

Interreg



EVROPSKÁ UNIE

Rakousko-Česká republika

Evropský fond pro regionální rozvoj



INFORMATIKA

Webové technologie



UNIVERSITY
OF APPLIED SCIENCES
UPPER AUSTRIA



EVROPSKÁ UNIE

Obsah

1. HTML.....	2
2. CSS – Kaskádové styly.....	4
3. CSS II.....	7
4. JavaScript.....	11
5. JavaScript – pokračování	14
6. XML a Json.....	17
7. Serverové části webových technologií.....	21
8. Serverové části webových technologií II.....	24
9. PHP: Hypertext Preprocessor.....	26
10. PHP II – Syntaxe	29
11. Webové aplikace spolupracující s databázemi	31
12. DOM – Document Object Model.....	34

1. HTML

HTML neboli **HyperText Markup Language** je jazyk pro tvorbu webových stránek – aktuální verze – HTML5 (HTML 5.2). Tento jazyk je standardizován organizací W3C.

Skládá se z tagů (značek) a atributů (vlastností).

Zobrazení HTML probíhá pomocí prohlížeče a to v několika krocích (=parsování).

- Prohlížeč načte dokument a provede syntaktickou analýzu podle DTD
- Ke každému prvku je přiřazen styl (způsob zobrazení)
- Aplikace předepsaného kódu skriptovacích jazyků
- Postupné vykreslení stránky

HTML značky dělíme podle významu na:

- Strukturální, které určují strukturu dokumentu. Např.: Odstavec (<p>), nadpisy (<h1>, <h2>...)
- Popisné (sémantické), popisující povahu obsahu prvku, např.: Nadpis (<title>), adresa (<address>)...
- Stylistické, které určují vzhled prvku při zobrazení, např.: Tučné písmo (), kurzíva (<i>), zvýraznění ()...

Každý HTML dokument by měl dodržovat **základní schéma**:

Na začátku je DTD direktiva, deklarace typu dokumentu:

```
<! DOCTYPE html >
```

Následuje kořenový element:

```
< html > </ html >
```

A hlavička dokumentu (Obsahuje meta data):

```
<head>  
<meta charset = "utf-8"> - Kodierung  
<title > Seitenüberschrift </ title >
```

Autor, popis, klíčová slova, kaskádové styly...

```
< head >
```

Za hlavičkou následuje tělo dokumentu, které zahrnuje obsah stránky jako text, obrázky, odkazy, tabulky...

```
<body> ... </body>
```

Pro psaní HTML je výhodné použít textový editor, který zvládá barevnou syntaxi (barevně rozlišuje jednotlivé části kódu – značky, vlastnosti, prostý text), dokáže napovídat značky, zná chytré tabulátory nebo zvládá validovat dokument podle předepsané specifikace.

Například: Notepad++, PSPad...

Případně lze použít WYSIWYG (What You See Is What You Get) editory, které pracují přímo s hotovou stránkou. Uživatel tak nemusí znát jazyk HTML –poskládá stránku a editor vygeneruje příslušný kód.

Například: Adobe Dreamweaver, Microsoft Expression Web, LibreOffice Writer/Web, Bluegrifon.

2. CSS – KASKÁDOVÉ STYLY

CSS je jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v jazycích HTML, XHTML nebo XML vyvinutý a standardizovaný organizací W3C. Aktuální verze je CSS 3.

CSS lze připojit do HTML kódu několika způsoby:

- V kódu html stránky pomocí elementu `<style> </style>`

```
<style = "text/css ">
# head {
  width : 200px;
  height : 450px;
}
</style>
```

- Pomocí externího souboru - element `<link>`

```
<head>
<link rel = 'style sheet' href = 'styles.css' type
= 'text/css ' >
</head>
```

- Přímý inline zápis stylu pomocí atributu `style`

```
<p style = " color : red ; text- decoration : underline ">
Tento odstavec bude červený a podtržený.</ p>
```

Jazyk PHP se skládá z pravidel, kde každé pravidlo obsahuje selektor a blok deklarací. Každý blok deklarací pak obsahuje jednotlivé deklarace oddělené středníkem. Každá deklarace se skládá z identifikátoru vlastnosti, dvojtečky a hodnoty. Může ještě následovat označení `!important`, které zvyšuje sílu deklarace.

Proč používat CSS místo formátování pomocí HTML?

