

Webový prohlížeč

Webový prohlížeč je software pro prohlížení hypertextového obsahu. Mezi jeho hlavní funkce mj. patří:

- Komunikace se serverem po protokolu HTTP, tvorba požadavků, interpretace odpovědí
- Interpretace a vykreslování hypertextového obsahu (HTML, CSS a dodatečná média, jako jsou obrázky, video, audio, fonty, ...)
- Zabezpečení běhu JavaScript kódu, poskytování rozhraní pro objektovou reprezentaci HTML dokumentu (DOM) a další API (WebRTC, AJAX, WebGL, Websockets, Web Workers, Web Storage, ...)
- Možnost prohlížeč rozšířit doplňky
- Zaznamenávání historie, anonymní režim
- Uschovávání oblíbených míst ve formě záložek či seznamu oblíbených stránek

HTML, struktura internetového dokumentu

Hypertextový obsah se popisuje primárně pomocí HTML – Hypertext Markup Language. Svou syntaxí připomíná značkovací jazyk XML, avšak implementace HTML parserů jsou tolerantnější vůči chybám v syntaxi. Zápis obsahu v jazyce HTML probíhá pomocí značek (tagů), které jsou definované svým názvem (který určuje jejich primární význam) a svými atributy (vlastnostmi, chování prvků upřesňují). Značky popisují HTML prvky a ty se dělí do dvou základních kategorií:

- párové – prvek může mít další podřazené prvky
- nepárové – prvek podřazené prvky mít nemůže

Validní HTML dokument má tuto základní strukturu:

- Určení verze standardu HTML – *doctype*
- Prvek *html* obsahující:
 - hlavičku (*head*) – obsahuje metadata dokumentu (např. jeho název, připojený obsah, chtěné chování zobrazovače – viewportu, OpenGraph metadata, ...)
 - tělo (*body*) – samotný obsah dokumentu

Jednoduchý HTML 5 dokument může vypadat takto:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Můj dokument</title>
    <meta charset="UTF-8">
  </head>

  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

Mezi nejčastěji používané HTML prvky patří:

- *div*
- *span*
- nadpisy (*h1* – *h7*)
- *p*
- *strong*
- *em*
- *a*
- *img*
- *br*

CSS

CSS (Cascading Style Sheets) je jazyk blíže popisující vzhled, rozložení a chování prvků v HTML dokumentu, ke kterému je skupina CSS pravidel přidružena.

Přidružit stylová pravidla k HTML je možné třemi způsoby:

- inline styly – pravidla jsou přímo specifikována v atributu *style* daného prvku
- interně – stylopis je vložen pomocí elementu *style* v hlavičce dokumentu a tedy je jeho nedílnou součástí
- externě – stylopis je realizován dodatečným souborem a je přidružen pomocí tagu *link* v hlavičce dokumentu

Základní syntaktickou jednotkou tohoto jazyka je pravidlo – skládající se z názvu vlastnosti a její hodnoty. Pokud je pravidel uvedeno více, oddělují se středníkem. Příkladem může být:

```
display: block;
```

Nejčastěji používané vlastnosti jsou mj. tyto:

- *display*
- *color*
- *background-color*
- *font-style*
- *font-weight*
- *font-size*
- *position*
- *margin*
- *padding*

Pokud jsou styly specifikovány interně či externě, cílení pravidel se řeší pomocí tzv. selektorů, výrazů popisujících, na které prvky se mají pravidla zapsaná v sadě pravidel (rulesetu) použít. Selektor se uvádí na začátek sady pravidel:

```
(selektor) {  
    display: block;  
}
```

V selektorech lze použít následující výrazy:

