10, B10, and C10. The formula =SUM(A1:C1,A10:C10) adds the values in all six cells.

[Cell References](#)

[Automatically Entering Cell References](#)

Automatically Entering Cell References

Cell references can be automatically entered as you enter a formula.

To automatically enter a cell reference:

1. Enter the formula to the point of the range reference.
2. With the mouse, select the cell or range you want to reference.

The reference of the range you select is automatically placed in the formula.

When you enter a cell reference in this manner, Formula One assumes it is a relative reference.

[Cell References](#)

[Absolute and Relative References](#)

Built-In Worksheet Functions

Formula One contains a set of 130 built-in worksheet functions that provide the ability to perform complex calculations with very little work.

Worksheet functions:

- calculate and evaluate data.
- can be used alone or in a formula.
- are entered directly in the worksheet.

Like formulas, worksheet functions return data to the cell in which they are entered.

Each function performs a specific calculation. The [SQRT](#) function is an example of a built in function. With this function, you can easily calculate the square root of a number. The following example calculates the square root of 118:

```
=SQRT(118)
```

[Understanding Functions](#)

[Entering Functions](#)

[Nesting Functions](#)

[Entering Arguments](#)

[Syntax Errors](#)

Understanding Functions

Most worksheet functions are composed of keywords and arguments. Every worksheet function contains a keyword, but not all functions require arguments.

- The keyword identifies the function and tells the worksheet what type of calculation or evaluation is performed. Each function keyword is unique.
- Arguments provide the data for the function to calculate or evaluate. The arguments for a function immediately follow the function keyword and are enclosed in parentheses.

[Built-In Worksheet Functions](#)

[Entering Functions](#)

[Nesting Functions](#)

[Entering Arguments](#)

[Syntax Errors](#)

Entering Functions

When entering functions in a worksheet, all functions are preceded by an equal sign (=). The leading equal sign tells the worksheet that the following information is to be evaluated or calculated.

The function keyword follows the equal sign. It can be entered in lowercase or uppercase characters. After the function is entered, the worksheet records the function keyword in uppercase characters, regardless of how it was entered.

If a function requires multiple arguments, the arguments are separated by commas. Some functions contain optional arguments. If you omit an optional argument, a default value is assumed for the argument.

Functions that do not require arguments still require a set of parentheses following the function keyword.

[Built-In Worksheet Functions](#)

[Understanding Functions](#)

[Nesting Functions](#)

[Entering Arguments](#)

[Syntax Errors](#)

Nesting Functions

A function can be used as an argument for another function. When a function is used in this manner, you are nesting functions. The nested function must return the appropriate type of data for the function in which it is nested. You must also provide the necessary arguments for the nested function.

In the following example, the [AVERAGE](#) function is used as an argument for the [SUM](#) function. In this case, AVERAGE is nested in SUM.

```
=SUM(5.23, 6.82, AVERAGE(2.45, 5.62, 7.74), 8.95, 9.01)
```

[Built-In Worksheet Functions](#)

[Understanding Functions](#)

[Entering Functions](#)

[Entering Arguments](#)

[Syntax Errors](#)

Entering Arguments

The arguments for a function can be:

- Numerical values
- Logical values
- Text strings
- Error values
- References to cells or ranges

For most arguments, you can substitute a cell or range reference for the data required by an argument. For example, if an argument requires a number, you can substitute a reference to a cell that contains a number. The number in the referenced cell is used in the calculation of the function. The data in the referenced cell must be appropriate for the argument for which it is used.

[Built-In Worksheet Functions](#)

[Understanding Functions](#)

[Entering Functions](#)

[Nesting Functions](#)

[Syntax Errors](#)

Syntax Errors

If the worksheet function you enter contains syntax errors, Formula One does not allow the function to be entered. You must correct the errors before proceeding with other tasks.

[Built-In Worksheet Functions](#)

[Understanding Functions](#)

[Entering Functions](#)

[Nesting Functions](#)

[Entering Arguments](#)

ABS

Description

Returns the absolute value of a number.

Syntax

ABS (*number*)

r	Descriptio
<u><i>number</i></u>	<u><i>n</i></u> Any number.

Remarks

An absolute value does not display a positive or negative sign.

Examples

These functions both return 1:

ABS (-1)

ABS (1)

ACOS

Description

Returns the arc cosine of a number.

Syntax

ACOS (*number*)

Parameter	Description
<i>number</i>	The cosine of the angle. The cosine can range from 1 to -1.

Remarks

The resulting angle is returned in radians (from 0 to π). To convert the resulting radians to degrees, multiply the radians by $180/\pi$ ().

Examples

This function returns 1.05:

```
ACOS (.5)
```

This function returns 1.77:

```
ACOS (-.2)
```

See Also

[COS](#)

ACOSH

Description

Returns the inverse hyperbolic cosine of a number.

Syntax

ACOSH (*number*)

Parameter	Description
<u>r</u> <i>number</i>	Any number equal to or greater than 1.

Examples

This function returns .62:

```
ACOSH (1.2)
```

This function returns 1.76:

```
ACOSH (3)
```

See Also

[ASINH](#)

[ATANH](#)

[COSH](#)

ADDRESS

Description

Creates a cell address as text.

Syntax

ADDRESS (*row*, *column*, *ref_type* [, *a1*] [, *sheet*])

Parameter	Description
<i>row</i>	The row number for the cell address.
<i>column</i>	The column number for the cell address.
<i>ref_type</i>	The cell reference type. Following are the valid values for this argument. 1 Absolute 2 Absolute row, relative column 3 Relative row, absolute column 4 Relative
<i>a1</i>	The reference format. This argument must be TRUE() to represent an A1 reference format; Formula One does not support the R1C1 reference format.
<i>sheet</i>	The name of an external worksheet view control. Omitting this argument assumes that the reference exists in the current spreadsheet.

Examples

This function returns \$F\$5:

```
ADDRESS(5, 6, 1)
```

This function returns SALES!F5:

```
ADDRESS(5, 6, 4, TRUE(), "SALES.")
```

See Also

[COLUMN](#)

[OFFSET](#)

[ROW](#)

AND

Description

Returns True if all arguments are true; returns False if at least one argument is false.

Syntax

AND (*logical_list*)

Parameter	Description
<i>logical_list</i>	A list of conditions separated by commas. You can include as many as 30 conditions in the list. The list can contain logical values or a reference to a range containing logical values. Text and empty cells are ignored. If there are no logical values in the list, the error #VALUE! is returned.

Examples

This function returns True because both arguments are true:

```
AND(1+1=2, 5+5=10)
```

This function returns False:

```
AND(TRUE(), FALSE())
```

See Also

[ROW](#)

[NOT](#)

[OR](#)

ASIN

Description

Returns the arcsine of a number.

Syntax

ASIN (*number*)

Parameter	Description
<i>r</i> <i>number</i>	The sine of the resulting angle, ranging from -1 to 1.

Remarks

The resulting angle is returned in radians (ranging from $-\pi/2$ to $\pi/2$). To convert the resulting radians to degrees, multiply the radians by $180/\pi()$.

Examples

This function returns -1.57:

```
ASIN(-1)
```

This function returns .41:

```
ASIN(.4)
```

See Also

[ASINH](#)

[PI](#)

[SIN](#)

ASINH

Description

Returns the inverse hyperbolic sine of a number.

Syntax

ASINH (*number*)

r	Descriptio
<u><i>number</i></u>	<u><i>n</i></u> Any number.

Examples

This function returns 2.37:

`ASINH(5.3)`

This function returns -2.09:

`ASINH(-4)`

See Also

[ACOSH](#)

[ASIN](#)

[ATANH](#)

[SINH](#)

ATAN

Description

Returns the arctangent of a number.

Syntax

ATAN (*number*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>number</i>	The tangent of the angle.

Remarks

The resulting angle is returned in radians, ranging from $-\pi/2$ to $\pi/2$. To convert the resulting radians to degrees, multiply the radians by $180/\pi$ ().

Examples

This function returns 1.29:

```
ATAN (3.5)
```

This function returns -1.33:

```
ATAN (-4)
```

See Also

[ATAN2](#)

[ATANH](#)

[PI](#)

[TAN](#)

ATAN2

Description

Returns the arctangent of the specified coordinates.

Syntax

ATAN2 (*x*, *y*)

<u>Parameter</u>	<u>Description</u>
<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.

Remarks

The arctangent is the angle from the x axis to a line with end points at the origin (0, 0) and a point with the given coordinates (x, y). The angle is returned in radians, ranging from $-\pi$ to π , excluding $-\pi$.

Examples

This function returns 1.11:

```
ATAN2 (3, 6)
```

This function returns 3.04:

```
ATAN2 (-1, .1)
```

See Also

[ATAN](#)

[ATANH](#)

[PI](#)

[TAN](#)

ATANH

Description

Returns the inverse hyperbolic tangent of a number.

Syntax

ATANH (*number*)

Parameter	Description
<i>r</i>	
<i>number</i>	A number between -1 and 1, excluding -1 and 1.

Examples

This function returns .55:

```
ATANH (.5)
```

This function returns -.26:

```
ATANH (-.25)
```

See Also

[ACOS](#)

[ASINH](#)

[TANH](#)

AVERAGE

Description

Returns the average of the supplied numbers. The result of **AVERAGE** is also known as the arithmetic mean.

Syntax

AVERAGE (*number_list*)

Parameter	Description
<i>number_list</i>	A list of numbers separated by commas. As many as 30 numbers can be included in the list, and the list can contain numbers or a reference to a range that contains numbers. Text, logical expressions, or empty cells in a referenced range are ignored. All numeric values (including 0) are used.

Examples

This function returns 8.25:

```
AVERAGE(5, 6, 8, 14)
```

This function returns 134, the average of the values in the range C15:C17:

```
AVERAGE(C15:C17)
```

See Also

[MIN](#)

[MAX](#)

CEILING

Description

Rounds a number up to the nearest multiple of a specified significance.

Syntax

CEILING (*number*, *significance*)

<u>Parameter</u>	<u>Description</u>
<i>number</i>	The value to round.
<i>significance</i>	The multiple to which to round.

Remarks

Regardless of the sign of the number, the value is rounded up, away from zero. If number is an exact multiple of significance, no rounding occurs.

If number or significance is non-numeric, the error #VALUE! is returned. When the arguments have opposite signs, the error #NUM! is returned.

Examples

This function returns 1.25:

```
CEILING(1.23459, .05)
```

This function returns -150:

```
CEILING(-148.24, -2)
```

See Also

[EVEN](#)

[FLOOR](#)

[INT](#)

[ODD](#)

[ROUND](#)

[TRUNC](#)

CHAR

Description

Returns a character that corresponds to the supplied ASCII code.

Syntax

CHAR (*number*)

Parameter	Description
<u>r</u> <i>number</i>	A value between 1 and 255 that specifies an ASCII character.

Remarks

The character and associated numeric code are defined by Windows in the ASCII character set.

Examples

This function returns F:

`CHAR(70)`

This function returns #:

`CHAR(35)`

See Also

[CODE](#)

CHOOSE

Description

Returns a value from a list of numbers based on the index number supplied.