CSS nabízí:

- větší možnosti formátování, které nenabízí HTML- např.: určení vzdálenosti elementů od okrajů stránky.
- Jednodušší provádění úprav vzhledu - změna barvy všech nadpisů najednou, změna fontu,...
- Možnost vytvoření šablony pro několik stránek najednou
- Oddělení struktury a stylu - v HTML obsah, v CSS vzhled
- Cachování stylů - rychlejší načítání stránek (ale navíc jeden HTTP požadavek na načtení externího souboru)
- Dynamické změny CSS vlastností pomocí Javascriptu
- Pomocí CSS lze formátovat jakýkoli jazyk XML
- Nastavení zobrazení pro jednotlivá zařízení - podmíněné CSS

Koncový uživatel si může napsat svůj vlastní CSS styl pro libovolnou stránku - lze nastavit, aby všechny odkazy na všech webech byly vždy podtržené nebo aby na tomto konkrétním webu mělo písmo vždy černou barvu.

V kombinaci s Javascriptem mohou vzniknout účinné bookmarklety, které mohou různě vylepšovat vzhled stránky. Například odstranit všechny obrázky na pozadí, změnit pozadí na bílé a písmo na černé apod.

S CSS se také pojí jisté nevýhody. Tou hlavní je, že ne vždy dostatečná podpora v majoritních prohlížečích, případně chyby v implementaci CSS v prohlížečích. Toto lze obejít použitím různých stylů pro různé prohlížeče.

Lze použít podmíněné komentáře pro nastavení různých prohlížečů

```
<!--[if IE]> <style type="text/css"> # alert {color: blue;}  
</style> <![endif]-->
```

Tento kód bude interpretován pouze Internet Explorerem, ostatní prohlížeče uvidí obyčejný HTML komentář a interní stylpis tak nebudou interpretovat.

I CSS má své určité limity.

- CSS selektory neposkytují přístup k rodičovským prvkům
 - Nemůžete například nastylovat jen ty odstavce, které obsahují odkaz.
- Horizontální kontrola prvků na stránce je intuitivní a jednoduchá, naopak vertikální stylování vyžaduje komplexnější přístup (např.: flexbox nebo grid).
- CSS neposkytuje možnost pro symbolický zápis proměnné nebo konstanty
- Všechny hodnoty musí být vepsány přímo v kódu.
 - Například pokud se na vícero místech používá stejná barva, nemůže se použít symbolický zápis `barva=red`; a poté už jen psát proměnnou `barva`, všude se musí uvádět přímo hodnota `red`. Toto omezení odstraňují CSS preprocesory (např. SASS, LESS, Stylus).
- CSS2 nenabízí žádnou možnost pro tvorbu kulatých rámečků nebo jiných kulatých objektů. Pracuje pouze s obdélníky.
- CSS2 nenabízí žádnou možnost, jak jednomu elementu přiřadit více obrázků na pozadí.

3. CSS II

Je potřeba si ujasnit, co která verze CSS nabízí a definuje.

CSS verze 1 zavádí syntaxi používanou i v následujících verzích, přináší způsob vybírání prvků přes selektory, několik pseudotříd a definuje hodnoty a jejich jednotky.

Kategorie, pro které CSS 1 definuje hodnoty a jednotky:

- vlastnosti písma
- barvy textu a pozadí
- vlastnosti textu
- vlastnosti blokových elementů
- způsoby zobrazení prvků
- řízení pozice

Hodnoty a jednotky:

u desetinných hodnot se používá desetinná tečka

Jednotky délky

- (žádné) – u bezrozměrných vlastností (např. line-height)
- % – procenta, jednotka relativní vůči implicitnímu rozměru, zapisována bez mezery
- pt – typografický bod, výchozí jednotka je 1/72 palce
- px – 0,75 pt
- pc – pica, 1 pc = 12 pt
- cm – centimetr
- mm – milimetr
- in – palec
- em – čtverčik, je rovna základní výšce písma
- ex – výška písmene "x", je relativní k použitému písmu

Barvy v CSS se zadávají pomocí RGB palety a to:

- buď číselně #rrggbb – dvouciferná hexadecimální hodnota (00 až ff) – 16 milionů barev
- #rgb – hexadecimálně jednociferně (0 až f) - 4 096 barev.
- jako bezpečné barvy, což je ještě užší podmnožina barev: Jde o jednociferné barvy, jejichž hodnoty vzrůstají o 3 - kombinace šesti hodnot {0, 3, 6, 9, C, F} = 216 barev.
- rgb(r,g,b) - v desítkové soustavě (0 až 255) a syntaxí jako funkce – 16 milionů barev
- rgb(r%,g%,b%) – hodnoty v rozsahu (0 až 100)

Případně lze použít předdefinované konstanty a textové názvy pojmenovaných barev. Např. aqua (jasná modrozelená), black (černá) = #000, blue (modrá), fuchsia (anilínová červená), gray (šedivá), green (zelená), lime (citrónově zelená), maroon (kaštanová), navy (námořnická modř), olive (olivová), purple (purpurová), red (červená), silver (stříbrná), teal (tmavá modrozelená), white (bílá) = #FFF, yellow (žlutá). V CSS verze 4.0 je pojmenováno 148 barev.