- název prvku (např. *h1*) – sada pravidel bude aplikována na všechny prvky s daným názvem
- název třídy (např. *.trida*) – sada pravidel bude aplikována na všechny HTML prvky nesoucí uvedenou třídu (tedy nesou atribut *class* s uvedenou třídou – *class="trida"*), lze konjunkčně vybírat více tříd, pokud jsou uvedeny přímo za sebou (např. *.trida1.trida2*)
- ID prvku (např. *#mujprvek*) – sada pravidel bude aplikována na prvek se zadaným ID (tedy nesoucí atribut *id="mujprvek"*)
- pseudotřídy (např. *:hover*, *:visited*, *:active*, *:first-child*) – vybírají speciální stav daného prvku (např. prvek mající fokus či prvek, nad kterým je umístěný kurzor myši)
- pseudoelementy (*:before* a *:after*) – vybírají elementy existující před a resp. za obsahem vybíraného prvku
- * - vybírá všechny prvky
- *[atribut]* – vybírá prvky mající uvedený atribut
- *[atribut=hodnota]* – vybírá prvky s danou hodnotou uvedeného atributu
- *[atribut~=hodnota]*, *[atribut*=hodnota]* – vybírá prvky s daným atributem, jehož hodnota obsahuje uvedený výraz
- *[atribut|=hodnota]*, *[atribut^="hodnota"]* – vybírá prvky s daným atributem, jejichž hodnota začíná uvedeným výrazem
- *[atribut\$="hodnota"]* – vybírá prvky s daným atributem, jejichž hodnota končí uvedeným výrazem

Taktéž lze použít následující operátory:

- , - kombinuje selektory (disjunkce), pro výběr prvků budou platit oba selektory
- (mezera) – vybírá podřazený prvek nacházející se kdekoli mezi potomky nadřazeného prvku
- > - vybírá podřazený prvek mezi přímými potomky nadřazeného prvku
- + - vybírá prvek přímo následující po prvku vybíraném prvním selektorem
- ~ - vybírá prvek přímo předcházející prvek vybíraný prvním selektorem

Klient-server

Protokol HTTP, který se pro distribuci hypertextového obsahu používá, je protokol fungující nad TCP typu klient-server – existuje tedy entita poskytující obsah a vyřizující požadavky (server) a entita, která obsah konzumuje a požadavky vytváří (klient – potažmo webový prohlížeč).

Komunikace po protokolu HTTP začíná navázáním připojení ze strany klienta a odesláním požadavku serveru. Server provede naprogramovanou akci a odpoví odpovědí, která, v ideálním případě, obsahuje žádaný obsah.

JavaScript

Pro tvorbu webového obsahu se používá i jazyk JavaScript. Je to Turing-kompletní programovací jazyk a používá se pro skriptování na straně klienta - tvorba uživatelských interakcí (animace, validace formulářů, asynchronní komunikace se serverem, ...), či rovnou realizace celé prezentační logiky webové aplikace.

Dialogová okna

JavaScript použitý ve webových prohlížečích zpřístupňuje 3 typy blokujících dialogových oken:

- oznámení – implementováno funkcí *alert*, která akceptuje řetězec s textem oznámení
- potvrzení – implementováno funkcí *confirm*, která akceptuje řetězec s obsahem dialogového okna a vrací *true* či *false* v závislosti na vstupu od uživatele
- vstup – implementováno funkcí *prompt*, akceptující řetězec s obsahem dialogového okna a výchozí hodnotou, vrací zadanou či výchozí hodnotu v podobě textového řetězce

Události

Události, obecně, jsou kolekce odkazů na funkce, které jsou v nějakém bodu provádění volány zdrojem události (návrhový vzor Observer) a jsou jedním z hlavních konstrukcí využívaných při tvorbě uživatelských interakcí.

Přidávat vlastní funkce (handlers) k událostem HTML prvků lze třemi způsoby:

- uvedením úryvku kódu jako hodnotu atributu – např. `<button onClick="myfunction()">`
- přímým nastavením na získanou instanci prvku – např. `window.onload = myfunction;`
- použitím funkce `addEventListener` na získané instanci prvku – např. `window.addEventListener("load", myfunction);`, lze takto uvést více handlerů

Mezi nejčastěji využívané události mj. patří:

- *change*
- *click*
- *mouseover*
- *mouseout*
- *keydown*
- *keyup*
- *load*

Funkce

Funkce je seskupená množina příkazů, která může být parametrizovaná sadou parametrů a může vrátit nějakou hodnotu.