Syntax

CHOOSE (*index*, *item_list*)

Parameter	Description
<i>r</i>	
<i>index</i>	A number that refers to an item in <i>item_list</i> .
<i>item_list</i>	A list of numbers, formulas, or text separated by commas. This argument can also be a range reference. You can specify as many as 29 items in the list.

Remarks

Index can be a cell reference; *index* can also be a formula that returns any value from 1 to 29. If *index* is less than 1 or greater than the number of items in *item_list*, #VALUE! is returned. If *index* is a fractional number, it is truncated to an integer.

Examples

This function returns Q2:

```
CHOOSE(2,"Q1", "Q2", "Q3", "Q4")
```

This function returns the average of the contents of range A1:A10:

```
AVERAGE(CHOOSE(1, A1:A10, B1:B10, C1:C10))
```

See Also

[INDEX](#)

CLEAN

Description

Removes all nonprintable characters from the supplied text.

Syntax

CLEAN (*text*)

Parameter	Description
<i>r</i>	
<i>text</i>	Any worksheet information.

Remarks

Text that is imported from another environment may require this function.

Examples

This function returns Payments Due because the character returned by CHAR (8) is nonprintable:

```
CLEAN("Payments " & CHAR(8) & "Due")
```

See Also

[CHAR](#)

[TRIM](#)

CODE

Description

Returns a numeric code representing the first character of the supplied string.

Syntax

CODE (*text*)

r	Descriptio
<u><i>text</i></u>	<u><i>n</i></u> Any string.

Remarks

The numeric code and associated string are defined in your computer's character set. The character set used by Windows is the ANSI character set.

Examples

This function returns 65:

```
CODE ("A")
```

This function returns 98:

```
CODE ("b")
```

See Also

[**CHAR**](#)

COLUMN

Description

Returns the column number of the supplied reference.

Syntax

COLUMN (reference)

Parameter	Description
<i>r</i> <i>reference</i>	A reference to a cell or range. Omitting the argument returns the number of the column in which COLUMN is placed.

Examples

This function returns 2:

`COLUMN (B3)`

This function returns 4 if the function is entered in cell D2:

`COLUMN ()`

See Also

[COLUMNS](#)

[ROW](#)

COLUMNS

Description

Returns the number of columns in a range reference.

Syntax

COLUMNS (*range*)

<u>Parameter</u>	<u>Description</u>
<u>r</u>	
<i>range</i>	A reference to a range of cells.

Example

This function returns 4:

```
COLUMNS (A1 : D5)
```

See Also

[COLUMN](#)

[ROWS](#)

CONCATENATE

Description

Joins several text strings into one string.

Syntax

CONCATENATE (*text1*, *text2*,)

<u>Parameter</u>	<u>Description</u>
<i>text1</i> ,	Up to 30 text items to be joined into a single text item. The text items can be strings, numbers, or single-cell references.
<i>text2</i> , ...	

Remarks

The "&" operator can be used instead of CONCATENATE to join text items.

Examples

The following example returns "Sale Price" it is the same as typing "Sale"&" "&"Price":

```
CONCATENATE ("Sale ", "Price")
```

Suppose in an inventory worksheet, C2 contains "extruder1", C5 contains " gaskets", and C8 contains the number 15. The following example returns "Inventory currently holds 15 gaskets for extruder1.":

```
CONCATENATE ("Inventory currently holds ", C8, " ", C5," for ", C2)
```

See Also

[COLUMN](#)

[ROWS](#)

COS

Description

Returns the cosine of an angle.

Syntax

COS (*number*)

<u>Parameter</u>	<u>Description</u>
<i>number</i>	The angle in radians. If the angle is in degrees, convert the angle to radians by multiplying the angle by PI()/180.

Examples

This function returns .126:

```
COS(1.444)
```

This function returns .28:

```
COS(5)
```

See Also

[ACOS](#)

[ASINH](#)

[ATANH](#)

[COSH](#)

[PI](#)

COSH

Description

Returns the hyperbolic cosine of a number.

Syntax

COSH (*number*)

r	Descriptio
<u><i>number</i></u>	<u><i>n</i></u> Any number.

Examples

This function returns 4.14:

```
COSH (2.10)
```

This function returns 1.03:

```
COSH (.24)
```

See Also

[ASINH](#)

[ATANH](#)

[COS](#)

COUNT

Description

Returns the number of values in the supplied list.

Syntax

COUNT (*value_list*)

Parameter	Description
<i>value_list</i>	A list of values. The list can contain as many as 30 values.

Remarks

COUNT only numerates numbers or numerical values such as logical values, dates, or text representations of dates. If you supply a range, only numbers and numerical values in the range are counted. Empty cells, logical values, text, and error values in the range are ignored.

Examples

This function returns 2:

```
COUNT(5, 6, "Q2")
```

This function returns 3:

```
COUNT("03/06/94", "06/21/94", "10/19/94")
```

See Also

[AVERAGE](#)

[COUNTA](#)

[SUM](#)

COUNTIF

Description

Returns the number of cells within a range which meet the given criteria.

Syntax

COUNTIF (*range*, *criteria*)

Parameter	Description
<i>r</i>	
<i>range</i>	Range of cells you want to count.
<i>criteria</i>	Number, expression, or text that defines which cells are counted.

See Also

[**AVERAGE**](#)

[**COUNTA**](#)

[**SUM**](#)

[**SUMIF**](#)

COUNTA

Description

Returns the number of nonblank values in the supplied list.

Syntax

COUNTA (*expression_list*)

<u>Parameter</u>	<u>Description</u>
<i>expression_list</i>	A list of expressions. As many as 30 expressions can be included in the list.

Remarks

COUNTA returns the number of cells that contain data in a range. Null values ("") are counted, but references to empty cells are ignored.

Examples

This function returns 4:

```
COUNTA(32, 45, "Earnings", "")
```

This function returns 0 when the specified range contains empty cells:

```
COUNTA(C38:C40)
```

See Also

[AVERAGE](#)

[COUNT](#)

[PRODUCT](#)

[SUM](#)

DATE

Description

Returns the serial number of the supplied date.

Syntax

DATE (*year, month, day*)

Parameter	Description
<i>year</i>	A number from 1900 to 2078. If year is between 1920 to 2019, you can specify two digits to represent the year; otherwise specify all four digits.
<i>month</i>	A number representing the month (for example, 12 represents December). If a number greater than 12 is supplied, the number is added to the first month of the specified year.
<i>day</i>	A number representing the day of the month. If the number you specify for day exceeds the number of days in that month, the number is added to the first day of the specified month.

Examples

This function returns 34506:

```
DATE(94, 6, 21)
```

This function returns 36225:

```
DATE(99, 3, 6)
```

See Also

[DATEVALUE](#)

[DAY](#)

[MONTH](#)

[NOW](#)

[TIMEVALUE](#)

[TODAY](#)

[YEAR](#)

DATEVALUE

Description

Returns the serial number of a date supplied as a text string.

Syntax

DATEVALUE (*text*)

Parameter	Description
-----------	-------------

<i>text</i>	A date in text format between January 1, 1900, and December 31, 2078. If you omit the year, the current year is used.
-------------	---

Examples

This function returns 34399:

`DATEVALUE ("3/6/94")`

This function returns 35058:

`DATEVALUE ("12/25/95")`

See Also

[NOW](#)

[TIMEVALUE](#)

[TODAY](#)

DAY

Description

Returns the day of the month that corresponds to the date represented by the supplied number.

Syntax

DAY (*serial_number*)

<u>Parameter</u>	<u>Description</u>
<i>serial_number</i>	A date represented as a serial number or as text (for example, 06-21-94 or 21-Jun-94).

Examples

This function returns 6:

```
DAY(34399)
```

This function returns 21:

```
DAY("06-21-94")
```

See Also

[NOW](#)

[HOUR](#)

[MINUTE](#)

[MONTH](#)

[SECOND](#)

[TODAY](#)

[WEEKDAY](#)

[YEAR](#)

DB

Description

Returns the real depreciation of an asset for a specific period of time using the fixed-declining balance method.

Syntax

DB (*cost, salvage, life, period* [, *months*])

Paramete	Description
r	
<i>cost</i>	The initial cost of the asset.
<i>salvage</i>	The salvage value of the asset.
<i>life</i>	The number of periods in the useful life of the asset.
<i>period</i>	The period for which to calculate the depreciation. The time units used to determine period and life must match.
<i>months</i>	The number of months in the first year of the item's life. Omitting this argument assumes there are 12 months in the first year.

Example

This function returns 1451.52:

```
DB(10000, 1000, 7, 3)
```

See Also

[DDB](#)

[SLN](#)

[SYD](#)

[VDB](#)

DDB

Description

Returns the depreciation of an asset for a specific period of time using the double-declining balance method or a declining balance factor you supply.

Syntax

DDB (*cost, salvage, life, period* [, *factor*])

Parameter	Description
<i>cost</i>	The initial cost of the asset.
<i>salvage</i>	The salvage value of the asset.
<i>life</i>	The number of periods in the useful life of the asset.
<i>period</i>	The period for which to calculate the depreciation. The time units used to determine period and life must match.
<i>factor</i>	The rate at which the balance declines. Omitting this argument assumes a default factor of 2, the double-declining balance factor.

Remarks

The double-declining balance method uses an accelerated rate where the highest depreciation occurs in the first period, decreasing in successive periods.

All arguments for this function must be positive numbers.

Example

This function returns 1457.73:

```
DDB(10000,1000, 7, 3)
```

See Also

[DB](#)

[SLN](#)

[SYD](#)

[VDB](#)

DOLLAR

Description

Returns the specified number as text, using currency format and the supplied precision.

Syntax

DOLLAR (*number* [, *precision*])

Parameter	Description
<i>r</i>	
<i>number</i>	A number, a formula that evaluates to a number, or a reference to a cell that contains a number.
<i>precision</i>	A value representing the number of decimal places to the right of the decimal point. Omitting this argument assumes two decimal places.

Examples

This function returns \$1023.79:

```
DOLLAR(1023.789)
```

This function returns \$500:

```
DOLLAR(495.301, -2)
```

See Also

[FIXED](#)

[TEXT](#)

[VALUE](#)

ERROR.TYPE

Description

Returns a number corresponding to an *error*.

Syntax

ERROR.TYPE (*error_ref*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>error_ref</i>	A cell reference.

Remarks

The following error text or numbers can be returned by this function.

<u>Number</u>	<u>Description</u>
1	#NULL!
2	#DIV/0!
3	#VALUE!
4	#REF!
5	#NAME?
6	#NUM!
7	#N/A
#N/A	Other

Example

This function returns 2 if the formula in cell A1 attempts to divide by zero:

```
ERROR.TYPE(A1)
```

See Also

[ISERR](#)

[ISERROR](#)

EVEN

Description

Rounds the specified number up to the nearest even integer.

Syntax

EVEN (*number*)

Parameter	Description
<i>r</i> <i>number</i>	Any number, a formula that evaluates to a number, or a reference to a cell that contains a number.

Examples

This function returns 4:

`EVEN(2.5)`

This function returns 2032:

`EVEN(2030.45)`

See Also

[**CEILING**](#)

[**FLOOR**](#)

[**INT**](#)

[**ODD**](#)

[**ROUND**](#)

[**TRUNC**](#)

EXACT

Description

Compares two expressions for identical, case-sensitive matches. True is returned if the expressions are identical; False is returned if they are not.