Písmo v CSS 1 má své atributy.

- font-style: normal, italic (kurzíva), oblique (šikmá antikva)
- font-size: medium; menší velikosti: xx-small, x-small, small; větší velikosti: large, x-large, xx-large, smaller, larger, velikost v procentech 100% je standardní velikost
- font-weight: tučnost: normal, bold, bolder, thicker, lze zadat číslem: 100, 200 ... 900 (400=normal, 700=bold)
- font-variant: small-caps (kapitálky), normal
- font-decoration: underline (podtržené), overline (nadtržené), blink (blikající), line-through (přeškrtnuté), none (standardní)

CSS 1 také zavádí obecné druhy písma

- serif – klasické patkové písmo (např. Times New Roman)
- sans-serif – bezpatkové písmo (např. Helvetica nebo Arial)
- cursive – kurzíva
- fantasy – ozdobné písmo
- monospace – neproporcionální písmo (např. Courier)

Dále CSS 1 definuje prvek URL tak, že:

- konstrukce url(), kde se mezi závorky specifikuje adresa zdroje
- Absolutní: url(http://www.example.com/images/logo.png)
- Relativní vůči serveru: url(/images/logo.jpg)
- Relativní vůči aktuálnímu adresáři: url(images/logo.jpg)

Obsahuje-li URL čárky, mezery, uvozovky nebo konec kulaté závorky, tyto znaky se dají eskapovat pomocí zpětného lomítka.

Dále pak Selektory:

- Typový selektor: A - Všechny elementy typu A
- Selektor třídy: A.třída - Všechny elementy A s atributem class=„třída“
- ID selektor: A#ID - Všechny elementy s identifikátorem ID
- Selektor následovníka: A B - Všechny elementy B uvnitř A

Definice CCS 2 přinesla další definice:

- outline – vnější ohraničení
- max-height, max-width, min-height, min-width – minimální a maximální šířka nebo výška elementu
- content – nastavitelný obsah elementu
- counter – nástroj pro číslování kapitol
- quotes – styl citací
- clip – ořezávání
- cursor – kurzor nad elementem
- position – možnost pozicovat element v řádku, v bloku, absolutně, relativně, ...
- top, bottom, right, left – okrajové hodnoty pro position:absolute;
- overflow – zobrazení při přetékání obsahu
- visibility – viditelnost elementů
- z-index – možnosti překrývání
- page-break, orphans, widows – typografická pravidla pro dělení stránek
- table-layout, border-collapse, border-spacing, caption-side, empty-cells – nové možnosti pro zobrazení tabulek
- direction – směr psaní textu

CSS 2 umožňuje užívání nových typů selektorů:

- Univerzální: * - Vztahuje se na úplně všechny elementy
- Selektor dítěte: A>B - bere v potaz pouze ty elementy B, které jsou vnořeny do A přímo
- Selektor sourozenců: A+B - vybere všechny elementy B, které mají stejného rodiče jako A a které po něm zároveň přímo následují

Definuje také selektory pomocí atributů:

- A[attr] - všechny elementy A, které mají nastavený atribut attr
- A[attr=hodnota] - všechny elementy A, které mají nastaveny atribut attr="hodnota"
- A[attr~="hodnota"] - všechny elementy A s atributem attr, jehož hodnotu tvoří seznam slov oddělených mezerou a právě jedno z těchto slov se shoduje s "hodnota"
- A[attr|=hodnota] - všechny elementy A, jejichž hodnota atributu attr začíná řetězcem "hodnota", pak následuje pomlčka a další řetězec

Dále pak zavádí systém pseudoelementů a pseudotříd.

- Pseudoelementy
 - První řádek - :first-line
 - První písmeno (iniciála) - :first-letter
 - Před - :before

- Za - :after
- Jazyková pseudotřída
 - :lang
- Rodičovská pseudotřída
 - První potomek - :first-child
- Pseudotřída odkazů
 - Nenavštívený odkaz - :link
 - Navštívený odkaz - :visited
 - Zaměřený odkaz - :focus
 - Odkaz pod myší - :hover
 - Aktivní odkaz - :active

Aby vše správně fungovalo, je u odkazů potřeba dodržet pořadí. Každá pseudotřída má jinou prioritu.

