

Obsah

Obsah nápovědy lze zobrazit stiskem klávesy F1 nebo výběrem příslušné položky z menu Nápověda.

V nápovědi programu HE! můžete najít:

- [Syntaxe jednotlivých tagů](#)
- [Informace o programu](#)
- [Informace o autorovi](#)
- [Ceník programů](#)
- [Jak získat ostrou verzi nebo upgrade](#)
- [FAQ \(často kladené otázky\)](#)
- [Bohemia.Net \(Internet provider\)](#)

Autor

ing. David Macek
Konstantinova 1495/42, Praha 4, 14900
Tel.: 0603/43 80 34

Home page: <http://www.comports.com/dm/>
E-mail: dm@comports.com

Program

Program HE! se vytváří již třetí rok. Je to editor HTML dokumentů, který umožňuje velmi snadno a rychle vkládat většinu existujících tagů. K tomu mimo jiné přispívá přehledné menu a tři lišty s tlačítky. Ve spodní části okna je stavová řádka, která podává několik důležitých informací. Nad ní je řádka pro rychlé přepínání mezi otevřenými dokumenty. Z této řádky je rovněž patrné, které dokumenty jsou uloženy a které nikoliv. HE! podporuje všechny běžné operace, které nabízejí i ostatní editory, t.j. vyhledávání a záměna textu, práce se schránkou, operace undo, skok na zvolený řádek a samozřejmě práce se soubory. Jediným omezením je to, že HE! neumí editovat soubory větší než 64kB.

Vzhledem k tomu, že He! (HTML Editor) obsahuje velkou řadu užitečných funkcí, považuji za účelné uvést zde přehled těch nejdůležitějších. Vzhledem k rozsahu těchto specialit jsem je rozdělil do několika skupin:

Práce se soubory:

- při vytváření nového souboru lze použít šablonu, vytvoření základní kostry dokumentu nebo vytvoření kostry pro použití rámců (funkce menu/soubor/nový)
- funkce pro uložení souboru a jeho načtení v prohlížeči (klávesa F12, tlačítko na horní lišti nebo menu/soubor/uložit a otevřít v prohlížeči)
- funkce "Reload upon Save" umožňuje rychlé ladění dokumentu (pracuje jen pro Netscape a není garantována funkčnost pro všechny verze), aktivuje se klávesou Alt+F12 nebo přes menu/soubor/uložit a reload v Netscapu
- program podporuje HTML dokumenty ukládané ve formátu pro DOS i pro UNIX, typ formátu lze přepnout v menu/soubor/speciální/změna typu DOS/UNIX
- export souboru do formátu bez HTML tagů (menu/soubor/speciální/Export do TXT)
- kontrola správného využití (syntaxe) HTML tagů (menu/soubor/speciální/kontrola dokumentu)
- He! umí vyhledávat soubory dle zadané specifikace a dle šetřice, který se má v souboru nacházet, nalezené soubory lze okamžitě v editoru otevřít (menu/nástroje/hledání souborů)
- při otevírání HTML souborů podává He! doplňující informace pro snadnější výběr požadovaného dokumentu (tímto informacemi jsou datum poslední aktualizace a velikost souboru)
- funkce pro odstranění diakritiky ze souboru (využití při vytváření stránek určených nejen pro windowsovské prohlížeče), tato funkce se nachází v menu/soubor/speciální/odstranit diakritiku
- pokud pracujete s více soubory najednou, lze mezi nimi rychle přepínat pomocí lišty v dolní části hlavního okna aplikace nebo pomocí klávesy F6; posledních 5 souborů, jež byly v minulosti otevřeny, lze také rychle otevřít pomocí položek v dolní části menu soubor

Prostředí programu:

- na spodní části okna naleznete lištu, pomocí které lze rychle přepínat mezi editovacími okny; zároveň je z ní patrné, který soubor byl modifikován a který ne
- program zobrazuje on-line nápovědu ke všem tlačítkům na lištách, tato nápověda se zobrazí při umístění myši nad tlačítko; program zobrazuje i on-line nápovědu pro všechny položky menu (na spodní řádce hlavního okna aplikace)
- pokud při stisku tlačítka na lišti držíte klávesu SHIFT, použije se rozšířená syntaxe zvoleného tagu (pokud existuje) a nabídne se také příslušný on-line help; ěudlík ULOŽIT se SHIFTEM se miní na funkci uložit všechny soubory, ěudlík OTEVŘÍT SOUBOR se SHIFTEM se miní na funkci Nový soubor; podpora rozšířené syntaxe se pozná podle ěerveného křížku v pravém horním rohu tlačítka
- speciální menu editovaných oken, tato menu se vyvolávají pravým tlačítkem myši v okně (standardní okenní menu) nebo totéž s podrženou klávesou SHIFT -> speciální menu pro

rychlé vkládání nejčastěji používaných tagů

- z He! lze spustit několik systémových nástrojů (menu/nástroje/systémové nástroje)
- program má speciálně rozšířeny (prodlouženy) seznamy souborů a adresářů v oknech pro otevírání a ukládání dokumentů (efekt spočívá v menší nutnosti rolovat seznamy při hledání správného souboru)
- program barevně odlišuje tagy, komentáře a citizce od textu (pokud se Vám to nelíbí nebo je program na Vašem počítači příliš pomalý, můžete barvy vypnout v menu/nástroje/volitelné parametry); pokud se Vám barevné odlišení líbí, ale nelíbí se Vám předdefinovaná struktura barev, lze ji zmínit v menu/nástroje/barvy editoru
- při akcích, které mohou zabrat více času, se zobrazuje tzv. progress bar určující v jaké fázi se daná operace nachází (progress bar se zobrazuje na spodní řádce hlavního okna programu)
- pokud Vás při psaní textu nebaví stále kontrolovat délku řádky, můžete nastavit volbu Formátovat text (menu/nástroje/volitelné parametry), která se postará o to, aby žádný text nepřesahoval hranici okna (zalamování je tvrdé, tzn. že se vkládají standardní odřádkování)
- jestliže Vám nevyhovuje Vaše ovladače češtiny, můžete použít interní ovladače programu, který se aktivuje menu/nástroje/čeština (ve stejném menu lze také provádět záměnu kláves Z a Y)

Tabulky a seznamy:

- tvorba tabulek importem z formátu DBF (menu/entity/tabulky/import z DBF)
- při vytváření tabulky lze definovat parametry, např. tloušťka okraje, vzdálenosti buněk a další (menu/entity/tabulky/tabulka)
- tabulku lze též vytvářet vpisováním textů jednotlivých buněk do dialogu, zároveň lze definovat i parametry jednotlivých buněk (menu/entity/tabulky/interaktivní tabulka)
- pokud si přejete z již hotového textu vytvořit seznam nebo tabulku, není nic snazšího; stačí jen celý tento text označit jako blok a aktivovat funkci pro vytváření seznamu nebo tabulky - po definici potřebných parametrů je vygenerování příslušných tagů dílem okamžiku
- pokud si přejete vkládat více tabulek dohromady, je třeba uvádět u všech tagů definujících řádky a buňky tabulky i jejich koncové části

Další tagy:

- u tagů je především užitečná možnost upravovat jejich parametry, i když již byly vloženy; tuto operaci lze spustit kombinací ^E, přes hlavní nebo lokální menu a příslušné tlačítko lze nalézt na levé lišti; jedinou nutností je mít kurzor umístěný na příslušném tagu, jehož parametry chcete upravovat
- speciální vkládání tagu IMG - všechny parametry lze snadno definovat pomocí dialogu; zde lze i vybrat příslušný soubor, přičemž vložena bude jen RELATIVNÍ cesta k tomuto souboru (nikoliv cesta celá) - to má výhodu při přesouvání souborů a obrazů do jiných adresářů, ale se zachováním původní struktury (menu/entity/obrazy/obraz)
- lze speciálně definovat velikost a barvu písma a to tak, že se vybere tlačítko FONT nebo menu/entity/text/font
- lze aplikovat tag na označený text, např. "vyboldovat" text můžete tak, že ho označíte jako blok a stisknete tlačítko pro tučný text (B), stejně tak lze i sestavovat seznamy a tabulky dle již existujícího textu
- program umí zobrazit kompletní (snad) syntaxi složitějších tagů (hlavně BODY, TABLE, IMG, FRAMESET, FRAME; dále pak P, BR, HR, OL, UL, TR, TH, TD) - aktivuje se klávesou Ctrl+F1 s kurzorem na příslušném tagu
- při vkládání odkazu (reference) na jiné URL lze využít drop-down listu, který si pamatuje posledních 30 vkládaných odkazů, z nichž lze snadno vybrat
- při otevírání obrazových souborů (pro tag IMG) podává He! doplňující informace pro snadnější výběr požadovaného obrázku (jsou jimi rozměry obrázku, počet barev, velikost

souboru a případně u animovaných GIFů i počet obrázků v souboru)