Syntax

EXACT (*expression1*, *expression2*)

<u>Parameter</u>	<u>Description</u>
<i>expression</i> 1	Any text.
<i>expression</i> 2	Any text.

Examples

This function returns True:

```
EXACT("Match", "Match")
```

This function returns False:

```
EXACT("Match", "match")
```

See Also

[LEN](#)

[SEARCH](#)

EXP

Description

Returns e raised to the specified power. The constant e is 2.71828182845904 (the base of the natural logarithm).

Syntax

EXP (*number*)

Paramete	Description
<u>r</u> <i>number</i>	Any number as the exponent.

Examples

This function returns 12.18:

EXP (2.5)

This function returns 20.09:

EXP (3)

See Also

[LN](#)

[LOG](#)

FACT

Description

Returns the factorial of a specified number.

Syntax

FACT (*number*)

Parameter	Description
<i>r</i>	
<i>number</i>	Any non-negative integer. If you supply a real number, FACT truncates the number to an integer before calculation.

Examples

This function returns 2:

```
FACT (2.5)
```

This function returns 720:

```
FACT (6)
```

See Also

[PRODUCT](#)

FALSE

Description

Returns the logical value False. This function always requires the trailing parentheses.

Syntax

FALSE ()

See Also

[TRUE](#)

FIND

Description

Searches for a string of text within another text string and returns the character position at which the search string first occurs.

Syntax

FIND (search_text, text [, start_position])

<u>Parameter</u>	<u>Description</u>
search_text	The text to find. If you specify an empty string (""), FIND matches the first character in text.
text	The text to be searched.
start_position	The character position in text where the search begins. The first character in text is character number 1. When you omit this argument, the default starting position is character number 1.

Remarks

FIND is case-sensitive. You cannot use wildcard characters in the search_text.

Examples

This function returns 12:

```
FIND("time", "There's no time like the present")
```

This function returns 19:

```
FIND("4", "Aisle 4, Part 123-4-11", 9)
```

See Also

[EXACT](#)

[LEN](#)

[MID](#)

[SEARCH](#)

FIXED

Description

Rounds a number to the supplied precision, formats the number in decimal format, and returns the result as text.

Syntax

FIXED (*number* [, *precision*][, *no_commas*])

Parameter	Description
<i>number</i>	Any number.
<i>precision</i>	The number of digits that appear to the right of the decimal place. When this argument is omitted, a default precision of 2 is used. If you specify negative precision, number is rounded to the left of the decimal point. You can specify a precision as great as 127 digits.
<i>no_commas</i>	Determines if thousands separators (commas) are used in the result. Use 1 to exclude commas in the result. If <i>no_commas</i> is 0 or the argument is omitted, thousands separators are included (for example, 1,000.00).

Examples

This function returns 2,000.500:

```
FIXED(2000.5, 3)
```

This function returns 2010:

```
FIXED(2009.5, -1, 1)
```

See Also

[DOLLAR](#)

[ROUND](#)

[TEXT](#)

[VALUE](#)

FLOOR

Description

Rounds a number down to the nearest multiple of a specified significance.

Syntax

FLOOR (*number*, *significance*)

<u>Parameter</u>	<u>Description</u>
<i>number</i>	The value to round.
<i>significance</i>	The multiple to which to round.

Remarks

Regardless of the sign of the number, the value is rounded down, toward zero. If number is an exact multiple of significance, no rounding occurs.

If number or significance is non-numeric, #NAME? is returned. When the arguments have opposite signs, #NUM! is returned.

Examples

This function returns 1.2:

```
FLOOR(1.23459, .05)
```

This function returns -148:

```
FLOOR(-148.24, -2)
```

See Also

[CEILING](#)

[EVEN](#)

[INT](#)

[ODD](#)

[ROUND](#)

[TRUNC](#)

FV

Description

Returns the future value of an annuity based on regular payments and a fixed interest rate.

Syntax

FV (*interest*, *nper*, *payment* [, *pv*] [, *type*])

Parameter	Description
<i>r</i>	
<i>interest</i>	The fixed interest rate.
<i>nper</i>	The number of payments in an annuity.
<i>payment</i>	The fixed payment made each period.
<i>pv</i>	The present value, or the lump sum amount, the annuity is currently worth. When you omit this argument, a present value of 0 is assumed.
<i>type</i>	Indicates when payments are due. Use 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. When you omit this argument, 0 is assumed.

Remarks

The units used for interest must match those used for nper. For example, if the annuity has an 8 percent annual interest rate over a period of 5 years, specify 8 percent/12 for interest and 5*12 for nper.

Cash paid out, such as a payment, is shown as a negative number. Cash received, such as a dividend check, is shown as a positive number.

Examples

This function returns 4,774.55:

```
FV(5%, 8, -500)
```

This function returns 531,550.86:

```
FV(10%/12, 240, -700, 1)
```

See Also

[IPMT](#)

[NPER](#)

[PMT](#)

[PPMT](#)

[PV](#)

[RATE](#)

HLOOKUP

Description

Searches the top row of a table for a value and returns the contents of a cell in that table that corresponds to the location of the search value.

Syntax

HLOOKUP (*search_item*, *search_range*, *row_index*)

Parameter	Description
<i>search_item</i>	A value, text string, or reference to a cell containing a value that is matched against data in the top row of <i>search_range</i> .
<i>search_range</i>	A reference to the range (table) to be searched. The cells in the first row of <i>search_range</i> can contain numbers, text, or logical values. The contents of the first row must be in ascending order (for example, -2, -1, 0, 2...A through Z, False, True). Text searches are not case-sensitive.
<i>row_index</i>	The row in <i>search_range</i> from which the matching value is returned. <i>row_index</i> can be a number from 1 to the number of rows in <i>search_range</i> . If <i>row_index</i> is less than 1, the error #VALUE! is returned. When <i>row_index</i> is greater than the number of rows in the table, the error #REF! is returned.

Remarks

HLOOKUP compares the information in the top row of *search_range* to the supplied *search_item*. When a match is found, information located in the same column and supplied row (*row_index*) is returned.

If *search_item* cannot be found in the top row of *search_range*, the largest value that is less than *search_item* is used. When *search_item* is less than the smallest value in the first row of the *search_range*, the error #REF! is returned.

Examples

The following examples use this worksheet.

	A	B	C	D	E
1		Midwest	Northeast	Pacific	South
2	Q1	48.23	278.21	61.97	164.80
3	Q2	163.83	22.63	161.73	183.96
4	Q3	43.96	233.56	278.16	171.98
5	Q4	245.69	167.09	245.23	163.00

This function returns 22.63:

```
HLOOKUP("Northeast", B1:E5, 3)
```

This function returns #REF!:

```
HLOOKUP("Pacific", B1:E5, 7)
```

See Also

[INDEX](#)

[LOOKUP](#)

[MATCH](#)

[VLOOKUP](#)

HOUR

Description

Returns the hour component of the specified time in 24-hour format.

Syntax

HOUR (*serial_number*)

Parameter	Description
<i>serial_number</i>	The time as a serial number. The decimal portion of the number represents time as a fraction of the day.

Remarks

The result is an integer ranging from 0 (12:00 AM) to 23 (11:00 PM).

Examples

This function returns 9:

```
HOUR(34259.4)
```

This function returns 23:

```
HOUR(34619.976)
```

See Also

[DAY](#)

[MINUTE](#)

[MONTH](#)

[NOW](#)

[SECOND](#)

[WEEKDAY](#)

[YEAR](#)

IF

Description

Tests the condition and returns the specified value.

Syntax

IF (*condition*, *true_value*, *false_value*)

Parameter	Description
<i>condition</i>	Any logical expression.
<i>true_value</i>	The value to be returned if condition evaluates to True.
<i>false_value</i>	The value to be returned if condition evaluates to False.

Example

This function returns Greater if the contents of A1 is greater than 10 and Less if the contents of A1 is less than 10:

```
IF(A1>10, "Greater", "Less")
```

See Also

[AND](#)

[FALSE](#)

[NOT](#)

[OR](#)

[TRUE](#)

INDEX

Description

Returns the contents of a cell from a specified range.

Syntax

INDEX (*reference* [, *row*] [, *column*] [, *range_number*])

Parameter	Description
<i>reference</i>	A reference to one or more ranges. If reference specifies more than one range, separate each reference with a comma and enclose reference in parentheses. For example, (A1:C6, B7:E14, F4). If each range in reference contains only one row or column, you can omit the row or column argument. For example, if reference is A1:A15, you can omit the column argument INDEX(A1:A15, 3,, 1).
<i>row</i>	The row number in reference from which to return data.
<i>column</i>	Column number in reference from which to return data.
<i>range_number</i>	Specifies the range from which data is returned if reference contains more than one range.
<i>r</i>	For example, if reference is (A1:A10, B1:B5, D14:E23), A1:A10 is range_number 1, B1:B5 is range_number 2, and D14:E23 is range_number 3.

Remarks

If row, column, and range_number do not point to a cell within reference, #REF! is returned. If row and column are omitted, INDEX returns the range in reference specified by range_number.

Examples

The following examples use this worksheet.

	A	B	C	D	E
1	Sales Group 1			Sales Group 2	
2	Adams	\$1,225.14		Cash	\$1,819.47
3	Baker	\$1,415.35		Johnson	\$1,733.67
4	Martinez	\$1,573.57		Nelson	\$1,138.23
5	Smith	\$1,469.78		Randall	\$1,634.58
6	White	\$1,390.89		Schultz	\$1,093.82

This function returns \$1415.35:

INDEX(A2:B6, 2, 2)

This function returns \$1634.58:

INDEX((A2:B6, D2:E6), 4, 2, 2)

See Also

[CHOOSE](#)

[HLOOKUP](#)

[LOOKUP](#)

[MATCH](#)

[VLOOKUP](#)

INDIRECT

Description

Returns the contents of the cell referenced by the specified cell.

Syntax

INDIRECT (*ref_text* [, *a1*])

Paramete	Description
-----------------	--------------------

ref_text

A reference to a cell that references a third cell. If *ref_text* is not a valid reference, the error #REF! is returned.

a1

The reference format. This argument must be TRUE() to represent an A1 reference format; Formula One does not support the R1C1 reference format.

Example

This function returns the contents of the cell that C1 references. If C1 contains "D1," then the contents of D1 is returned:

INDIRECT (C1)

See Also

[**OFFSET**](#)

INT

Description

Rounds the supplied number down to the nearest integer.

Syntax

INT (*number*)

Parameter	Description
<i>r</i> <i>number</i>	Any real number.

Examples

This function returns 10:

```
INT(10.99)
```

This function returns -11:

```
INT(-10.99)
```

See Also

[CEILING](#)

[FLOOR](#)

[MOD](#)

[ROUND](#)

[TRUNC](#)

IPMT

Description

Returns the interest payment of an annuity for a given period, based on regular payments and a fixed periodic interest rate.

Syntax

IPMT (*interest*, *per*, *nper*, *pv*, [*fv*], [*type*])

Paramete	Description
<u>r</u>	
<i>interest</i>	The fixed periodic interest rate.
<i>per</i>	The period for which to return the interest payment. This number must be between 1 and nper.
<i>nper</i>	The number of payments.
<i>pv</i>	The present value, or the lump sum amount the annuity is currently worth.
<i>fv</i>	The future value, or the value after all payments are made. If this argument is omitted, the future value is assumed to be 0.
<i>type</i>	Indicates when payments are due. Use 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. When you omit this argument, 0 is assumed.