A konečně poslední verze CSS3, která se pojí se standardem HTML5, jež je má plně využívat. Nabízí tak:

- animace – CSS3 přímo podporuje animaci elementů (jejich vlastností), doposud se animace dělaly přes DHTML, např. jQuery
- dodatečné možnosti stylování pozadí blokových elementů, včetně např. oříznutí jejich pozadí, vržených stínů nebo zakulacených okrajů
- dodatečná pravidla pro přetékání obsahu blokových elementů
- opacita – míra neprůhlednosti prvků
- další podpora stránkovaného obsahu – záložky a možnosti dělení textu
- flexibilní blokové elementy
- cílové odkazy – jak a kde se mají otevírat
- vlastnosti pro drag'n'drop
- dodatečné vlastnosti pro písma – načítání písma z externího zdroje, přizpůsobení velikosti při nízké čitelnosti, zužování/rozšiřování písma
- vlastnosti pro generovaný obsah – zkracování obsahu s možností expanze, přesunování elementů dále ve stránce
- mřížky (zatím nikde neimplementované)
- nové vlastnosti pro marquee
- automatický vícesloupcový layout
- nové vlastnosti Ruby
- vlastnosti pro čtený text
- 2D a 3D transformace
- vlastnosti spolupracující s navigací
- uživatelsky definované vlastnosti

4. JAVASCRIPT

Javascript je programovací jazyk používaný na webových stránkách, zapisuje se přímo do HTML kódu a patří mezi klientské skripty (vykonávají se na straně klienta)

Javascript je:

- interpretovaný – nemusí se kompilovat
- objektový – využívá objektů prohlížeče a zabudovaných objektů
- závislý na prohlížeči – funguje ale ve většině prohlížečů
- case sensitivní – záleží na velikosti písem v zápisu
- syntaxí podobný jazykům C, Java a podobným

Z podstaty Javascriptu vyplývají jistá omezení:

- Funguje pouze v prohlížeči.
- Uživatel může JavaScript zakázat
- Existují různé odlišné verze jazyka i prohlížečů, což vede k častým chybám.
- Neumí přistupovat k souborům (kromě cookies) ani k žádným systémovým objektům.
- Neumí žádná data uložit (kromě cookies).

Přesto je Javascript hojně využíván a lze jej mnohdy použít jako vestavěný skriptovací jazyk u mnohých aplikací.

Lze jej nalézt:

- Většina rozšíření pro webové prohlížeče
- Některé NoSQL datábáze jako je MongoDB nebo CouchDB akceptují dotazy napsané v JavaScriptu.
- Adobe – Acrobat and Adobe Reader, nástroje v Adobe Creative Suite (Photoshop, Illustrator, Dreamweaver a InDesign)
- Kancelářský balík aplikací OpenOffice umožňuje JavaScript používat jako skriptovací jazyk.
- Interaktivní zpracování signálu hudebního software Max/MSP
- Digitální software Apple Logic Pro X audio workstation umožňuje vytvořit vlastní MIDI efekty pluginy pomocí JavaScriptu.
- ECMAScript (JavaScript) byl zahrnut v normě VRML97 pro skriptování uzlů souborů VRML.
- Herní engine Unity 3D podporuje upravenou verzi JavaScriptu pro skriptování pomocí Mono.
- DX Studio (3D engine) používá implementaci JavaScriptu SpiderMonkey pro hry a simulace logiky.

- Maxwell Render poskytuje skriptovací engine ECMA standardu pro automatizaci úkolů.
- Google Apps Script v tabulkách Google a Google Sites umožňuje uživatelům vytvářet vlastní vzorce, automatizovat opakující se úlohy a také komunikovat s ostatními produkty Google jako je Gmail.
- Produkty SpinetiX používají SpiderMonkey JavaScript pro scriptování v SVG souborech.

JavaScript lze také použít jako skriptovací engine:

- Technologie Active Scripting od Microsoftu
- Programovací jazyk Java v 6. verzi představil balíček javax.script
- Nástroj Qt C++ obsahuje modul QtScript, který interpretuje **JavaScript** stejně jako Java balíček javax.script.

JavaScript lze do HTML zapisovat několika způsoby podobně jako CSS.

Lze využít tag `<script>` a zapsat tak skript přímo do proudu HTML.

```
<script>
  alert('Hlavu vzhůru, bude hůř!');
</script>
```

Nebo připojit externí soubor se skriptem.

```
<script src="externi_skript.js"></script>
```

Další variantou je in-line zápis

```
<p><a href="#" onClick="alert('Hello');">Click Me</a></p>
```

Nejčastěji se však využívá kombinovaného zápisu. Externím skriptem jsou definovány funkce, normálním zápisem (pomocí `<script>`) jsou inicializovány proměnné a startovní funkce a in-line skripty volají funkce podle událostí v závislosti na reakcích uživatele.