- vkládání obrázků jako klikovací mapy (menu/entity/obrazy/klikovací mapa)
- možnost definovat a využívat uživatelské tagy (menu/entity/uživatelské tagy nebo tlačítko na levé liště)
- pro rychlejší přístup k tagům týkajícím se formulářů je na pravé liště speciální tlačítko
- podpora vkládání META instrukcí a [JavaScriptů](#)
- možnost rychlého přístupu k logickým a fyzickým stylům (přes tlačítko na levé liště)
- pomocí menu/entity/speciální lze vkládat HTML vyjádření některých užitečných znaků
- program umí pomocí aktivace jediné funkce vymazat celý tag i s jeho případnou koncovou částí (menu/editace/vymaz tagu pod kurzorem)

Kombinace kláves:

- ^O - Otevření dokumentu
- ^N - Nový dokument
- ^S - Uložení dokumentu
- F12 - Uložení všech dokumentů a otevření aktuálního v prohlížeči
- Alt+F12 - Uložení všech dokumentů a reload v Netscapu
- Alt+~ - Definice barvy
- ^D - Vymaz tagu pod kurzorem
- ^F - Najdi text
- F3 - Najdi další výskyt
- ^E - Oprava parametrů tagu pod kurzorem
- ^G - Skok na zvolený řádek
- F6 - Další okno
- ^Enter - tag BR
- ^P - tag P
- ^H - tag HR
- ^B - tag B
- ^I - tag I
- F1 - nápověda
- ^F1 - nápověda k tagu pod kurzorem
- Alt+1 - tag H1 (Alt+2 - tag H2 atd.)
- Alt+0 - ukončení otevřeného tagu Hn
- ^T - vymaz slova za kurzorem
- ^Y - vymaz řádku
- ^+Alt+C - přepnutí české klávesnice
- ^+Alt+X - přepnutí Z a Y
- Alt+< - levá párová část tagu
- Alt+> - pravá párová část tagu
- Alt+F4 - Konec programu

Ostatní:

- He! umí komunikovat s programy Netscape Navigator a Microsoft Internet Explorer, tato komunikace se využívá při otvírání a znovunačítání dokumentů v prohlížečích
- možnost definování barev výběrem z dialogového okna; pokud si přejete definovat barvu zadanou v textu jako RGB (např. #ffffff), umístíte na tento text kurzor a aktivujete funkci menu/editace/výběr barvy nebo stisknete Alt+~, po zavření dialogu se definice barvy sama v textu přepíše; pokud před aktivací nestojí kurzor na žádné RGB barvě, bude textové vyjádření zvolené barvy uloženo do schránky systému (opět jako text)
- zajímají-li Vás ceny ostrých verzí programů He! a MGC a ceny jejich upgradů, je k dispozici [ceník](#)
- máte-li k programu dotaz, možná najdete odpověď v často kladených otázkách ([FAQ](#))
- pokud si chcete pořídit novější sharewarovou verzi programu a jste připojeni k Internetu,

Ize využít položky Nová verze He! a Domovská stránka autora v menu Reklama (tyto položky bohužel v minulosti nefungovaly, teď již je vše v pořádku)

Nejèastiji kladené otázky (FAQ)

Jak získat ostrou verzi programu nebo upgrade?

Bližší informace najdete [zde](#).

Pøestože jsem program právi nainstaloval, pøi spuštìní mi hlásí, že doba testování programu vypršela. Jak mohu program vyzkoušet?

Tato hláška se může objevit, pokud jste již testovali starší verzi programu, nebo pokud se v adresáři Windows nebo Windows\System nachází soubory s chybným datem (napø. 12.2.1920 nebo 1.10.2060). Pokud tato data zmìníte tøeba na dnešní, problém bude odstranìn. Datum lze zmìnit buï editací (jen u textových souborù) nebo pomocí některých programù pro správu souborù (napø. XTreeGold nebo Manažer602 vs 3.0+).

Proè He! nepodporuje standardní kombinace kláves pro práci se schránkou?

Od verze 6.2 He! podporuje kromì kombinací uvedených v menu Editace i kombinace ^C, ^X a ^V.

Proè nejde pøes schránku systému kopírovat text z mého oblíbeného textového editoru do He! bez chyb v diakritice?

Tato chyba byla od verze 6.2 opravena. Rád bych se omluvil všem, kterým v souvislosti s touto chybou vznikly nijaké problémy.

Proè se na mém poèítaèi zobrazuje editovaný text nekorektnì?

Tento nedostatek je zpùsoben nekompatibilitou Vaší verze Windows s verzemi pøedchozími. V programu jsme provedli některá opatøení, která by měla tyto chyby odstranit, ale i pøesto se může stát, že právi na Vašem poèítaèi tento problém trvá. V takovém pøípadì nám prosím dejte vidìt (nejlépe mailem), abychom mohli problém vyøešit. Pokuste se v mailu co nejlépe popsat chování programu na Vašem poèítaèi.

Jak mohu snadno upravovat parametry již vložených tagù?

Opravovat parametry tagù lze od verze 6.2. Existuje hned několik možností, jak opravné dialogy vyvolat, napø. kombinací ^E, v hlavním nebo lokálním menu anebo také pomocí pøíslušného tlačítka na levé lištì.

Proè He! nepodporuje můj oblíbený tag?

V pøípadì, že He! skuteènì nezná tag, který používáte, dejte prosím [autorovi](#) vidìt a od pøíštích verzí již bude tag zaøazen. Nezbytné ale je, abyste tag popsali a sdìlili k èemu slouží, pøípadnì uvedli i jeho možné parametry. Do doby, než se tag v nijaké nové verzi objeví, můžete využívat uživatelem definované tagy, kde si lze nadefinovat jakýkoliv tag.

Jak se vytváøejí a používají uživatelské tagy?

Používat uživatelské tagy lze prostøednictvím menu Entity/Uživatelské tagy nebo prostøednictvím pøíslušného tlačítka na levé lištì. Vytváøení uživatelských tagù je pomìrnì jednoduché a je blíže popsáno pøímo v souboru He!.usr, který tyto tagy obsahuje.

Jakou funkci má klávesa Shift?

Klávesa Shift se používá pro rozšíøení funkènosti některých tlačítek na výbìrových lištách. Možnost rozšíøené funkènosti tlačítka se pozná podle malého køížku v pravém horním rohu tlačítka. U tlačítek, pomocí nichž se vkládají tagy, znamená tato rozšíøená funkènost vkládání vìtšího množství parametrù daného tagu. U ostatních tlačítek pøi stisknutí klávesy Shift mìnì svou funkènost na vytvoøení nového souboru). Aby byla orientace mezi tlačítky snazší, obsahuje program tzv. tooltips, což jsou rychlé on-line nápovìdy ke všem tlačítkùm na ploše. Pokud si pøejete aktivovat rozšíøenou funkci nijakého tlačítka na lištì, můžete to udìlat i

pomocí pravého tlačítka myši.

Jak mohu aplikovat nějaký tag na již existující text?

Pokud se například rozhodnete, že chcete určitou část textu zvýraznit pomocí tagu , označete tuto část jako blok (pomocí tažení myši se stisknutým levým tlačítkem nebo pomocí kombinace Shift+šipky) a klikněte na zvolený tag. Uvozující část tagu se vložila před označený text a zakončující za něj, přičemž označení bloku se zrušilo. Stejným způsobem lze používat i složitější tagy, např. tagy pro vytváření seznamů nebo tabulek. V tomto případě se však navíc vkládají i jednotlivé tagy oddělující položky seznamu nebo tabulky. Pokud jste tento způsob aplikace tagů zatím nepoužívali, rozhodněte si ho vyzkoušet, neboť Vám v budoucnu ušetří mnoho práce!!!

Jak lze nejlépe vyřešit problém aktualizace ceníků na WWW?

Program He! umí vytvářet HTML tabulky importem z formátu DBF. Pokud tedy budete mít Vaše ceníky v tomto formátu, převedení do HTML je již dílem okamžiku.

Proč při vkládání tagu FONT lze vložit parametr size=+7, když efekt je stejný, jako pro size=+4?

V dokumentech je standardně nastavena velikost písma na hodnotu 3. Pokud u tagu FONT určíte odchylku od této hodnoty výše zmíněným způsobem, může výsledný součet dát hodnotu větší než 7. Hodnota 7 je však maximální možná absolutní velikost písma a každé další navýšení (8,9...) prohlížeč ignoruje. Proto tedy není na první pohled rozdíl mezi hodnotami +7 a +4.