Remarks

The units used for interest must match those used for nper. For example, if the annuity has an 8 percent annual interest rate over a period of 5 years, specify 8 percent/12 for interest and 5*12 for nper.

Cash paid out, such as a payment, is shown as a negative number. Cash received, such as a dividend check, is shown as a positive number.

Examples

This function returns -117.87:

```
IPMT(8%/12, 2, 48, 18000)
```

This function returns -117.09:

```
IPMT(8%/12, 2, 48, 18000, 0, 1)
```

See Also

[FV](#)

[PMT](#)

[PPMT](#)

[RATE](#)

IRR

Description

Returns internal rate of return for a series of periodic cash flows.

Syntax

IRR (*cash_flow* [, *guess*])

Parameter	Description
<i>r</i>	
<i>cash_flow</i>	A reference to a range that contains values for which to calculate the internal rate of return. The values must contain at least one positive and one negative value. During calculation, IRR uses the order in which the values appear to determine the order of the cash flow. Text, logical values, and empty cells in the range are ignored.
<i>guess</i>	The estimate of the internal rate of return. If no argument is supplied, a rate of return of 10 percent is assumed.

Remarks

The internal rate of return is the interest rate received for an investment consisting of payments (specified by negative numbers) and investments (specified by positive numbers). **IRR** is calculated iteratively, cycling through the calculation until the result is accurate to .00001 percent. If the result cannot be found after 20 iterations, #NUM! is returned. When this occurs, supply a different value for guess.

Examples

The following examples use this worksheet.

	A	B
1	Investment	(\$60,000.00)
2	1989 income	\$9,590.00
3	1990 income	\$10,580.00
4	1991 income	\$12,790.00
5	1992 income	\$15,830.00
6	1993 income	\$18,930.00

This function returns 3.72 percent:

`IRR(B1:B6)`

This function returns -49.26 percent:

`IRR(B1:B3, -20%)`

See Also

[MIRR](#)

[NPV](#)

[RATE](#)

ISBLANK

Description

Determines if the specified cell is blank.

Syntax

ISBLANK (*reference*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>reference</i>	A reference to any cell.

Remarks

If the referenced cell is blank, True is returned. False is returned if the cell is not blank.

Example

This function returns True if A1 is a blank cell:

```
ISBLANK (A1)
```

See Also

[ISERR](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNONTEXT](#)

[ISNUMBER](#)

[ISREF](#)

[ISTEXT](#)

ISERR

Description

Determines if the specified expression returns an error value.

Syntax

ISERR (*expression*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>expression</i>	Any expression.

Remarks

If the expression returns any error except #N/A!, True is returned. Otherwise, False is returned.

Example

This function returns True if A1 contains a formula that returns an error such as #NUM!:

```
ISERR(A1)
```

See Also

[ISBLANK](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNONTTEXT](#)

[ISNUMBER](#)

[ISREF](#)

[ISTEXT](#)

ISERROR

Description

Determines if the specified expression returns an error value.

Syntax

ISERROR (*expression*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>expression</i>	Any expression.

Remarks

If the expression returns any error value, such as #N/A!, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!, True is returned. Otherwise, False is returned.

Examples

This function returns True:

`ISERROR(4/0)`

This function returns False if A1 contains a formula that does not return an error.

`ISERROR(A1)`

See Also

[ISBLANK](#)

[ISERR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNONTTEXT](#)

[ISNUMBER](#)

[ISREF](#)

[ISTEXT](#)

ISLOGICAL

Description

Determines if the specified expression returns a logical value.

Syntax

ISLOGICAL (*expression*)

<u>Parameter</u>	<u>Description</u>
<i>r</i> <i>expression</i>	Any expression.

Remarks

If the expression returns a logical value, True is returned. Otherwise, False is returned.

Example

This function returns True because ISBLANK returns a logical value:

```
ISLOGICAL ( ISBLANK ( A1 ) )
```

See Also

[ISBLANK](#)

[ISERR](#)

[ISERROR](#)

[ISNA](#)

[ISNONTEXT](#)

[ISNUMBER](#)

[ISREF](#)

[ISTEXT](#)

ISNA

Description

Determines if the specified expression returns the value not available error.

Syntax

ISNA (*expression*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>expression</i>	Any expression.

Remarks

If the expression returns the #N/A! error, True is returned. Otherwise, False is returned.

Example

This function returns True if cell A1 contains the NA () function or returns the error value #N/A!:

ISNA(A1)

See Also

[ISBLANK](#)

[ISERR](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNONTTEXT](#)

[ISNUMBER](#)

[ISREF](#)

[ISTEXT](#)

ISNONTEXT

Description

Determines if the specified expression is not text.

Syntax

ISNONTEXT (*expression*)

<u>Parameter</u>	<u>Description</u>
<i>r</i> <i>expression</i>	Any expression.

Remarks

If the expression returns any value that is not text, True is returned. Otherwise, False is returned.

Examples

This function returns True if cell F3 contains a number or is a blank cell:

```
ISNONTEXT (F3)
```

This function returns False:

```
ISNONTEXT ("text")
```

See Also

[ISBLANK](#)

[ISERR](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNUMBER](#)

[ISREF](#)

[ISTEXT](#)

ISNUMBER

Description

Determines if the specified expression is a number.

Syntax

ISNUMBER (*expression*)

<u>Parameter</u>	<u>Description</u>
<i>r</i> <i>expression</i>	Any expression.

Remarks

If the expression returns a number, True is returned. Otherwise, False is returned. If expression returns a number represented as text (for example, "12"), False is returned.

Examples

This function returns True:

```
ISNUMBER(123.45)
```

This function returns False:

```
ISNUMBER("123")
```

See Also

[ISBLANK](#)

[ISERR](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNONTEXT](#)

[ISREF](#)

[ISTEXT](#)

ISREF

Description

Determines if the specified expression is a range reference.

Syntax

ISREF (*expression*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>expression</i>	Any expression.

Remarks

If the expression returns a range reference, True is returned. Otherwise, False is returned.

Example

This function returns True:

```
ISREF(A3)
```

See Also

[ISBLANK](#)

[ISERR](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNONTTEXT](#)

[ISNUMBER](#)

[ISTEXT](#)

ISTEXT

Description

Determines if the specified expression is text.

Syntax

ISTEXT (*expression*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>expression</i>	Any expression.

Remarks

If the expression returns text, True is returned. Otherwise, False is returned.

Example

This function returns True:

```
ISTEXT("2nd Quarter")
```

See Also

[ISBLANK](#)

[ISERR](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNONTTEXT](#)

[ISNUMBER](#)

[ISREF](#)

LEFT

Description

Returns the leftmost characters from the specified text string.

Syntax

LEFT (*text* [, *num_chars*])

Parameter	Description
<i>r</i>	
<i>text</i>	Any text string.
<i>num_chars</i>	The number of characters to return. This value must be greater than or equal to zero. If <i>num_chars</i> is greater than the number of characters in <i>text</i> , the entire string is returned. Omitting this argument assumes a value of 1.

Examples

This function returns 2:

```
LEFT("2nd Quarter")
```

This function returns 2nd:

```
LEFT("2nd Quarter", 3)
```

See Also

[MID](#)

[RIGHT](#)

LEN

Description

Returns the number of characters in the supplied text string.

Syntax

LEN (*text*)

Parameter	Description
<i>r</i>	
<i>text</i>	Any text string. Spaces in the string are counted as characters.

Examples

This function returns 11:

```
LEN("3rd Quarter")
```

This function returns 3:

```
LEN("1-3")
```

See Also

[EXACT](#)

[SEARCH](#)

LN

Description

Returns the natural logarithm (based on the constant e) of a number.

Syntax

LN (*number*)

<u>Parameter</u>	<u>Description</u>
<i>r</i> <i>number</i>	Any positive real number.

Remarks

LN is the inverse of the **EXP** function.

Examples

This function returns 2.50:

`LN(12.18)`

This function returns 3.00:

`LN(20.09)`

See Also

[EXP](#)

[LOG](#)

[LOG10](#)

LOG

Description

Returns the logarithm of a number to the specified base.

Syntax

LOG (*number* [, *base*])

Parameter	Description
<i>r</i>	
<i>number</i>	Any positive real number.
<i>base</i>	The base of the logarithm. Omitting this argument assumes a base of 10.

Examples

This function returns 0:

```
LOG (1)
```

This function returns 1:

```
LOG (10)
```

See Also

[EXP](#)

[LN](#)

[LOG10](#)

LOG10

Description

Returns the base-10 logarithm of a number.

Syntax

LOG10 (*number*)

<u>Parameter</u>	<u>Description</u>
<i>r</i> <i>number</i>	Any positive real number.

Examples

This function returns 2.41:

```
LOG10(260)
```

This function returns 2:

```
LOG10(100)
```

See Also

[EXP](#)

[LN](#)

[LOG](#)

LOOKUP

Description

Searches for a value in one range and returns the contents of the corresponding position in a second range.

Syntax

LOOKUP (*lookup_value*, *lookup_range*, *result_range*)

<u>Parameter</u>	<u>Description</u>
<i>lookup_value</i>	The value for which to search in the first range.
<i>lookup_range</i>	The first range to search and contains only one row or one column. The range can contain numbers, text, or logical values. To search <i>lookup_range</i> correctly, the expressions in the range must be placed in ascending order (for example, -2, -1, 0, 1, 2...A through Z, False, True). The search is not case-sensitive.
<i>result_range</i>	A range of one row or one column that is the same size as <i>lookup_range</i> .

Remarks

If *lookup_value* does not have an exact match in *lookup_range*, the largest value that is less than or equal to *lookup_value* is found and the corresponding position in *result_range* is returned. When *lookup_value* is smaller than the data in *lookup_range*, #N/A is returned.

Examples

The following examples use this worksheet.

	A	B
1	Region	Headquarters
2	Midwest	Kansas City
3	North	Detroit
4	Northeast	Philadelphia
5	Pacific	Portland
6	South	Atlanta
7	Southwest	Phoenix

This function returns Detroit:

```
LOOKUP("North", A2:A7, B2:B7)
```

This function returns #N/A:

```
LOOKUP("Alabama", A2:A7, B2:B7)
```

See Also

[HLOOKUP](#)

[INDEX](#)

[VLOOKUP](#)

LOWER

Description

Changes the characters in the specified string to lowercase characters. Numeric characters in the string are not changed.

Syntax

LOWER (*text*)

r	Descriptio
<u>text</u>	<u>n</u> Any string.

Examples

This function returns 3rd quarter:

```
LOWER("3rd Quarter")
```

This function returns john doe:

```
LOWER("JOHN DOE")
```

See Also

[**PROPER**](#)

[**UPPER**](#)

MATCH

Description

A specified value is compared against values in a range. The position of the matching value in the search range is returned.