Funkce v JavaScriptu mají stejnou váhu jako proměnné, lze je tedy i jako proměnné předávat.

Deklarace funkce je možná třemi způsoby:

- pojmenovaná funkce – např. `function myfunction(x) { return x * 2; }`
- výrazem – např. `function(x) { return x * 2; }` (který lze uložit třeba do proměnné – `let myfunction = function(x) { return x * 2; }`)
- lambda výrazem – např. `(x) => x * 2`, vhodné pro funkce skládající se z jednoho příkazu

PHP

PHP je Turing-kompletní programovací jazyk pro skriptování na straně serveru, kde většinou bývá realizována doménová logika webové aplikace. Původním autorem je Rasmus Lerdorf, nyní je však vyvíjen komunitně. Syntakticky vychází z jazyků C++ a Java, podporuje i objektově orientované programování.

Řídící struktury

Řídící struktury jsou struktury sloužící pro větvení či opakování části psaného programu.

Podmínka:

Konstrukce podmínky (*if*) vyhodnotí uvedený výraz a podle jeho platnosti (tedy zda je možné jej vyhodnotit jako *true*) spustí patřičný příkaz či blok příkazů. Klauzule *else* není povinná.

```
if($x === 5)
{
    echo "Proměnná x se rovná číslu 5.";
}
else
{
    echo "Proměnná x se nerovná číslu 5.";
}
```

Switch:

Konstrukce switch porovnává zadaný výraz vůči zadaným konstantám a podle toho spustí patřičný blok příkazů.

```
switch($x)
{
    case 1:
        echo "Proměnná je rovna číslu 1.";
        break;

    case 2:
        echo "Proměnná je rovna číslu 1.";
        break;

    default:
        echo "Proměnná nabývá jiné hodnoty.";
        break;
}
```

Cyklus while:

Tento cyklus je ze všech cyklů v jazyce PHP nejobecnější, příkaz či blok příkazů bude opakovat tehdy, dokud zadaný podmínkový výraz bude možné vyhodnotit jako *true*.

```
while(podmínka)
{
    echo "Podmínka platí.";
}
```

Existuje varianta s vyhodnocováním podmínkového výrazu na konci – *do-while*, kde se první iterace vykoná vždy:

```
do
{
    echo "Podmínka platí.";
}
while(podminka)
```

Cyklus for:

Tato konstrukce slouží pro implementaci cyklů se známým počtem opakování. Definice cyklu se skládá z deklarace iterační proměnné, podmínkou pro pokračování cyklu a aktualizacím výrazem.

```
for($i = 0; $i < 5; $i++)
{
    echo "Cyklus probíhá.";
}
```

Cyklus foreach:

Tento cyklus se prochází na procházení iterovatelných výrazů (pole, implementace rozhraní Iterator, ...). Lze takto procházet jak hodnotu, tak volitelně její klíč.

```
foreach(iterovatelny_vyraz as $key => $value)
{
    echo $key. " => " . $value;
}
```

Formuláře

Jazyk PHP je schopen nativně zpracovávat formulářová data odeslaná jako součást URL požadavku, s typem těla *multipart/form-data* či *application/x-www-form-urlencoded*.

Data uvedená jako součást URL jsou dostupná v superglobální proměnné `$_GET`, nesoucí asociativní pole, následující URL:

```
http://example.com/myhandler.php?name1=value1&name2=value2
```

...způsobí, že proměnná `$_GET` bude nabývat této hodnoty:

```
[ "name1" => "value1", "name2" => "value2" ]
```

V případě, že jsou formulářová data uvedena jako tělo, budou textové hodnoty obsaženy v superglobální proměnné `$_POST` a to ve stejném formátu, jako výše. Pokud jsou součástí formuláře soubory, jejich metadata budou uvedena v superglobální proměnné `$_FILES`, které lze později zpracovat jejich otevřením, nebo přesunutím funkcí `move_uploaded_file`.