Proč v nápovědě není popis všech tagů?

V nápovědě jsou obsaženy jen ty tagy, které jsou trochu složitější. Pokud Vám zde nějaký tag chybí, oznamte to [autorovi](#).

Proč má menu Entity v pravé části lišt názvy tagů a co znamená znak ^ (stříška)?

Menu Entity má v pravé části lišt názvy tagů pro zjednodušení a zrychlení orientace v poměrně obsáhlém menu. Pokud tedy uživatel ví přesně, jaký tag hledá, může se orientovat přímo podle v menu uvedených názvů tagů.

Znak ^ (stříška) uvedený v menu zastupuje klávesu Control (Ctrl). Pokud je tedy v menu uvedena položka **Najdi ^F**, znamená to, že kombinací kláves Ctrl+F lze tuto funkci zavolat přímo bez vyhledávání v menu. Na rozdíl od klávesy Ctrl nemá klávesa Alt, která se také často používá, žádný zkratkový znak.

Jak lze nejjednodušeji vytvořit pohyblivé ikony nebo pulsující nápisy pro zpestření WWW stránek?

Nejen tyto, ale i mnohé další typy obrázků lze vytvářet pomocí programu MGC (Multi-GIF Creator). Tento program je tedy zaměřen především na vytváření animovaných GIFů pro WWW. Pomocí MGC můžete snadno a velmi rychle vytvářet a ladit animace, které hodláte na WWW vystavit. Tento způsob "rozhýbání" WWW stránek je velmi populární, využívá se pro tvorbu malých ikonů i větších prezentací a výhodou je, že ho všechny lepší prohlížeče podporují. MGC je přes svůj název český a česky komunikující program s příjemným a snadno ovladatelným uživatelským rozhraním a množstvím užitečných funkcí.

Samozřejmě je také kontextová nápověda popisující vše, bez čeho se uživatel neobejde. Sharewarovou verzi MGC získáte na <http://www.comports.com/dm> nebo přímo u [autora](#) programu.

Co je David Macek Internet Pack?

David Macek Internet Pack je balík obsahující programy He! HTML Editor pro tvorbu HTML dokumentů a MGC Multi-GIF Creator pro tvorbu animovaných GIFů pro WWW stránky. Tento balík navíc obsahuje mnoho ikon, podkladových a dalších obrázků, kterými můžete zpestřit a zdokonalit Vaše WWW stránky. Celý balík je v češtině a je určen pro Windows

3.1/3.11/95/NT. Cena balíku je 1700,-Kč, podrobnější informace najdete v [ceníku](#).

Existují u produktů He! a MGC multilicence?

Ano, ceník najdete [zde](#).

Jak získat ostrou verzi He!

Pokud máte pouze sharewarovou verzi programu a rádi byste získali verzi ostrou, zde je několik informací, které Vám v této situaci jistě pomohou:

- Ceník ostrých verzí programu najdete [zde](#)
- Ostrou verzi si můžete objednat na domovské stránce firmy David Macek, jejíž URL je <http://www.comports.com/dm> nebo <http://www.bohemia.net/macek> (zde najdete i nejnovější sharewarové verze programů He! a MGC). Můžete však také použít poštu normální nebo elektronickou (potřebné údaje najdete [zde](#)) nebo objednávku prostřednictvím telefonu **0603/438034**.
- Po objednání Vám bude program zaslán na dobírku (cena programu + poštovné) nejpozději do tří dnů (podrobnější informace o cenách viz [ceník](#)). Poštovné je 70,- Kč, neboť zásilky nelze kvůli jejich častým ztrátám posílat jako obyčejné dobírky. Firma David Macek si v souvislosti se zasíláním programů neúčtuje balné ani žádné manipulační poplatky.
- Pokud preferujete objednání prostřednictvím pošty, můžete se obrátit na [adresu autora](#) nebo na e-mailovou adresu dm@comports.com, kde uvedete nezbytné údaje. V tomto případě Vám bude program zaslán nejpozději do tří dnů od obdržení příslušného dopisu. Stejně tak i v případě objednání prostřednictvím WWW.
- Všechny problémy, které se Vám při práci s programem objeví, a všechny nápady na jeho vylepšení můžete telefonovat na uvedené telefonní číslo nebo posílat mailem. [Autor](#) se Vaše připomínky rád dozví.

Jak získat novější verzi programu (upgrade)

Upgrade můžete získat stejným způsobem, který je uveden výše. Navíc je ale třeba doplnit, že jde o upgrade a kterou verzi legálně vlastníte. Ceny upgradů jsou patrné z [ceníku](#).

POZOR!!! Pokud se rozhodnete zakoupit si programy He! (HTML editor) i MGC (program pro tvorbu animovaných GIFů), získáte slevu 700,- Kč. **David Macek Internet Pack** obsahující tyto dva programy tedy stojí pouhých **1500,-Kč** + poštovné a můžete ho získat jen u [autora](#) programů. Podrobnější informace o cenách viz [ceník](#).

Ceník

David Macek Internet Pack (He! + MGC + bonus)

1 licence	1500,-Kč
multilicence do 5 počítačů	2300,-Kč
multilicence do 10 počítačů	3000,-Kč
multilicence do 100 počítačů	4000,-Kč
multilicence neomezená	5000,-Kč

He! HTML Editor

1 licence	1200,-Kč
multilicence do 5 počítačů	2000,-Kč
multilicence do 10 počítačů	2700,-Kč
multilicence do 100 počítačů	3700,-Kč
multilicence neomezená	4700,-Kč

Ceny upgradů

Původní verze	Cena upgradu
3.x	499,- Kč
4.x	299,- Kč
5.x	199,- Kč
6.x	149,- Kč

MGC (Multi-GIF Creator)

1 licence	1000,-Kč
multilicence do 5 počítačů	1500,-Kč
multilicence do 10 počítačů	1800,-Kč
multilicence do 100 počítačů	2000,-Kč
multilicence neomezená	2500,-Kč

Ceny upgradů

Původní verze	Cena upgradu
0.x	299,- Kč
1.x	149,- Kč
2.x	149,- Kč
3.x	99,- Kč

Karel vs 2.02 pro DOS (výukový programovací jazyk)

1 licence	500,-Kč
multilicence do 10 počítačů	1500,-Kč
multilicence neomezená	2000,-Kč

Pozn.: Autor si vyhrazuje právo na mírné změny uvedených cen. V případě zájmu si můžete vyžádat aktuální ceník.

JavaScript

Od verze 5.3 podporuje He! HTML Editor také META instrukce a JavaScripty. Pokud znáte nějaký užitečný JavaScript nebo nějakou META instrukci, kterou He! nezná, dejte mi prosím vědět a od příští verze ji bude podporovat i He!.

Vkládání JavaScriptů má jednu odlišnost od vkládání ostatních tagů. Tato odlišnost spočívá v tom, že při jejich vkládání dochází ke změnám souborů na více místech naráz. Konkrétně se přidává vlastní javascriptový kód, přidávají se parametry do tagu BODY a v neposlední řadě se vkládají příslušné tagy i do míst, kde se nachází kurzor před aktivací vkládání JavaScriptu. Vzhledem k tomu, že proces je poměrně složitý, je třeba, aby byl při používání JavaScriptů dodržován určitý princip. Tento princip spočívá v tom, že kurzor by měl být před aktivací vkládání JavaScriptů umístěn na tom místě, kde se má příslušný JavaScript zohlednit, nikoliv tam, kde má být vložena funkce JavaScriptu. V některých případech se JavaScript zohledňuje globálně a nemá žádný vztah ke konkrétnímu místu v dokumentu a v takových případech je požadavek takový, aby byl kurzor před aktivací vkládání umístěn mezi ohraničujícími částmi tagu BODY, t.j. mezi <BODY> a </BODY>.

Bohemia.Net (Internet Provider)

BOHEMIA-NET je první českou privátní, veřejně přístupnou sítí s plným připojením k Internetu, která vznikla nezávisle na akademické komunitě. Provozuje ji datac s.r.o.

Licenci k provozování datových telekomunikačních služeb udílil Český telekomunikační úřad provozovateli v červenci roku 1995. V téže měsíci získal datac i registraci RIPE NCC (Evropského koordinačního centra providerů Internetu).

Společnost datac je prvním poskytovatelem Internetu v Čechách, který pro napojení sítě Bohemia.Net do Internetu využívá dva typy mezinárodního připojení:

- první linka je vedena pronajatým datovým okruhem do Karlsruhe k firmě NTG/Xlink, zakladateli největší komerční sítě v Německu (odtud spojení s evropskou páteří Ebone linkami 34Mbps a 2Mbps).