Syntax

MATCH (*lookup_value*, *lookup_range*, *comparison*)

Parameter	Description
<i>lookup_value</i>	The value against which to compare. It can be a number, text, or logical value or a reference to a cell that contains one of those values.
<i>lookup_range</i>	The range to search and contains only one row or one column. The range can contain numbers, text, or logical values.
<i>comparison</i>	A number that represents the type of comparison to be made between <i>lookup_value</i> and the values in <i>lookup_range</i> . When you omit this argument, comparison method 1 is assumed. When <i>comparison</i> is 0, the first value that is equal to <i>lookup_value</i> is matched. When using this comparison method, the values in <i>lookup_range</i> can be in any order. When <i>comparison</i> is 1, the largest value that is less than or equal to <i>lookup_value</i> is matched. When using this comparison method, the values in <i>lookup_range</i> must be in ascending order (for example, ...-2, -1, 0, 1, 2..., A through Z, False, True). When <i>comparison</i> is -1, the smallest value that is greater than or equal to <i>lookup_value</i> is matched. When using this comparison method, the values in <i>lookup_range</i> must be in descending order (for example, True, False, Z through A, ...2, 1, 0, -1, -2...).

Remarks

When using comparison method 0 and *lookup_value* is text, *lookup_value* can contain wildcard characters. The wildcard characters are * (asterisk), which matches any sequence of characters, and ? (question mark), which matches any single character.

When no match is found for *lookup_value*, #N/A is returned.

Examples

The following examples use this worksheet.

	A	B
1	Mfr. Code	Stock No.
2	BAJ	0677
3	DOD	0753
4	FMH	0816
5	JMF	0913
6	PLY	7534
7	TJL	7763

This function returns 5:

```
MATCH(7600, B2:B7,1)
```

This function returns 2:

```
MATCH("D*", A2:A7,0)
```

See Also

[HLOOKUP](#)

[INDEX](#)

[LOOKUP](#)

VLOOKUP

MAX

Description

Returns the largest value in the specified list of numbers.

Syntax

MAX (*number_list*)

Parameter	Description
<i>number_list</i>	A list of as many as 30 numbers, separated by commas. The list can contain numbers, logical values, text representations of numbers, or a reference to a range containing those values. Error values or text that cannot be translated into numbers return errors. If a range reference is included in the list, text, logical expressions, and empty cells in the range are ignored If there are no numbers in the list, 0 is returned.

Examples

This function returns 500:

```
MAX(50, 100, 150, 500, 200)
```

This function returns the largest value in the range:

```
MAX(A1:F12)
```

See Also

[AVERAGE](#)

[MIN](#)

MID

Description

Returns the specified number of characters from a text string, beginning with the specified starting position.

Syntax

MID (*text*, *start_position*, *num_chars*)

<u>Parameter</u>	<u>Description</u>
<i>text</i>	The string from which to return characters.
<i>start_position</i>	The position of the first character to return from text. If <i>start_position</i> is 1, the first character in text is returned. If <i>start_position</i> is greater than the number of characters in text, an empty string ("") is returned. If <i>start_position</i> is less than 1, #VALUE! is returned.
<i>num_chars</i>	The number of characters to return. If <i>num_chars</i> is negative, #VALUE! is returned.

Remarks

If *start_position* plus the number of characters in *num_chars* exceeds the length of text, the characters from *start_position* to the end of text are returned.

Examples

This function returns Expenses:

```
MID("Travel Expenses", 8, 8)
```

This function returns 45:

```
MID("Part #45-7234", 7, 2)
```

See Also

[CODE](#)

[FIND](#)

[LEFT](#)

[RIGHT](#)

[SEARCH](#)

MIN

Description

Returns the smallest value in the specified list of numbers.

Syntax

MIN (*number_list*)

Parameter	Description
<i>number_list</i>	A list of as many as 30 numbers, separated by commas. The list can contain numbers, logical values, text representations of numbers, or a reference to a range containing those values. Error values or text that cannot be translated into numbers return errors. If a range reference is included in the list, text, logical expressions, and empty cells in the range are ignored. If there are no numbers in the list, 0 is returned

Examples

This function returns 50:

```
MIN(50, 100, 150, 500, 200)
```

This function returns the smallest value in the range:

```
MIN(A1:F12)
```

See Also

[**AVERAGE**](#)

[**MAX**](#)

MINUTE

Description

Returns the minute that corresponds to the supplied date.

Syntax

MINUTE (*serial_number*)

<u>Parameter</u>	<u>Description</u>
<i>serial_number</i>	The time as a serial number. The decimal portion of the number represents time as a fraction of the day.

Remarks

The result is an integer ranging from 0 to 59.

Examples

This function returns 36:

```
MINUTE (34506.4)
```

This function returns 48:

```
MINUTE (34399.825)
```

See Also

[DAY](#)

[HOUR](#)

[MONTH](#)

[NOW](#)

[SECOND](#)

[WEEKDAY](#)

[YEAR](#)

MIRR

Description

Returns the modified internal rate of return for a series of periodic cash flows.

Syntax

MIRR (*cash_flows*, *finance_rate*, *reinvest_rate*)

Parameter	Description
<i>cash_flow</i>	A reference to a range that contains values for which to calculate the modified internal rate of return. The values must contain at least one positive and one negative value. Values that represent cash received should be positive; negative values represent cash paid. During calculation, MIRR uses the order in which the values appear to determine the order of cash flow. Text, logical values, and empty cells in the range are ignored.
<i>finance_rate</i>	The interest rate paid on money used in the cash flow.
<i>reinvest_rate</i>	The interest rate received on money reinvested from the cash flow.

Remarks

The modified internal rate of return considers the cost of the investment and the interest received on the reinvestment of cash.

Examples

The following examples use this worksheet.

This function returns 5.20 percent:

```
MIRR(B1:B6, 12%, 8%)
```

This function returns -40.93 percent:

```
MIRR(B1:B3, 12%, 8%)
```

See Also

[IRR](#)

[NPV](#)

[RATE](#)

MOD

Description

Returns the remainder after dividing a number by a specified divisor.

Syntax

MOD (*number*, *divisor*)

Parameter	Description
<i>r</i>	
<i>number</i>	Any number.
<i>divisor</i>	Any nonzero number. If divisor is 0, #DIV/0! is returned.

Examples

This function returns 1:

`MOD(-23, 3)`

This function returns -2:

`MOD(-23, -3)`

See Also

[INT](#)

[ROUND](#)

[TRUNC](#)

MONTH

Description

Returns the month that corresponds to the supplied date.

Syntax

MONTH (*serial_number*)

<u>Parameter</u>	<u>Description</u>
<i>serial_number</i>	The date as a serial number or as text (for example, 06-21-94 or 21-Jun-94).

Remarks

MONTH returns a number ranging from 1 (January) to 12 (December).

Examples

This function returns 6:

```
MONTH("06-21-94")
```

This function returns 10:

```
MONTH(34626)
```

See Also

[DAY](#)

[NOW](#)

[HOURL](#)

[MINUTE](#)

[SECOND](#)

[TODAY](#)

[WEEKDAY](#)

[YEAR](#)

N

Description

Tests the supplied value and returns the value if it is a number.

Syntax

N (*value*)

Parameter	Description
<i>value</i>	A value or a reference to a cell containing a value to test.

Remarks

Numbers are returned as numbers, serial numbers formatted as dates are returned as serial numbers, and the logical function TRUE() is returned as 1. All other expressions return 0.

Examples

This function returns 32467:

`N(32467)`

This function returns 1 if A4 contains the logical function TRUE:

`N(A4)`

See Also

[I](#)

[VALUE](#)

NA

Description

Returns the error value #N/A, which represents “not available.”

Syntax

NA ()

Remarks

Use **NA** to mark cells that lack data without leaving them empty. Empty cells may not be correctly represented in some calculations.

Although **NA** does not use arguments, you must supply the empty parentheses to correctly reference the function.

See Also

[ISNA](#)

NOT

Description

Returns a logical value that is the opposite of its value.

Syntax

NOT (*logical*)

Parameter	Description
<i>r</i>	
<i>logical</i>	An expression that returns a logical value such as True or False.

Remarks

If *logical* is false, **NOT** returns True. Conversely, if *logical* is true, **NOT** returns False.

Examples

This function returns False:

```
NOT (TRUE ())
```

This function returns False:

```
NOT (MONTH ("12/25/94") = 12)
```

See Also

[AND](#)

[IF](#)

[OR](#)

NOW

Description

Returns the current date and time as a serial number.

Syntax

NOW ()

Remarks

In a serial number, numbers to the left of the decimal point represent the date; numbers to the right of the decimal point represent the time. The result of this function changes only when a recalculation of the worksheet occurs.

See Also

[DATE](#)

[DAY](#)

[HOUR](#)

[MINUTE](#)

[MONTH](#)

[SECOND](#)

[TODAY](#)

[WEEKDAY](#)

[YEAR](#)

NPER

Description

Returns the number of periods of an investment based on regular periodic payments and a fixed interest rate.

Syntax

NPER (*interest*, *pmt*, *pf* [, *fv*] [, *type*])

Parameter	Description
<u>r</u>	
<i>interest</i>	The fixed interest rate.
<i>pmt</i>	The fixed payment made each period. Generally, pmt includes the principle and interest, not taxes or other fees.
<i>pf</i>	The present value, the lump-sum amount that a series of future payments is currently worth.
<i>fv</i>	The future value, the balance to attain after the final payment. Omitting this argument assumes a future balance of 0.
<i>type</i>	Indicates when payments are due. Use 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. When you omit this argument, 0 is assumed.

Examples

This function returns 36.67:

```
NPER(12%/12, -350, -300, 16000, 1)
```

This function returns 36.98:

```
NPER(1%, -350, -300, 16000)
```

See Also

[FV](#)

[IPMT](#)

[PMT](#)

[PPMT](#)

[PV](#)

[RATE](#)

NPV

Description

Returns the net present value of an investment based on a series of periodic payments and a discount rate.

Syntax

NPV (*discount_rate*, *value_list*)

Parameter	Description
<i>discount_rate</i>	The rate of discount for one period.
<i>value_list</i>	A list of as many as 29 arguments or a reference to a range that contains values that represent payments and income. During calculation, NPV uses the order in which the values appear to determine the order of cash flow. Numbers, empty cells, and text representations of numbers are included in the calculation. Errors and text that cannot be translated into numbers are ignored. If <i>value_list</i> is a range reference, only numeric data in the range is included in the calculation. Other types of data in the range, such as empty cells, logical values, text, and error values, are ignored.

Remarks

The time span **NPV** uses for calculation begins one period before the first cash flow date and ends when the last cash flow payment is made. This function is based on future cash flows. When your first cash flow occurs at the beginning of the first period, the first value must be added to the **NPV** result, not supplied as a value in *value_list*.

Example

This function returns 811.57:

```
NPV(8%, -12000, 3000, 3000, 3000, 7000)
```

See Also

[FV](#)

[IRR](#)

[PV](#)

ODD

Description

Rounds the specified number up to the nearest odd integer.

Syntax

ODD (*number*)

Parameter	Description
<i>number</i>	Any number, a formula that evaluates to a number, or a reference to a cell that contains a number.

Examples

This function returns 5:

ODD (3.5)

This function returns 7:

ODD (6)

See Also

[**CEILING**](#)

[**EVEN**](#)

[**FLOOR**](#)

[**INT**](#)

[**ROUND**](#)

[**TRUNC**](#)

OFFSET

Description

Returns the contents of a range that is offset from a starting point in the spreadsheet.