- druhá linka je satelitní spoj prostřednictvím družice Orion Atlantic do jedné z nejsilnějších komerčních sítí v USA - sítě firmy Digex (tuto síť využívají mnohé instituce v USA).

Tuzemská konektivita je zajištěna prostřednictvím uzlu sdružení NIX.CZ, jehož je datac zakládajícím členem.

Připojit do Internetu je možné jednotlivé počítače, i celé sítě. K dispozici je nameserver s nabídkou domén druhé úrovně. Můžete si zvolit subdoménu pod některou z registrovaných domén provozovatele, například 'bohemia.net', případně si vyžádat registraci domény vlastní. Adresový prostor a jeho členění na subsítě je k dispozici podle potřeby.

K dispozici je transportní protokol TCP/IP, hierarchie routovacích protokolů BGP - OSPF - RIP, komunikační protokoly SLIP, PPP, HDLC a poštovní POP3, SMTP i klasický UUCP. Terminálový přístup je řízen serverem AAA/TACACS+, samozřejmě jsou moduly BOOTP a DHCP. Neopomenutou možností je nabídka pronájmu diskové kapacity pro www stránky, případně pro gopher, ftp nebo listserver. Ke klientskému přístupu je možné využít komerčního software, jakož i bohatého výběru shareware.

Bohemia.Net je k dispozici malým i velkým firmám jakož i jednotlivcům.

Pozor! Uživatelé s účtem na Bohemia.Net mají při zakoupení programu He! nárok na slevu ve výši 20%!

Datac s.r.o., Za papírnu 5, Praha 7, Psè.: 17000
Tel.: +420 2 667 126 39 **Fax.:** +420 2 667 126 38
<http://www.bohemia.net/>



Syntaxe

Tento help obsahuje syntaxi složitějších tagů. Pokud si přejete zobrazit syntaxi k určitému tagu, vyvolejte si rejstřík, který obsahuje kompletní seznam všech tagů popsanych tímto helpem. V rejstříku můžete hledat jak příslušný tag (např. IMG nebo TABLE), tak i klíčové slovo (např. Obraz nebo Tabulka).

Cílem vytvoření tohoto seznamu bylo zpřístupnit pokud možno kompletní syntaxi těch tagů, které mají nějaké parametry. Tagy bez parametrů tedy nejsou součástí tohoto seznamu.

Pokud jsou parametry u některých tagů popsány chybně nebo nejsou kompletní, dejte mi prosím vědět. Kontakt na mě najdete [zde](#).

Jaké prostředky budou využívány pro vyjadřování syntaxe:

nn	- číslo
[]	- jedna položka ze seznamu
#ff	- definice barvy formátem RGB

Èervení zvýrazněné parametry jsou syntaxí vyžadovány nebo bez nich tag nemá smysl. Èervení je také zvýrazněn samotný tag.

Pokud si přejete znát kompletní syntaxi některého tagu, umístíte na něj kurzor a stisknete kombinaci Ctrl+F1. Pokud je tento tag uveden v nápovědi, bude zobrazeno požadované téma.

Formátem RGB lze definovat barvu. Èíslo jím vyjádřené je v rozmezí 0-ffffff (hexadecimální), přičemž toto èíslo definuje intenzitu tří složek spektra; èervené (**R**), zelené (**G**) a modré (**B**). Složky jsou v èísle zastoupeny takto:

#RRGGBB

Pokud tedy chceme definovat určitou barvu, je třeba definovat intenzitu každé ze tří složek. Takto můžeme získat více než 16 mil. různých barev. Znak # je třeba před èíslo vždy psát.

Příklad:

#ff0000	- světle èervená
#008000	- zelená
#ffffff	- bílá

Další témata:

[Kontrola užití HTML tagů](#)

Tagy popsané v nápovědi:

[A \(Reference a kotva\)](#)
[APPLET](#)
[BODY \(Tělo dokumentu\)](#)
[BR \(Zalomení řádku\)](#)
[DL \(Seznam definic\)](#)
[FONT \(Barva a velikost písma\)](#)
[FRAME a FRAMESET \(Rámce\)](#)
[HR \(Vodorovná linka\)](#)
[IMG \(Obraz\)](#)
[LI \(Položka seznamu OL nebo UL\)](#)
[MAP a AREA \(klikovací mapy\)](#)
[OL \(Èíslovaný seznam\)](#)
[P \(Odstavec\)](#)
[TABLE \(Tabulka\)](#)

UL (Nečíslovaný seznam)

Kontrola použití tagů

He! má zabudovány jednoduchou kontrolu správnosti využívání HTML tagů. Kontrolovat dokument lze spuštěním volby Soubor/Kontrola dokumentu. Tato kontrola spočívá v tom, že se hlídá, aby

- všechny tagy měly koncové části </xxxx> odpovídající počátečním částem <xxxx>
- tagy, které nemají mít koncovou část, měly skutečně jen část úvodní
- nebyly využívány nepovolené znaky <, >, "
- bylo v dokumentu používáno správné vnořování tagů (systémem zásobníku)
- nezůstal otevřen tag (chybějící uvozovky)

Naopak se nekontroluje syntaxi parametrů jednotlivých tagů a to, zda tag je platným existujícím tagem (to je způsobeno požadavkem na použitelnost této verze editoru i po rozšířeních HTML standardů).

Další témata:

Syntaxe jednotlivých tagů

Obraz - IMG

Tag IMG se využívá pro vložení obrázku. Obrázek musí být ve formátu GIF nebo JPEG (JPG). Tag IMG nemá koncový tag.

Syntaxe:

```
<IMG SRC="url_obrazku" LOWSRC="url_podkladoveho_obrazku"  
      BORDER=nn WIDTH=nn HEIGHT=nn VSPACE=nn HSPACE=nn  
      ALIGN=[ ] ALT="alternativni_text" USEMAP="#name">
```

SRC - URL obrázku (může to být je rel. cesta)

LOWSRC - URL obrázku, který se nakreslí dospod

BORDER - šířka okraje

WIDTH - šířka obrázku v bodech

HEIGHT - výška obrázku v bodech

VSPACE - prostor nad a pod obrázkem (odsazení od textu)

HSPACE - prostor po stranách obrázku (odsazení od textu)

ALIGN - zarovnání obrázku:

left - k levému okraji

right - k pravému okraji

top - na vršek

bottom - na spodek

texttop - zarovnání s vrcholem textu

middle - na střed

absmiddle - absolutní střed

absbottom - absolutní spodek

baseline - na základní linku

ALT - alternativní text (pokud se obrázky nenačítají)

USEMAP - určení názvu mapy, definující odkazy

Další témata:

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

[MAP a AREA](#)

BR - Zalomení řádku

Tag BR se používá pro oděádkování. Rychlé vložení tohoto tagu způsobí kombinace Ctrl+Enter. Tag BR nemá koncový tag.

Syntaxe:

<BR CLEAR=[]>

CLEAR - odsazení až pod obrázek na příslušné straně
left - zarovnání pod obrázek vlevo
right - zarovnání pod obrázek vpravo
all - zarovnání pod obrázky z obou stran

Další témata:

Tag P

Tag HR

Syntaxe jednotlivých tagů

Kontrola užití HTML tagů

P a DIV

Tag P se používá pro vytvoření odstavce. Rychlé vložení tohoto tagu způsobí kombinace Ctrl+P. Tag P může a nemusí mít koncový tag (viz níže).

Syntaxe:

<P ALIGN=[]>

ALIGN - zarovnání odstavce

left - zarovnání odstavce vlevo

right - zarovnání odstavce vpravo

center - centrování odstavce (raději používejte CENTER)

Tag P může mít koncový tag a nemusí. Důrazně doporučuji obě možnosti **nekombinovat**, nebo se pak můžete setkat s nepøíjemnostmi, které nelze nijak odstranit. Pokud používáte jen poèáteèní tag P, pak nemají parametry smysl. Samotný tag P se používá pro odsazení (je podobný jako BR, jen obvykle odsazuje o dva øádky).

Pokud tag P používáte vèetnì koncové èásti, lze ho na urèitý text aplikovat tak, že tento text oznaíte jako blok a poté pomocí menu, lišty nebo kombinací Ctrl+P vložíte tag P. Poèáteèní èást se vloží pøed oznaèený blok a koncová za něj (tímto způsobem lze vkládat i mnoho dalších tagù).

Tag DIV má stejné parametry jako P, liší se jen tím, že neodøádkovává. <DIV align=center> je totéž jako <CENTER>.

Další témata:

[Tag BR](#)

[Tag HR](#)

[Syntaxe jednotlivých tagù](#)

[Kontrola užití HTML tagù](#)

HR - Vodorovná čára

Tag HR se používá pro vytvoření vodorovné linky. Rychlé vložení tohoto tagu způsobí kombinace Ctrl+H. Tag HR nemá koncovou část.

Syntaxe:

<HR WIDTH=nn SIZE=nn ALIGN=[] NOSHADE>

WIDTH - šířka linky (lze definovat i v procentech)

SIZE - velikost linky (tloušťka) v bodech

ALIGN - zarovnání

left - zarovnání vlevo

right - zarovnání vpravo

center - centrování (raději používejte CENTER)

NOSHADE - linka bez stínu

Další témata:

[Tag BR](#)

[Tag P](#)

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

A - Link (reference) a kotva

Tag A se používá ve dvou různých variantách, přičemž obě potřebují koncovou část.

Syntaxe linku (reference):

```
<A HREF="url_dokumentu" TARGET="okno">
```

HREF - obsahuje URL dokumentu, na který je odkaz směřován

TARGET - definuje, do kterého okna se dokument načte
(má smysl jen ve spojení s rámcí)

Syntaxe ukotvení:

```
<A NAME="pojmenování_části_dokumentu">
```

NAME - určité části dokumentu dá pojmenování (kotvu)

Příklad využití:

..

```
<A NAME="SPORT"> <H1>SPORT</H1> </A>
```

..zde se pojednává o sportu..

..

..

```
<A HREF="#SPORT">Klikněte zde pro skok na sport</A>
```

Tento odkaz způsobí skok do aktuálního dokumentu do části, s níž je spojen pojem SPORT.

..

Tag A NAME se využívá tehdy, pokud potřebujete otevírat některý dokument a zároveň v něm automaticky nalistovat určitou jeho část. Pokud se tedy dokument, z nějž byl úryvek, jmenuje např. POKUS.HTML, pak odkaz na tento dokument s nalistováním SPORTu se vytvoří takto:

```
<A HREF="pokus.html#SPORT">Zvýrazněný text</A>
```

Další témata:

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

Font - Velikost a barva písma

Tag FONT se používá pro definici velikosti a barvy písma. Tag FONT musí mít koncovou část.

Syntaxe:

```
<FONT SIZE=nn COLOR=#ff FACE="seznam_písem">
```

SIZE - velikost písma, standardní je 3

- lze definovat jako absolutní hodnotu nebo relativně (např. +2)

COLOR - definuje barvu ve formátu RGB (čísla vyjádřená hexadecimálně)

FACE - definuje seznam písem, která mají být pro text použita (pokud žádné z nich není dostupné, použije se standardní)

Tag FONT musí obsahovat alespoň jeden ze tří uvedených parametrů, jinak nemá smysl.

Další témata:

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

Èíslovaný seznam

Tag OL se používá pro vytváøení èíslovaných seznamù. Tag OL musí mít koncovou èást. Tag OL se úzce pojí s tagem LI, který koncovou èást mít nemusí a který definuje jednotlivé položky seznamu.

Syntaxe:

<OL TYPE=[] START=nn>

TYPE - jedna z možností 1,A,a,l,i; definuje typ èíslování

START - urèuje pořadí první položky

Seznamy UL, OL i DL lze rychle vytváøet tak, že pøíslušná èást textu, ze které se má seznam vytvoøit, oznaí jako blok a poté se stiskne tlačítko na lištì odpovídající požadovanému seznamu.

Další témata:

[Tag UL](#)

[Tag DL](#)

[Syntaxe jednotlivých tagù](#)

[Kontrola užití HTML tagù](#)

Nečíslovaný seznam

Tag UL se používá pro vytváření nečíslovaných seznamů. Tag UL musí mít koncovou část. Tag UL se úzce pojí s tagem LI, který koncovou část mít nemusí a který definuje jednotlivé položky seznamu.

Syntaxe:

```
<UL TYPE=[ ]>
```

TYPE - jedna z možností disc, circle, square; určuje prefix položek

Tag LI lze v tomto seznamu použít se stejným parametrem.

Seznamy UL, OL i DL lze rychle vytvářet tak, že příslušná část textu, ze které se má seznam vytvořit, označí jako blok a poté se stiskne tlačítko na lišti odpovídající požadovanému seznamu.

Další témata:

[Tag OL](#)

[Tag DL](#)

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

Tabulka

Tag TABLE se používá pro vytváření tabulek. Tento tag musí mít koncovou část. Tag TABLE se úzce pojí s tagy TR, TH a TD, které koncovou část mít mohou a nemusí a které definují jednotlivé řádky resp. buňky tabulky.

Syntaxe:

```
<TABLE BORDER=nn WIDTH=nn BGCOLOR=#ff  
CELLSPACING=nn CELLPADDING=nn>
```

BORDER - šířka lemu tabulky

WIDTH - šířka tabulky, lze definovat i v procentech

BGCOLOR - podkladová barva tabulky (formát RGB)

CELLSPACING - mezery mezi buňkami (v bodech)

CELLPADDING - vzdálenost textu buněk od okrajů buněk (v bodech)

Tag TR definuje jeden řádek tabulky.

Syntaxe:

```
<TR ALIGN=[ ] VALIGN=[ ] BGCOLOR=#ff>
```

ALIGN - zarovnání buněk řádku (horizontální)

- jedna z možností left, right, center

VALIGN - zarovnání buněk řádku (vertikální)

- jedna z možností top, bottom, middle, baseline

BGCOLOR - podkladová barva řádku (formát RGB)

Tagy TH a TD definují buňky tabulky. TH se používá pro nadpis v tabulce a od TD se liší tím, že obsah buňky je centrován a tučný. Oba tyto tagy mají stejnou syntaxi.

Syntaxe:

```
<TD ALIGN=[ ] VALIGN=[ ] BGCOLOR=#ff  
COLSPAN=nn ROWSPAN=nn NOWRAP WIDTH=nn>
```

ALIGN a VALIGN - jako u TR

BGCOLOR - podkladová barva buňky (ve formátu RGB)

COLSPAN - kolik sloupců má buňka zabírat

ROWSPAN - kolik řádků má buňka zabírat

NOWRAP - zákaz zalamování textu v buňce

WIDTH - šířka buňky v bodech nebo procentech

Další témata:

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

BODY - Tělo dokumentu

Tag BODY se používá pro definici základních parametrů dokumentu. Tento tag by měl mít koncovou část.

Syntaxe:

```
<BODY BGCOLOR=#ff TEXT=#ff LINK=#ff ALINK=#ff  
VLINK=#ff BACKGROUND="url_obrazku"  
TOPMARGIN=nn LEFTMARGIN=nn>
```

BGCOLOR - barva pozadí (ve formátu RGB)

TEXT - barva textu (RGB)

LINK - barva odkazů (RGB)

ALINK - barva aktivovaného odkazu (RGB)

VLINK - barva již navštívených odkazů (RGB)

BACKGROUND - obrázek, kterým se vyplní pozadí dokumentu

TOPMARGIN - odsazení od horního okraje (jen MSIE)

LEFTMARGIN - odsazení od levého okraje (jen MSIE)

Pozn.: Pro odsazení od okrajů se u Netscapu používá tag LAYER. Tento tag má následující syntaxi:

```
<LAYER LEFT=nn TOP=nn WIDTH=nn>
```

LEFT - odsazení od levého okraje (v bodech)

TOP - odsazení od horního okraje (v bodech)

WIDTH - šířka stránky

Pokud tedy chcete změnit standardní velikost okrajů, doporučujeme do tagu BODY doplnit parametry TOPMARGIN a LEFTMARGIN a za něj umístit tag LAYER, který definuje okraje pro Netscape...

Další témata:

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

FRAME a FRAMESET - Definice rámců

Tag FRAMESET se používá pro definici rámců. Tento tag musí mít koncovou část.

Syntaxe:

```
<FRAMESET ROWS="viz_dále" COLS="viz_dále">
```

viz_dále - seznam definic velikostí (řádků nebo sloupců) oddělených čárkou

- položkou může být číslo, procentualní hodnota nebo znak *

- znak * znamená, že se doplní zbývající velikost (dopočítá browser)

ROWS - definuje počet a velikost řádků, v uvozovkách je seznam "viz_dále"

COLS - totéž jako ROWS, jen se jedná o sloupce

Jedním nebo více tagy FRAMESET se rozdělí plocha prohlížeče na menší části, přičemž v každé části bude otevřen jiný dokument. Ve které bude jaký dokument se určuje tagem FRAME. Tag FRAME nemá koncovou část.