Syntax

OFFSET (*reference*, *rows*, *columns* [, *height*] [, *width*])

Parameter	Description
<i>reference</i>	A reference to a cell from which the offset reference is based. If you specify a range reference, #VALUE! is returned.
<i>rows</i>	The number of rows from reference that represents the upper-left cell of the offset range. A positive number represents rows below the starting cell; a negative number represents rows above the starting cell. If rows places the upper-left cell of the offset range outside the spreadsheet boundary, #REF! is returned.
<i>columns</i>	The number of columns from reference that represents the upper-left cell of the offset range. A positive number represents columns right of the starting cell; a negative number represents columns left of the starting cell. If columns places the upper-left cell of the offset range outside the spreadsheet boundary, #REF! is returned.
<i>height</i>	A positive number representing the number of rows to include in the offset range. Omitting this argument assumes a single row.
<i>width</i>	A positive number representing the number of columns to include in the offset range. Omitting this argument assumes a single column.

Remarks

OFFSET does not change the current selection in the worksheet. Because it returns a reference, **OFFSET** can be used in any function that requires or uses a cell or range reference as an argument.

Examples

This function returns the contents of cell D4:

```
OFFSET(B1, 3, 2, 1, 1)
```

This function returns the sum of the values in the range E3:F5:

```
SUM(OFFSET(A1, 2, 4, 3, 2))
```

OR

Description

Returns True if at least one of a series of logical arguments is true.

Syntax

OR (*logical_list*)

Parameter	Description
<i>logical_list</i>	A list of conditions separated by commas. You can include as many as 30 conditions in the list. The list can contain logical values or a reference to a range containing logical values. Text and empty cells are ignored. If there are no logical values in the list, the error value #VALUE! is returned.

Example

This function returns True because one of the arguments is true:

```
OR(1 + 1 = 1, 5 + 5 = 10)
```

See Also

[AND](#)

[IF](#)

[NOT](#)

PI

Description

Returns the value of pi (π), which is approximately 3.14159265358979 when calculated to 15 significant digits.

Syntax

PI ()

Remarks

Although **PI** does not use arguments, you must supply the empty parentheses to correctly reference the function.

See Also

[**COS**](#)

[**SIN**](#)

[**TAN**](#)

PMT

Description

Returns the periodic payment of an annuity, based on regular payments and a fixed periodic interest rate.

Syntax

PMT (*interest*, *nper*, *pv* [, *fv*] [, *type*])

Parameter	Description
<i>r</i>	
<i>interest</i>	The fixed periodic interest rate.
<i>nper</i>	The number of periods in the annuity.
<i>pv</i>	The present value, or the amount the annuity is currently worth.
<i>fv</i>	The future value, or the amount the annuity will be worth. When you omit this argument, a future value of 0 is assumed.
<i>type</i>	Indicates when payments are due. Use 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. When you omit this argument, 0 is assumed.

Remarks

PMT returns only the principal and interest payment, it does not include taxes or other fees. The units used for interest must match those used for *nper*. For example, if the annuity has an 8 percent annual interest rate over a period of 5 years, specify 8 percent/12 for interest and 5*12 for *nper*.

Cash paid out, such as a payment, is shown as a negative number. Cash received, such as a dividend check, is shown as a positive number.

Examples

This function returns -439.43:

```
PMT(8%/12, 48, 18000)
```

This function returns -436.52:

```
PMT(8%/12, 48, 18000, 0, 1)
```

See Also

[IPMT](#)

[FV](#)

[NPER](#)

[PPMT](#)

[PV](#)

[RATE](#)

PPMT

Description

Returns the principle paid on an annuity for a given period.

Syntax

PPMT (*interest*, *per*, *nper*, *pv*, [*fv*], [*type*])

Parameter	Description
<i>r</i>	The fixed periodic interest rate.
<i>per</i>	The period for which to return the principle.
<i>nper</i>	The number of periods in the annuity.
<i>pv</i>	The present value, or the amount the annuity is currently worth.
<i>fv</i>	The future value, or the amount the annuity will be worth.
	When you omit this argument, a future value of 0 is assumed.
<i>type</i>	Indicates when payments are due. Use 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. When you omit this argument, 0 is assumed.

Remarks

The units used for interest must match those used for nper. For example, if the annuity has an 8 percent annual interest rate over a period of 5 years, specify 8 percent/12 for interest and 5*12 for nper.

Examples

This function returns -321.56:

```
PPMT(8%/12, 2, 48, 18000)
```

This function returns -319.43:

```
PPMT(8%/12, 2, 48, 18000, 0, 1)
```

See Also

[FV](#)

[IPMT](#)

[NPER](#)

[PMT](#)

[PV](#)

[RATE](#)

PRODUCT

Description

Multiplies a list of numbers and returns the result.

Syntax

PRODUCT (*number_list*)

Parameter	Description
<i>number_list</i>	<p>A list of as many as 30 numbers, separated by commas. The list can contain numbers, logical values, text representations of numbers, or a reference to a range containing those values.</p> <p>Error values or text that cannot be translated into numbers return errors.</p> <p>If a range reference is included in the list, text, logical expressions, and empty cells in the range are ignored.</p> <p>All numeric values, including 0, are used in the calculation.</p>

Example

This function returns 24:

```
PRODUCT(1, 2, 3, 4)
```

See Also

[FACT](#)

[SUM](#)

PROPER

Description

Returns the specified string in proper-case format.

Syntax

PROPER (*text*)

<u>Parameter</u>	<u>Description</u>
<i>r</i> <i>text</i>	<i>n</i> Any string.

Remarks

In proper-case format, the first alphabetic character in a word is capitalized. If an alphabetic character follows a number, punctuation mark, or space, it is capitalized. All other alphabetic characters are lowercase. Numbers are not changed by **PROPER**.

Examples

This function returns 3rd Quarter:

```
PROPER("3rd Quarter")
```

This function returns John Doe:

```
PROPER("JOHN DOE")
```

See Also

[LOWER](#)

[UPPER](#)

PV

Description

Returns the present value of an annuity, considering a series of constant payments made over a regular payment period.

Syntax

PV (*interest*, *nper*, *pmt* [, *fv*] [, *type*])

Parameter	Description
r	
<i>interest</i>	The fixed periodic interest rate.
<i>nper</i>	The number of payment periods in the investment.
<i>pmt</i>	The fixed payment made each period.
<i>fv</i>	The future value, or the amount the annuity will be worth. When you omit this argument, a future value of 0 is assumed.
<i>type</i>	Indicates when payments are due. Use 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. When you omit this argument, 0 is assumed.

Remarks

The units used for *interest* must match those used for *nper*. For example, if the annuity has an 8 percent annual interest rate over a period of 5 years, specify 8 percent/12 for interest and 5*12 for *nper*.

Cash paid out, such as a payment, is shown as a negative number. Cash received, such as a dividend check, is shown as a positive number.

Examples

This function returns -17999.89:

```
PV(8%/12, 48, 439.43)
```

This function returns 17999.89:

```
PV(8%/12, 48, -439.43)
```

See Also

[FV](#)

[IPMT](#)

[NPER](#)

[PMT](#)

[PPMT](#)

[RATE](#)

RAND

Description

Returns a number selected randomly from a uniform distribution greater than or equal to 0 and less than 1.

Syntax

RAND ()

Remarks

Although RAND does not use arguments, you must supply the empty parentheses to correctly reference the function.

Example

This function returns a random number greater than or equal to 0 and less than 10:.

```
RAND() *10
```

RATE

Description

Returns the interest rate per period of an annuity, given a series of constant cash payments made over a regular payment period.

Syntax

RATE (*nper*, *pmt*, *pv* [, *fv*] [, *type*] [, *guess*])

Parameter	Description
r	
<i>nper</i>	The number of periods in the annuity.
<i>pmt</i>	The fixed payment made each period. Generally, <i>pmt</i> includes only principle and interest, not taxes or other fees.
<i>pv</i>	The present value of the annuity.
<i>fv</i>	The future value, or the amount the annuity will be worth. When you omit this argument, a future value of 0 is assumed.
<i>type</i>	Indicates when payments are due. Use 0 if payments are due at the end of the period or 1 if payments are due at the beginning of the period. When you omit this argument, 0 is assumed.
<i>guess</i>	Your estimate of the interest rate. If no argument is supplied, a value of 0.1 (10 percent) is assumed.

Remarks

RATE is calculated iteratively, cycling through the calculation until the result is accurate to .00001 percent. If the result cannot be found after 20 iterations, #NUM! is returned. When this occurs, supply a different value for *guess*.

Example

The following example returns the monthly interest rate of .0067; the annual interest rate (.0067 multiplied by 12) is 8 percent:

```
RATE(48, -439.43, 18000)
```

See Also

[FV](#)

[IPMT](#)

[NPER](#)

[PMT](#)

[PPMT](#)

[PV](#)

REPLACE

Description

Replaces part of a text string with another text string.

Syntax

REPLACE (*orig_text*, *start_position*, *num_chars*, *repl_text*)

Parameter	Description
<i>orig_text</i>	The original text string.
<i>start_position</i>	The character position where the replacement begins. If <i>start_position</i> is greater than the number of characters in <i>orig_text</i> , <i>repl_text</i> is appended to the end of <i>orig_text</i> . If <i>start_position</i> is less than 1, #VALUE! is returned.
<i>num_chars</i>	The number of characters to replace. If this argument is negative, #VALUE! is returned.
<i>repl_text</i>	The replacement text string.

Examples

This function returns "For the year: 1994":

```
REPLACE("For the year: 1993", 18, 1, "4")
```

See Also

[MID](#)

[SEARCH](#)

[TRIM](#)

REPT

Description

Repeats a text string the specified number of times.

Syntax

REPT (*text*, *number*)

Parameter	Description
<i>r</i>	
<i>text</i>	Any text string.
<i>number</i>	The number of times you want text to repeat. If number is 0, empty text ("") is returned.

Remarks

The result of **REPT** cannot exceed 255 characters.

Example

This function returns error-error-error-:

```
REPT("error-", 3)
```


RIGHT

Description

Returns the rightmost characters from the given text string.

Syntax

```
RIGHT ( text [, num_chars] )
```

Parameter	Description
<i>r</i>	
<i>text</i>	Any text string.
<i>num_chars</i>	The number of characters to return. The value must be greater than or equal to zero. If <i>num_chars</i> is greater than the number of characters in <i>text</i> , the entire string is returned. Omitting this argument assumes a value of 1.

Examples

This function returns r:

```
RIGHT("2nd Quarter")
```

This function returns Quarter:

```
RIGHT("2nd Quarter", 7)
```

See Also

[LEFT](#)

[MID](#)

ROUND

Description

Rounds the given number to the supplied number of decimal places.

Syntax

ROUND (*number*, *precision*)

<u>Parameter</u>	<u>Description</u>
<i>number</i>	Any value.
<i>precision</i>	The number of decimal places to which number is rounded. When a negative precision is used, the digits to the right of the decimal point are dropped and the absolute number of significant digits specified by precision are replaced with zeros. If precision is 0, number is rounded to the nearest integer.

Example

This function returns 123.46:

```
ROUND(123.456, 2)
```

This function returns 9900:

```
ROUND(9899.435, -2)
```

See Also

[CEILING](#)

[FLOOR](#)

[INT](#)

[MOD](#)

[ROUNDDOWN](#)

[ROUNDUP](#)

[TRUNC](#)

ROUNDDOWN

Description

Rounds a number down.

Syntax

ROUNDDOWN (*number*, *numberOfDigits*)

<u>Parameter</u>	<u>Description</u>
<i>number</i>	Any real number you want to round.
<i>numberOfDigits</i>	The number of decimal places to which number is rounded. When a negative precision is used, the digits to the right of the decimal point are dropped and the absolute number of significant digits specified by precision are replaced with zeros. If precision is 0, number is rounded down to the nearest integer.

Example

This function returns 31.141:

```
ROUNDDOWN(3.14159, 3)
```

This function returns 31.400:

```
ROUNDDOWN(31415.92654, -2)
```

See Also

[CEILING](#)

[FLOOR](#)

[INT](#)

[MOD](#)

[ROUND](#)

[ROUNDUP](#)

[TRUNC](#)

ROUNDUP

Description

Rounds the given number up to the supplied number of decimal places.

Syntax

ROUNDUP (*number*, *numberOfDigits*)

<u>Parameter</u>	<u>Description</u>
<i>number</i>	Any value you want to round up.
<i>numberOfDigits</i>	The number of decimal places to which number is rounded. When a negative precision is used, the digits to the right of the decimal point are dropped and the absolute number of significant digits specified by precision are replaced with zeros. If precision is 0, number is rounded up to the nearest integer.

Example

This function returns 77:

```
ROUNDUP (76.9, 0)
```

This function returns 3150:

```
ROUNDUP (31415.92654, -2)
```

See Also

[CEILING](#)

[FLOOR](#)

[INT](#)

[MOD](#)

[ROUND](#)

[ROUNDDOWN](#)

[TRUNC](#)

ROW

Description

Returns the row number of the supplied reference.

Syntax

ROW (*reference*)

<u>Parameter</u>	<u>Description</u>
<u>r</u> <i>reference</i>	A cell or range reference. Omitting this argument returns the row number of the cell in which ROW is entered.

Examples

This function returns 3:

ROW (B3)

See Also

[COLUMN](#)

[ROWS](#)

ROWS

Description

Returns the number of rows in a range reference.

Syntax

ROWS (*range*)

Paramete	Description
<u>r</u>	
<i>range</i>	A reference to a range of cells.

Examples

This function returns 5:

`ROWS (A1:D5)`

This function returns 6:

`ROWS (C30:F35)`

See Also

[COLUMNS](#)

[ROW](#)

SEARCH

Description

Locates the position of the first character of a specified text string within another text string.

Syntax

SEARCH (*search_text*, *text* [, *start_position*])

<u>Parameter</u>	<u>Description</u>
<i>search_text</i>	The text to find. To search for an asterisk or question mark, include a tilde (~) before the character. The search string can contain wildcard characters. The available wildcard characters are * (asterisk), which matches any sequence of characters, and ? (question mark), which matches any single character.
<i>text</i>	The text to be searched.
<i>start_position</i>	The character position where the search begins. If the number you specify is less than 0 or greater than the number of characters in text, #VALUE! is returned. Omitting this argument assumes a starting position of 1.

Remarks

Text is searched from left to right, starting at the position specified. The search is not case-sensitive. If text does not contain the search string, #VALUE! is returned.

Examples

This function returns 6:

```
SEARCH("?5", "Bin b45")
```

This function returns 5:

```
SEARCH("b", "Bin b45", 4)
```

See Also

[FIND](#)

[MID](#)

[REPLACE](#)

[SUBSTITUTE](#)

SECOND

Description

Returns the second that corresponds to the supplied date.

Syntax

SECOND (*serial_number*)

<u>Parameter</u>	<u>Description</u>
<i>serial_number</i>	The time as a serial number. The decimal portion of the number represents time as a fraction of the day.

Examples

This function returns 58:

```
SECOND (.259)
```

This function returns 46:

```
SECOND (34657.904)
```

See Also

[DAY](#)

[HOUR](#)

[MINUTE](#)

[MONTH](#)

[NOW](#)

[WEEKDAY](#)

SIGN

Description

Determines the sign of the specified number.

Syntax

SIGN (*number*)

<u>r</u>	<u>Descriptio</u>
<i>number</i>	Any number.

Remarks

SIGN returns 1 if the specified number is positive, -1 if it is negative, and 0 if it is 0.

Examples

This function returns -1:

```
SIGN(-123)
```

This function returns 1:

```
SIGN(123)
```

See Also

[ABS](#)

SIN

Description

Returns the sine of the supplied angle.

Syntax

SIN (*number*)

Parameter	Description
<i>r</i>	
<i>number</i>	The angle in radians. If the angle is in degrees, convert the angle to radians by multiplying the angle by PI()/180.

Examples

This function returns .85:

```
SIN(45)
```

This function returns .89:

```
SIN(90)
```

See Also

[ASIN](#)

[PI](#)

SINH

Description

Returns the hyperbolic sine of the specified number.

Syntax

SINH (*number*)

r	Descriptio
<u>number</u>	<u>n</u> Any number.

Examples

This function returns 1.18:

`SINH (1)`

This function returns 10.02:

`SINH (3)`

See Also

[ASINH](#)

[PI](#)

SLN

Description

Returns the depreciation of an asset for a specific period of time using the straight-line balance method.

Syntax

SLN (*cost, salvage, life*)

Parameter	Description
<i>r</i>	
<i>cost</i>	The initial cost of the asset.
<i>salvage</i>	The salvage value of the asset.
<i>life</i>	The number of periods of the useful life of the asset.

Example

This function returns 1285.71:

```
SLN(10000, 1000, 7)
```

See Also

[DDB](#)

[SYD](#)

[VDB](#)

SQRT

Description

Returns the square root of the specified number.

Syntax

SQRT (*number*)

Parameter	Description
<u>r</u> number	Any positive number. If you specify a negative number, the error #NUM! is returned.

Examples

This function returns 3:

`SQRT (9)`

This function returns 1.58:

`SQRT (2.5)`

See Also

[SUMSQ](#)

STDEV

Description

Returns the standard deviation of a population based on a sample of supplied values. The standard deviation of a population represents an average of deviations from the population mean within a list of values.

Syntax

STDEV (*number_list*)

<u>Parameter</u>	<u>Description</u>
<i>number_list</i>	A list of as many as 30 numbers, separated by commas. The list can contain numbers or a reference to a range that contains numbers.

Example

This function returns .56:

```
STDEV(4.0, 3.0, 3.0, 3.5, 2.5, 4.0, 3.5)
```

See Also

[STDEVP](#)

[VAR](#)

[VARP](#)

STDEVP

Description

Returns the standard deviation of a population based on an entire population of values. The standard deviation of a population represents an average of deviations from the population mean within a list of values.

Syntax

STDEVP (*number_list*)

Parameter	Description
<i>number_list</i>	A list of as many as 30 numbers, separated by commas. The list can contain numbers or a reference to a range that contains numbers.

Example

This function returns .52:

```
STDEVP(4.0, 3.0, 3.0, 3.5, 2.5, 4.0, 3.5)
```

See Also

[STDEV](#)

[VAR](#)

[VARP](#)

SUBSTITUTE

Description

Replaces a specified part of a text string with another text string.

Syntax

SUBSTITUTE (*text*, *old_text*, *new_text* [, *instance*])

Parameter	Description
<i>r</i>	
<i>text</i>	A text string that contains the text to replace. You can also specify a reference to a cell that contains text.
<i>old_text</i>	The text string to be replaced.
<i>new_text</i>	The replacement text.
<i>instance</i>	Specifies the occurrence of <i>old_text</i> to replace. If this argument is omitted, every instance of <i>old_text</i> is replaced.

Examples

This function returns "Second Quarter Results":

```
SUBSTITUTE("First Quarter Results", "First", "Second")
```

This function returns "Shipment 45, Bin 52":

```
SUBSTITUTE("Shipment 45, Bin 45", "45", "52", 2)
```

See Also

[REPLACE](#)

[TRIM](#)

SUM

Description

Returns the sum of the supplied numbers.

Syntax

SUM (*number_list*)

Parameter	Description
<i>number_list</i>	<p>A list of as many as 30 numbers, separated by commas. The list can contain numbers, logical values, text representations of numbers, or a reference to a range containing those values. Error values or text that cannot be translated into numbers return errors.</p> <p>If a range reference is included in the list, text, logical expressions, and empty cells in the range are ignored.</p>

Examples

This function returns 6000:

```
SUM(1000, 2000, 3000)
```

This function returns 4000 when each cell in the range contains 1000:

```
SUM(A10:D10)
```

See Also

[AVERAGE](#)

[COUNT](#)

[COUNTA](#)

[PRODUCT](#)

[SUMSQ](#)

SUMIF

Description

Returns the sum of the specified cells based on the given criteria.

Syntax

SUMIF (*range*, *criteria*, *sum_range*)

Parameter	Description
<i>range</i>	The range of cells you want evaluated.
<i>criteria</i>	A number, expression, or text that defines which cells are added. For example, criteria can be expressed as 15, "15", ">15", "cars".
<i>sum_range</i>	The actual cells to sum. These cells are only summed if their corresponding cells in range match the criteria. If this argument is omitted, the cells in range are summed.

See Also

[**AVERAGE**](#)

[**COUNT**](#)

[**COUNTA**](#)

[**COUNTIF**](#)

[**PRODUCT**](#)

[**SUM**](#)

SUMSQ

Description

Squares each of the supplied numbers and returns the sum of the squares.

Syntax

SUMSQ (*number_list*)

Parameter	Description
<i>number_list</i>	<p>A list of as many as 30 numbers, separated by commas.</p> <p>The list can contain numbers, logical values, text representations of numbers, or a reference to a range containing those values.</p> <p>Error values or text that cannot be translated into numbers return errors.</p> <p>If a range reference is included in the list, text, logical expressions, and empty cells in the range are ignored.</p>

Example

This function returns 302:

```
SUMSQ(9, 10, 11)
```

See Also

[SUM](#)

SYD

Description

Returns the depreciation of an asset for a specified period using the sum-of-years method. This depreciation method uses an accelerated rate, where the greatest depreciation occurs early in the useful life of the asset.

Syntax

SYD (*cost, salvage, life, per*)

Parameter	Description
<i>r</i>	
<i>cost</i>	The initial cost of the asset.
<i>salvage</i>	The salvage value of the asset.
<i>life</i>	The number of periods in the useful life of the asset.
<i>period</i>	The period for which to calculate the depreciation. The time units used to determine period and life must match.

Example

This function returns 1607.14:

```
SYD(10000, 1000, 7, 3)
```

See Also

[DDB](#)

[SLN](#)

[VDB](#)

T

Description

Tests the supplied value and returns the value if it is text.

Syntax

T (*value*)

Parameter	Description
r <i>value</i>	The value to test.

Remarks

Empty text ("") is returned for any value that is not text.

Examples

This function returns Report:

T("Report")

This function returns empty text (" ") if A4 contains a number:

T(A4)

See Also

[N](#)

[VALUE](#)

TAN

Description

Returns the tangent of the specified angle.