Syntaxe:

```
<FRAME SRC="url_dokumentu" NAME="jméno_rámce" NORESIZE  
SCROLLING=[ ] MARGINHEIGHT=nn>
```

SRC - URL dokumentu, který bude otevřen v příslušném rámcí

NAME - jméno rámcí (používá se u odkazů mezi rámcí)

NORESIZ - zabrání změnám velikosti

SCROLLING - zda se bude okno rolovat, možné hodnoty: yes, no, auto

MARGINHEIGHT - definuje výšku okraje

Příklad:

```
<FRAMESET ROWS="55,*">
```

```
<FRAME src="head.html" SCROLLING="NO">
```

```
<FRAMESET COLS="140,*">
```

```
<FRAME src="toolbar.html" SCROLLING="NO">
```

```
<FRAME name="mainwindow" src="welcome.html">
```

```
</FRAMESET>
```

```
</FRAMESET>
```

```
<NOFRAMES>Váš browser nepodporuje rámce.
```

```
<A HREF="welcome.html">Zde</A> naleznete uvítací stránku bez rámců.</NOFRAMES>
```

Tento příklad je vzat z mé domovské stránky na <http://lada.vse.cz/~xmacd01/>. Pokud daný browser nepodporuje rámce, zobrazí tu část dokumentu, která je uvozena tagem NOFRAMES. Dokument, který definuje rámce, by neměl obsahovat tag BODY.

Další témata:

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

LI - Položka (ne)èíslovaného seznamu

Tag LI se využívá pro definici položky èíslovaného seznamu OL nebo neèíslovaného seznamu UL. Tento tag nemusí mít koncovou èást.

Další témata:

Syntaxe jednotlivých tagù

Kontrola užití HTML tagù

DL - Seznam definic

Tag DL se využívá pro vytváření seznamů definic. Tento tag musí mít koncovou část. Tag DL je úzce spjat s tagy DT a DD. DT uvozuje termín, kdežto DD uvozuje jeho vysvětlení. Tagy DT a DD nemusí mít koncové části.

Příklad:

```
<DL>
<DT>Zde je termín
  <DD>Zde se termín vysvětlí
</DL>
```

Seznam definic lze rychle vytvářet tak, že příslušná část textu se označí jako blok a na tento blok se aplikuje tag DL. Ty řádky, které začínají od první pozice, dostanou prefix DT, ty ostatní DD.

Další témata:

[Tag OL](#)

[Tag UL](#)

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

MAP a AREA - Klikovací mapy

Tag MAP vymezuje oblast, v níž se definují výseky obrázku a k nim příslušící odkazy na další URL.

Syntaxe:

```
<MAP NAME="jméno_mapy">
```

Mezi počátečním a koncovým tagem MAP se nachází tagy AREA. Každý z nich definuje jednu oblast obdélníkového, kruhového nebo polygonového tvaru.

Syntaxe:

```
<AREA HREF="url" SHAPE=[rect|circle|poly]  
      COORDS=".." ALT="Alternativní text"  
      TARGET="jméno_rámce">
```

Parametr SHAPE definuje typ výseku obrázku - obdélník definovaný levým horním a pravým dolním rohem (rect), kružnici definovanou souřadnicemi středu a poloměrem (circle) nebo polygon definovaný neomezeným počtem bodů (poly). Pokud není parametr SHAPE uveden, využije se předdefinovaná hodnota rect. Parametr COORDS specifikuje proměnné v závislosti na parametru SHAPE. Parametr HREF určuje, na jaké URL daná oblast obrázku směřuje. Parametr ALT je nepovinný, ale ve specifikaci HTML 3.2 se všeobecně doporučuje ho používat kvůli prohlížečům pracujícím pouze v textovém režimu. Posledním parametrem je TARGET, kde lze definovat, v jakém rámcí má být dokument otevřen.

Příklad:

```
<MAP NAME="pom1">  
<AREA HREF="url1.htm" SHAPE=rect COORDS="10,10,20,20">  
<AREA HREF="url2.htm" SHAPE=circle COORDS="30,10,5">  
<AREA HREF="url3.htm" SHAPE=poly COORDS="20,20, 15,30, 20,40, 25,30">  
</MAP>
```

Další témata:

[Tag IMG](#)

[Syntaxe jednotlivých tagů](#)

[Kontrola užití HTML tagů](#)

APPLET

Tag APPLET se využívá pro vložení Java appletu do WWW stránky.

Syntaxe:

```
<APPLET CODE="jméno_souboru" ALIGN=[left|right|top|middle|bottom]  
CODEBASE="adresa_s_appletem" WIDTH=nn HEIGHT=nn  
HSPACE=nn VSPACE=nn ALT="alternativní_text">
```

Tag APPLET může mít koncovou část. Parametr CODE určuje soubor, ve kterém se applet nachází (koncovka .class), ALIGN určuje zarovnání appletu, CODEBASE určuje URL adresu, kde se applet nachází, parametry WIDTH a HEIGHT definují velikost appletu na ploše (v pixelech) a parametry HSPACE a VSPACE definují odstup oblasti vymezené appletu od textu.

Další témata:

[Tag IMG](#)

[Syntaxe jednotlivých tagů](#)

JavaScripty - obsah

V této části nápovědy můžete najít informace o používání JavaScriptu v HTML dokumentech. Všechna témata jsou logicky rozdělená, takže nalezení potřebné informace by mělo být velmi snadné. V opačném případě mi prosím zavolejte své podmínky či náměty na můj mobilní telefon (tato část je v tvorbě, takže Vaše podmínky jsou vítány).

Pozn.: Program He! má v sobě již zakomponováno vkládání několika již hotových javascriptů, ale pokud chcete být originální, je třeba přistoupit k tvorbě svého vlastního nebo alespoň k úpravě těchto předdefinovaných javascriptů. K tomu by Vám měla dopomoci právě tato část nápovědy...

Základní charakteristiky

Jak se JavaScripty používají

Jak se programuje

Události a jejich ošetření

Objekty

Funkce

Proměnné

Výrazy a operátory

Konstrukce

Rezervovaná slova

JavaScripty - používání

Pro definování scriptů se používá tag SCRIPT s parametrem LANGUAGE="JavaScript". Tento tag se může nacházet před tagem BODY (kde se obvykle definují funkce) nebo za ním (ten se spouští ve chvíli, kdy browser při čtení dokumentu k němu "dojde"). Např.:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
confirm ("Tento text se objeví před načtením dokumentu v dialogovém okně...")
</SCRIPT>

</HEAD>
<BODY>
Vítáme Vás na našich stránkách...<BR>
<SCRIPT LANGUAGE="JavaScript">
document.write("Tento text se objeví v okně prohlížeče...<P>")
</SCRIPT>
Copyright &copy; 1998 David Macek
</BODY>
</HTML>
```

Uvedený dokument se zobrazí takto:

Vítáme Vás na našich stránkách...
Tento text se objeví v okně prohlížeče...

Copyright (c) 1998 David Macek

JavaScripty - programování

Programování JavaScriptu je velmi podobné programovacímu jazyku C doplněnému o používání několika předdefinovaných objektů, přičemž název objektu se od jeho metody oddílí tečkou. Na rozdíl od C není v JavaScriptech žádná funkce main(), která by se spouštěla jako první. Platí zde totiž to pravidlo, že každý kód, který není součástí nějaké funkce, se spouští ve chvíli, kdy ho browser v rámci HTML dokumentu přečte. Browser tedy sekvencí provádí všechny příkazy, které nejsou součástí žádné JavaScriptové funkce.

JavaScripty - Události a jejich ošetření

Programování Javascriptu je objektové. S tím úzce souvisí i možnost každého objektu reagovat na zprávy. Seznam existujících druhů zpráv následuje, přičemž u každého druhu je stručný popis a seznam objektů, u kterých lze zprávu využít.

onBlur - když objekt ztratí kurzor (je přemístěn jinam)

- text, textarea, selection

onClick - když se na objekt klikne myší

- button, radiobutton, checkbox, submit, reset, link

onChange - když dojde ke změně obsahu pole

- text, textarea, selection

onFocus - když objekt získá kurzor (je přemístěn odjinud)

- text, textarea, selection

onLoad - když je načten obsah stránky

- body

onMouseOver - když se myš pohne nad objektem

- link

onSelect - když je označen blok textu

- text, textarea

onSubmit - když jsou data ve formuláři potvrzena (submit)

- form

onUnload - když uživatel opouští WWW stránku

- body

Pokud tedy chceme na zprávu nějak reagovat, zařídíme to např. takto:

```
<INPUT TYPE="button" VALUE="Calculate" onClick="compute()">
```

Po kliknutí na tento knoflík se zavolá funkce compute.

JavaScripty - Základní charakteristiky

JavaScripty lze charakterizovat několika body:

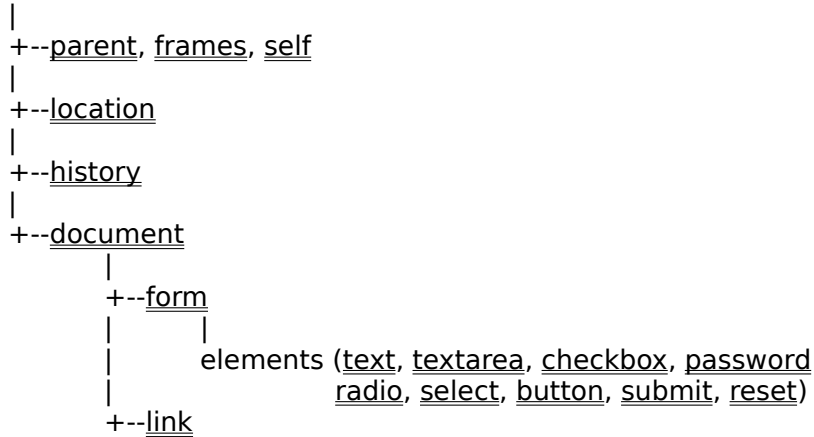
- JavaScript je interpretovaný, ne překládaný.
- Objektivně orientovaný, objekty jsou předdefinovány, nejsou ale třídy a nelze dědit.
- Kód může být integrován přímo do HTML dokumentů.
- Kontroly syntaxe se provádí při zobrazování stránky.
- Má mnoho omezení, nelze např. pracovat se soubory.

JavaScripty - Objekty

Při vyvážení JavaScriptu lze využívat několik předdefinovaných objektů, které nám jazyk nabízí. Zde je jejich seznam:

navigator

window



Math

Date

String

Každý objekt má celou řadu parametrů. Pro zobrazení seznamu proměnných nějakého objektu včetně jejich hodnot můžete použít následující kód:

```
function Promenne(obj, obj_jmeno)
{
    var vysledek = ""
    for (var i in obj)
        vysledek += obj_jmeno+"."+i+" = "+obj[i]+"<BR>"
    return vysledek;
}
```

```
document.write (Promenne (window,"window")); // zobraz parametry objektu WINDOW
document.write (Promenne (navigator,"navigator")); // zobraz parametry objektu
NAVIGATOR
```

JavaScripty - Objekt button

Objekt button reprezentuje tlačítko, na které může být "povišena" nějaká funkce.

Atributy:

name - název tlačítka

value - text na tlačítku

Metody:

click - simuluje stisknutí tlačítka

Události:

onClick

Rodiè:

form

JavaScripty - Objekt checkbox

Checkbox je zaškrťovací políčko, jež může nabývat hodnot zaškrtnuto nebo nezaškrtnuto.

Atributy:

check - určuje, zda je políčko zaškrtnuté (true/false)
defaultChecked - zda bylo políčko původně zaškrtnuté
name - název checkboxu
value - text u checkboxu

Metody:

click - simuluje kliknutí na objekt

Události:

onClick

Rodič:

form

JavaScripty - Objekt Date

Objekt Date je objekt reprezentující datum. Objekt lze použít například takto:

```
dnes = new Date(); // vytvoří novou instanci s dnešním datem  
vanoce98 = new Date(98,12,24);
```

Atributy:

žádné

Metody:

getDate - zjistí datum

getHours, getMinutes, getSeconds - zjistí hodiny/minuty/vteřiny z data/èasu

getTime - zjistí èas

getFullYear, getMonth, getDay - zjistí rok/mìsíc/den z data

setDate, setHours, setMinutes, setMonth, setSeconds, setTime, setYear

toGMTString - konverze data na øetìzec dle nastavení internetu

toLocaleString - konverze data na øetìzec dle místních nastavení

UTC - vrací poèet milisekund od 1.1.1970

Události:

žádné

Rodiè:

žádný

JavaScripty - Objekt document

Tento objekt reprezentuje dokument načtený v prohlížeči.

Atributy:

- alinkColor - odpovídá atributu ALINK z tagu BODY
- bgColor - odpovídá atributu BGCOLOR
- cookie - specifikuje cookie
- fgColor - odpovídá atributu TEXT
- forms - pole všech formulářů v dokumentu
- lastModified - datum poslední modifikace dokumentu
- linkColor - odpovídá atributu LINK
- links - pole všech odkazů
- location - kompletní URL dokumentu
- referrer - URL volajícího dokumentu
- title - titulek dokumentu
- vlinkColor - odpovídá atributu VLINK

Metody:

- clear - vyčistí obsah dokumentu
- open/close - otevření/zavření "proudu" dat
- write/writeln - zápis do dokumentu (musí být otevřen)

Události:

žádné

Rodič:

window

JavaScripty - Objekt form

Objekt form reprezentuje formulář, přičemž v rámci atributu elements jsou definovány odkazy na všechny jeho synovské objekty.

Atributy:

action - odpovídá parametru ACTION
elements - pole jednotlivých prvků formuláře
length - odpovídá počtu prvků formuláře
method - odpovídá parametru METHOD
target - odpovídá parametru TARGET

Metody:

submit - odeslání dat

Události:

onSubmit

Rodič:

document

JavaScripty - Objekt frame

Rámeček (frame) je samostatné okno, které má vlastní URL, historii a nezávislé rolování okna, přičemž v jedné chvíli může být zobrazeno více rámečků v jednom hlavním okně. Objekt frame odpovídá tagu FRAMESET nebo tagu FRAME.

Atributy:

- frames - pole rámečků
- name - odpovídá atributu NAME z definice tagu FRAME
- length - počet synovských rámečků
- parent - rodičovské okno nebo rámeček
- self - odpovídá aktuálnímu rámečku

Metody:

- setTimeout - specifikuje, za jak dlouho se má něco provést (provede se to jen JEDNOU)
- clearTimeout - zrušení nastavení timeoutu

Události:

- žádné

Rodiče:

- window
- frame

JavaScripty - Objekt history

Objekt history definuje historii navštívených stránek. Bohužel z něj ale nelze poznat, odkud se uživatel na Vaši stránku dostal.

Atributy:

length - počet již navštívených stránek

Metody:

back - skok na předchozí stránku

go - skok o určitý počet zpět nebo vpřed (dle znaménka parametru)

forward - skok vpřed

Události:

žádné

Rodiče:

document

frame

JavaScripty - Objekt link

Objekt link odpovídá tagu A HREF.

Atributy:

- hash - obsahuje parametry stránky (znaky za #)
- host - adresa serveru a port
- hostname - adresa serveru
- pathname - jen adresářová cesta
- port - jen port
- protocol - název protokolu (napø. http:)
- search - definice dotazu
- target - odpovídá atributu TARGET

Metody:

žádné

Události:

- onClick - reakce na kliknutí
- onmouseover - když je myš nad objektem
- onmouseout - když byla myš přesunuta jinam (jen Netscape)

Rodiè:

document

JavaScripty - Objekt location

Objekt location obsahuje údaje o umístění aktuálního dokumentu.

Atributy:

- host - adresa serveru a port
- hostname - adresa serveru
- pathname - jen adresářová cesta
- port - jen port
- protocol - název protokolu (např. http:)
- search - definice dotazu
- target - odpovídá atributu TARGET

Metody:

žádné

Události:

žádné

Rodič:

window

JavaScripty - Objekt Math

Tento objekt se používá pro vše, co má něco společného s matematickými výpočty.

Atributy:

- E - Eulerovo číslo (asi 2.718)
- LN2 - přirozený logaritmus dvou (asi 0.693)
- LN10 - přirozený logaritmus deseti (asi 2.302)
- LOG2E, LOG10E - logaritmus e při základu 2, resp. 10
- PI
- SQRT2 - odmocnina 2 (asi 1.414)

Metody:

- abs - absolutní hodnota
- acos, asin, atan
- ceil - vrátí nejmenší celé číslo, které je větší nebo rovno parametru
- cos, sin, tan
- exp - vrátí hodnotu e^{parametr}
- floor - vrátí největší celé číslo, které je menší nebo rovno parametru
- log - přirozený logaritmus
- max, min - vrátí větší/menší ze dvou parametrů
- round - zaokrouhlení
- sqrt - odmocnina

Události:

žádné

Rodič:

žádný

Při používání tohoto objektu lze dobře využít konstrukci with:

```
with (Math)
{
    a = PI * r * r
    y = r*sin(theta)
    x = r*cos(theta)
}
```

JavaScripty - Objekt navigator

Objekt navigator obsahuje informace o verzi prohlížeče.

Atributy:

appName - kódové označení prohlížeče

appVersion - název prohlížeče

appVersion - verze prohlížeče

Metody:

žádné

Události:

žádné

Rodič:

žádný

JavaScripty - Objekt password

Viz objekt text.

JavaScripty - Objekt radio

Objekt radio se využívá pro výběr jedné varianty z několika nabízených.

Atributy:

- checked - zda je daná volba vybraná
- defaultChecked - zda je volba implicitně vybraná
- length - počet voleb
- name - odpovídá parametru NAME v tagu INPUT
- value - odpovídá atributu VALUE (text u volby)

Metody:

- click - simulování zaškrtnutí volby

Události:

- onClick - reakce na kliknutí

Rodič:

- form

JavaScripty - Objekt reset

Objekt reset reprezentuje tlačítko, které při stisknutí způsobí opětovnou inicializaci formuláře (nastavení všech hodnot na implicitní).

Atributy:

name - jméno tlačítka (odpovídá parametru NAME)

value - text na tlačítku (parametr VALUE)

Metody:

click - simuluje stisknutí tlačítka

Události:

onClick - reakce na kliknutí

Rodiče:

form

JavaScripty - Objekt select

Objekt select reprezentuje seznam, ze kterého lze vybrat jednu nebo více položek.

Atributy:

length - počet položek (OPTIONS) v seznamu

name - název objektu (odpovídá parametru NAME)

options - seznam položek seznamu

selectedIndex - pořadí vybrané položky (nebo první vybrané položky)

Metody:

blur - ztráta kurzoru

focus - získání kurzoru

Události:

onBlur

onFocus

onChange

Rodič:

form

JavaScripty - Objekt option

Reprezentuje jednu volbu v seznamu (SELECT).

Atributy:

defaultSelected - zda je volba implicitně vybraná
index - pořadí volby
length - počet voleb v seznamu
selected - zda je volba vybraná
text - text volby

Metody:

žádné

Události:

žádné

Rodič:

select

JavaScripty - Objekt string

Objekt string reprezentuje řetězec znaků a implementuje několik funkcí pro práci s řetězcí.

Atributy:

length - délka řetězce

Metody:

big, blink, bold, fixed, fontcolor, fontsize, italics, small, strike, sub sup

- pokud bude řetězec vytisknut, bude to s odpovídajícím atributem

(všechny metody se chovají, jako by byl text ohraničen příslušným tagem)

charAt - vrátí znak na dané pozici (čísluje se od 0)

indexOf - vrátí pozici hledaného substringu

substring - vrátí část řetězce

toLowerCase toUpperCase - konverze znaků na malé/velké

Události:

žádné

Rodič:

žádný

JavaScripty - Objekt submit

Tlačítko submit se používá pro potvrzení a odeslání formuláře, objekt submit tomuto tlačítku odpovídá.

Atributy:

name - jméno tlačítka (odpovídá parametru NAME)

value - text na tlačítku (parametr VALUE)

Metody:

click - simuluje stisknutí tlačítka

Události:

onClick - reakce na kliknutí

Rodič:

form

JavaScripty - Objekt text

Objekt text reprezentuje textovou položku formuláře.

Atributy:

defaultValue - implicitní text
name - název objektu (odpovídá parametru NAME)
value - současná hodnota

Metody:

blur - ztráta kurzoru
focus - získání kurzoru
select - označení celého obsahu textu do bloku

Události:

onBlur
onFocus
onChange
onSelect

Rodič:

form

JavaScripty - Objekt textarea

Tento objekt reprezentuje textové pole. Charakteristiky objektu jsou stejné, jako u objektu text.

JavaScripty - Objekt window

Objekt window reprezentuje okno prohlížeče.

Atributy:

- defaultStatus - implicitní obsah stavové řádky
- frames - seznam rámců
- length - počet rámců v okně
- name - jméno okna
- status - obsah stavové řádky (je-li prázdný, uplatní se defaultStatus)
- top - odkaz na to okno, které je v hierarchii postaveno nejvýše

Metody:

- alert - dialog s varováním
- close - zavření daného okna
- confirm - dialog s otázkou
- open - otevření (vytvoření) okna
- prompt - umožní vstup nějaké hodnoty
- setTimeout - nastaví dobu, po které se má provést nějaká operace
- clearTimeout - zruší nastavení setTimeout

Události:

- onLoad
- onUnload

Rodič:

- žádný

JavaScripty - Zástupka parent

Objekt parent není objektem, spíše jen zástupkou za objekt v hierarchii přímo nadřazený.

JavaScripty - Zástupka self

Zástupka self se v objektech používá pro odkaz na sebe sama.

JavaScripty - Vestavěné funkce

Při programování JavaScriptu můžete využít i pár vestavěných funkcí. Jsou jimi eval, parseInt a parseFloat.

Funkce eval vyhodnotí výraz, který do funkce přichází jako parametr, a vrátí výsledek. Pø.:
Pø.:

```
int a = eval(obj.expr.value); // proměnnou a naplní příslušnou hodnotou výrazu
```

```
function setValue (myobj, myvalue)  
    eval ("document.forms[0]." + myobj + ".value") = myvalue;  
// Naplnění proměnné nějakého objektu prvního formuláře hodnotou myvalue
```

Funkce parseInt a parseFloat se používají pro konverzi řetězce na číslo celé nebo reálné. Funkce pracují tak, že prochází řetězec a hledají první znak, který nemůže být zastoupen v číselném vyjádření (vše kromě čísel, +, - a des. tečky). Konvertuje se poté vše až do tohoto znaku (pokud takový znak nalezen nebyl, konvertuje se řetězec celý).

JavaScripty - Promìnné

V JavaScriptech lze používat tyto druhy hodnot:

- èíslo (napø. 3 nebo 25.15)
- øetìzec (napø. "Ahoj")
- logická hodnota ano/ne (true/false)
- null, speciální prázdná hodnota

V JavaScriptech není třeba definovat typ promìnných, navíc každá promìnná může v průběhu provádění změnit svůj typ (napø. změna z øetìzce na èíslo nebo obrácení). V JavaScriptech se provádí automatické konverze typù, které mohou práci znaènì usnadnit:

```
x = "Výsledek je " + 10; // do x se vloží "Výsledek je 10"
```

Jména promìnných začínají písmenem nebo znakem podtržítka, ve zbytku názvu pak kromè těchto uvedených znakù mohou být i èíslice. Při psaní kódu je třeba mít na pamìti to, že promìnné a názvy funkcí se nekonvertují na velké nebo malé znaky, takže promìnná POKUS není totéž, jako promìnná pOkUs nebo jako Pokus. Tohle může dlat velké problémy zvlášť, když není třeba promìnné deklarovat.

Scripty - Výrazy a operátory

Výraz je řada konstant, proměnných a operátorů, které lze vyhodnotit a uložit do jiné proměnné. Výsledný typ může být řetězec, číslo nebo logická hodnota.

= - přiřazení hodnoty

+=, -=, *=, /=, %=, <<=, >>=, ^=, &=, |=
- přiřazení s operací (jako v C)

+, -, *, /
- standardní operátory

++, -- - přičtení nebo odečtení 1

(podmínka) ? výraz1 : výraz2

- pokud je podmínka true, vrátí se hodnota výraz1, jinak výraz2

==, >=, <=, >, <, !=
- porovnávání výrazů

JavaScripty - Konstrukce

- operace s promìnnými
- definice funkcí
- podmínky
- cykly
- konstrukce **with**
- komentáøe

Definice funkcí:

```
function _jméno_funkce_ (_parametry_funkce_)
{
  _tìlo_funkce_;
}
```

U definice funkcí platí, že všechny názvy a promìnné jsou case sensitive, tzn. že se rozlišují malá a velká písmena. Funkce VYPOCET je pak odlišná od funkce Vypocet.

Podmínky:

```
if (_podmínka_)
  { _pøíkazy;          // pokud je podmínka splněna
  }
else
  { _pøíkazy;          // pokud podmínka není splněna
  }
```

Cykly:

Používání cyklů je stejné jako v jazyce C, používá se for a while. Pro okamžité přerušování provádění cyklu lze použít příkaz break. Pro skončení právě prováděného bloku instrukcí a opětovné spuštění cyklu použijte continue.

with:

Konstrukce with má smysl u objektů, a to tehdy, když často pracujeme s atributy jednoho objektu. V takovém případě použijeme with a dále již nemusíme vždy psát jméno tohoto objektu. Pø.::

```
with window
{
  defaultStatus = "Vítejte na naší stránce";
  if (!confirm ("Chcete pokračovat?")) {close;}
}
```

Komentáøe:

Komentáø se uvozuje znaky // - vše za nimi se ignoruje. Pokud potřebujete komentáø na více řádků, použijte /* ... */.

JavaScripty - Rezervovaná slova

abstract, boolean, break, byte, case, catch, char, class, const, continue, default, do, double, else, extends, false, final, finally, float, for, function, goto, if, implements, import, in, instanceof, int, interface, long, native, new, null, package, private, protected, public, return, short, static, super, switch, synchronized, this, throw, throws, transient, true, try, var, void, while, with