Syntax

TAN (*number*)

Parameter	Description
<i>r</i> <i>number</i>	The angle in radians. To convert a number expressed as degrees to radians, multiply the degrees by $\text{PI}()$ /180.

Examples

This function returns 0.752:

```
TAN (0.645)
```

This function returns 1:

```
TAN (45*PI () /180)
```

See Also

[ATAN](#)

[ATAN2](#)

[PI](#)

[TANH](#)

TANH

Description

Returns the hyperbolic tangent of a number.

Syntax

TANH (*number*)

r	Descriptio
<u><i>number</i></u>	<u>n</u> Any number.

Examples

This function returns $-.96$:

`TANH (-2)`

This function returns $.83$:

`TANH (1.2)`

See Also

[ATANH](#)

[COSH](#)

[SINH](#)

[TAN](#)

TEXT

Description

Returns the given number as text, using the specified formatting.

Syntax

TEXT (*number*, *format*)

Parameter	Description
<i>r</i>	
<i>number</i>	Any value, a formula that evaluates to a number, or a reference to a cell that contains a value.
<i>format</i>	A string representing a number format. The string can be any valid format string including "General," "M/DD/YY," or "H:MM AM/PM." The format must be surrounded by a set of double quotation marks. Asterisks cannot be included in format.

Examples

This function returns 123.620:

```
TEXT(123.62, "0.000")
```

This function returns 10/19/94:

```
TEXT(34626.2, "MM/DD/YY")
```

See Also

[DOLLAR](#)

[FIXED](#)

[T](#)

[VALUE](#)

TIME

Description

Returns a serial number for the supplied time.

Syntax

TIME (*hour, minute, second*)

Parameter	Description
<i>hour</i>	A number from 0 to 23.
<i>minute</i>	A number from 0 to 59.
<i>second</i>	A number from 0 to 59.

Examples

This function returns .52:

```
TIME(12, 26, 24)
```

This function returns .07:

```
TIME(1, 43, 34)
```

See Also

[HOUR](#)

[MINUTE](#)

[NOW](#)

[SECOND](#)

[TIMEVALUE](#)

TIMEVALUE

Description

Returns a serial number for the supplied text representation of time.

Syntax

TIMEVALUE (*text*)

Parameter	Description
<i>r</i>	
<i>text</i>	A time in text format.

Examples

This function returns .07:

```
TIMEVALUE ("1:43:43 am")
```

This function returns .59:

```
TIMEVALUE ("14:10:07")
```

See Also

[HOUR](#)

[MINUTE](#)

[NOW](#)

[SECOND](#)

[TIME](#)

TODAY

Description

Returns the current date as a serial number.

Syntax

TODAY ()

Remarks

This function is updated only when the worksheet is recalculated.

See Also

[DATE](#)

[DAY](#)

[NOW](#)

TRIM

Description

Removes all spaces from text except single spaces between words.

Syntax

TRIM (*text*)

Parameter	Description
<u>r</u> text	Any text string or a reference to a cell that contains a text string.

Remarks

Text that is imported from another environment may require this function.

Example

This function returns Level 3, Gate 45:

```
TRIM(" Level 3, Gate 45 ")
```

See Also

[CLEAN](#)

[MID](#)

[REPLACE](#)

[SUBSTITUTE](#)

TRUE

Description

Returns the logical value True. This function always requires the trailing parentheses.

Syntax

TRUE ()

See Also

[FALSE](#)

TRUNC

Description

Truncates the given number to an integer.

Syntax

TRUNC (*number* [, *precision*])

Parameter	Description
<i>r</i>	
<i>number</i>	Any value.
<i>precision</i>	The number of decimal places allowed in the truncated number. Omitting this argument assumes a precision of 0.

Remarks

TRUNC removes the fractional part of a number to the specified precision without rounding the number.

Examples

This function returns 123.45:

```
TRUNC(123.456, 2)
```

This function returns 9800:

```
TRUNC(9899.435, -2)
```

See Also

[CEILING](#)

[FLOOR](#)

[INT](#)

[MOD](#)

[ROUND](#)

TYPE

Description

Returns the argument type of the given expression.

Syntax

TYPE (*expression*)

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<u>r</u>	<i>expression</i> Any expression.
----------	-----------------------------------

Remarks

The valid values returned by this argument are:

<u>Number</u>	<u>Description</u>
---------------	--------------------

<u>r</u>	
1	Number
2	Text string
4	Logical value
16	Error value

Examples

This function returns 1 if cell A1 contains a number:

`TYPE (A1)`

This function returns 2:

`TYPE ("Customer")`

See Also

[ISBLANK](#)

[ISERR](#)

[ISERROR](#)

[ISLOGICAL](#)

[ISNA](#)

[ISNONTTEXT](#)

[ISNUMBER](#)

[ISREF](#)

[ISTEXT](#)

UPPER

Description

Changes the characters in the specified string to uppercase characters.

Syntax

UPPER (*text*)

r	Descriptio
<u>text</u>	<u>n</u> Any string.

Remarks

Numeric characters in the string are not changed.

Examples

This function returns 3RD QUARTER:

```
UPPER("3rd Quarter")
```

This function returns JOHN DOE:

```
UPPER("JOHN DOE")
```

See Also

[**LOWER**](#)

[**PROPER**](#)

VALUE

Description

Returns the specified text as a number.

Syntax

VALUE (*text*)

Parameter	Description
<i>r</i>	
<i>text</i>	Any text string, a formula that evaluates to a text string, or a cell reference that contains a text string. You can also specify a date or time in a recognizable format (for example, M/DD/YY for dates or H:MM AM/PM for time). If the format is not recognized, #VALUE! is returned.

Examples

This function returns 9800:

```
VALUE(9800)
```

This function returns 123:

```
VALUE("123")
```

See Also

[DOLLAR](#)

[FIXED](#)

[TEXT](#)

VAR

Description

Returns the variance of a population based on a sample of values.

Syntax

VAR (*number_list*)

<u>Parameter</u>	<u>Description</u>
<i>number_list</i>	A list of as many as 30 numbers, separated by commas. The list can contain numbers or a reference to a range that contains numbers.

Example

This function returns .31:

```
VAR(4.0, 3.0, 3.0, 3.5, 2.5, 4.0, 3.5)
```

See Also

[STDEV](#)

[STDEVP](#)

[VARP](#)

VARP

Description

Returns the variance of a population based on an entire population of values.

Syntax

VARP (*number_list*)

<u>Parameter</u>	<u>Description</u>
<i>number_list</i>	A list of as many as 30 numbers, separated by commas. The list can contain numbers or a reference to a range that contains numbers.

Example

This function returns .27:

```
VARP(4.0, 3.0, 3.0, 3.5, 2.5, 4.0, 3.5)
```

See Also

[STDEV](#)

[STDEVP](#)

[VAR](#)

VDB

Description

Returns the depreciation of an asset for a specified period using a variable method of depreciation.

Syntax

VDB (*cost*, *salvage*, *life*, *start_period*, *end_period* [, *factor*] [, *method*])

Parameter	Description
<i>cost</i>	The initial cost of the asset.
<i>salvage</i>	The salvage value of the asset.
<i>life</i>	The number of periods in the useful life of the asset.
<i>start_period</i>	The beginning period for which to calculate the depreciation. The time units used to determine <i>start_period</i> and <i>life</i> must match.
<i>end_period</i>	The ending period for which to calculate the depreciation. The time units used to determine <i>end_period</i> and <i>life</i> must match.
<i>factor</i>	The rate at which the balance declines. Omitting this argument assumes a default of 2, which is the double-declining balance factor.
<i>method</i>	A logical value that determines if you want to switch to straight-line depreciation when depreciation is greater than the declining balance calculation. Use True to maintain declining balance calculation; use False or omit the argument to switch to straight-line depreciation calculation.

Example

This function returns 1041.23:

```
VDB(10000, 1000, 7, 3, 4)
```

See Also

[DDB](#)

[SLN](#)

[SYD](#)

VLOOKUP

Description

Searches the first column of a table for a value and returns the contents of a cell in that table that corresponds to the location of the search value.

Syntax

VLOOKUP (*search_item*, *search_range*, *column_index*)

Parameter	Description
<i>search_item</i>	A value, text string, or reference to a cell containing a value that is matched against data in the top row of <i>search_range</i> .
<i>search_range</i>	The reference of the range (table) to be searched. The cells in the first column of <i>search_range</i> can contain numbers, text, or logical values. The contents of the first column must be in ascending order (for example, -2, -1, 0, 2...A through Z, False, True). Text searches are not case-sensitive.
<i>column_index</i>	The column in the search range from which the matching value is returned. <i>column_index</i> can be a number from 1 to the number of rows in the search range. If <i>column_index</i> is less than 1, #VALUE! is returned. When <i>column_index</i> is greater than the number of rows in the table, #REF! is returned.

Remarks

VLOOKUP compares the information in the first column of *search_range* to the supplied *search_item*. When a match is found, information located in the same row and supplied column (*column_index*) is returned.

If *search_item* cannot be found in the first column of *search_range*, the largest value that is less than *search_item* is used. When *search_item* is less than the smallest value in the first column of the *search_range*, #REF! is returned.

Examples

The following examples use this worksheet.

	A	B	C	D	E
1	Employee	Start Date	Emp. No.	Salary	Exempt
2	Anderson	10/15/84	2348	\$37,800	Y
3	Clark	2/6/90	4891	\$28,700	N
4	Davis	6/21/80	2480	\$46,950	Y
5	Franklin	4/20/88	3793	\$30,275	Y
6	Lee	8/30/89	3961	\$25,000	N
7	Olson	11/1/81	2578	\$45,780	Y
8	Turner	2/15/93	5129	\$26,100	N
9	Wilson	9/1/89	3965	\$31,650	Y

This function returns \$28,700:

```
VLOOKUP("Clark", A2:E9, 4)
```

This function returns 3961:

```
VLOOKUP("Lee", A2:E9, 3)
```

See Also

[HLOOKUP](#)

INDEX
LOOKUP
MATCH

WEEKDAY

Description

Returns the day of the week that corresponds to the supplied date.

Syntax

WEEKDAY (*serial_number*)

<u>Parameter</u>	<u>Description</u>
<i>serial_number</i>	The date as a serial number or as text (for example, 06-21-94 or 21-Jun-94).

Remarks

WEEKDAY returns a number ranging from 1 (Sunday) to 7 (Saturday).

Examples

This function returns 1, indicating Sunday:

```
WEEKDAY(34399.92)
```

This function returns 3, indicating Tuesday:

```
WEEKDAY("06/21/94")
```

See Also

[DAY](#)

[NOW](#)

[TEXT](#)

[TODAY](#)

YEAR

Description

Returns the year that corresponds to the supplied date.

Syntax

YEAR (*serial_number*)

<u>Parameter</u>	<u>Description</u>
<i>serial_number</i>	The date as a serial number or as text (for example, 06-21-94 or 21-Jun-94).

Examples

This function returns 1993:

```
YEAR(34328)
```

This function returns 1994:

```
YEAR("06/21/94")
```

See Also

[DAY](#)

[HOUR](#)

[MINUTE](#)

[MONTH](#)

[NOW](#)

[SECOND](#)

[TODAY](#)

[WEEKDAY](#)