Javascript se na webových stránkách používá především pro zavádění stránky – odlišení prohlížečů, vkládání menu ze souboru, deklarace funkcí, `document.write()`, případně jako reakce na uživatelskou událost např. přejetí určitého elementu myší, kliknutí, změna velikosti okna, vyplnění formuláře...

5. JAVASCRIPT – POKRAČOVÁNÍ

JavaScript patří mezi objektově orientované programovací jazyky. Podporuje tedy klasický objektový model.

- objekt.metoda() – volání metody (funkce), příkaz, který něco dělá
- objekt.vlastnost – odkazuje na vlastnost daného objektu, má hodnotu, ale nic nedělá
- objekt.podobjekt – odkaz na vnořený objekt

JavaScript má přístup

- K objektům okna prohlížeče (třída Window)
- K prvkům stránky
- K objektům Math a Date, string
- K vytvořeným objektům

Třída Window (objekt Window) je vrcholem hierarchie objektů (tříd). Jeho podobjekty jsou:

- location – adresa načteného dokumentu
- history – historie prohlížení stránek
- navigator – informace o verzi a typu prohlížeče
- screen – vlastnosti obrazovky (šířka, výška, barvy)
- frames – práce s rámy (frame, frameset)
- event – události myši, klávesnice
- document – obrázky, formuláře, odkazy, barvy, jednotlivé HTML elementy...

Nejvyužívanější třídou je třída document, která umožňuje manipulaci s HTML (i jiným typem) dokumentem.

Její nejdůležitější metody jsou:

- document.images
- document.forms
- document.applets
- document.links
- document.anchors
- document.all
- document.frames
- document.styleSheets
- document.scripts
- document.selection
- document.getElementById()
- document.getElementsByTagName()

Třída Math slouží pro použití vyšší matematiky.

Obsahuje metody:

- `abs(x)`
- `exp(x)`
- `log(x)`
- `max(x,y)`
- `min(x,y)`
- `pow(a,x)`
- `random()`
- `sqrt(x)`
- `ceil(x)`
- `floor(x)`
- `round(x)`
- `acos(x)`
- `asin(x)`
- `atan(x)`
- `atan2(x,y)`
- `cos(x)`
- `sin(x)`
- `tan(x)`
- `Math.E`
- `Math.LN10`
- `Math.LN2`
- `Math.LOG10E`
- `Math.LOG2E`
- `Math.PI`
- `Math.SQRT2`
- `Math.SQRT1_2` – odmocnina z 1/2

Třída Date slouží pro práci s datem a časem (vytváření různých odpočítávání, kalendářů...)

Hlavní metody třídy Date:

- `get/setFullYear()`
- `get/setMonth()`
- `get/setDate()`
- `get/setDay()`
- `get/setHours()`
- `get/setMinutes()`
- `get/setSeconds()`
- `get/setMilliseconds()`
- `get/setTime()`

Třída `String` pracuje s textovými řetězci. Na textové řetězce lze aplikovat vlastnost `length`, která vrací délku řetězce.

Hlavní metody pro práci s řetězci jsou:

- `toUpperCase()`
- `toLowerCase()`
- `toString()`
- `charAt(n)`
- `charCodeAt(n)`
- `substring(a, b)`
- `substr(a, b)`
- `concat(řetězec1, řetězec2, řetězecN)`
- `fromCharCode(kód1, kód2, ..., kódN)`
- `indexOf(podřetězec)`
- `lastIndexOf(podřetězec)`
- `split(oddělovač)`

6. XML A JSON

XML neboli eXtensible Markup Language je značkovací jazyk standardizovaný konsorciem W3C. XML lze označit za standardní formát pro výměnu informací s mezinárodní podporou a vysokým informačním obsahem. Lze jej snadno převést do jiných formátů, existuje automatická kontrola struktury dokumentu a podporuje hypertext a odkazy.

Efektivita XML je silně závislá na struktuře, při špatně navržené struktuře je XML dokument nečitelný a neefektivní.

Každý XML dokument

- Musí mít právě jeden kořenový (root) element.
- Neprázdné elementy musí být ohraničeny startovací a ukončovací značkou. Prázdné elementy mohou být označeny tagem „prázdný element“.
- Všechny hodnoty atributů musí být uzavřeny v uvozovkách – jednoduchých (') nebo dvojitých ("). Opačný pár uvozovek může být použit uvnitř hodnot.
- Elementy mohou být vnořeny, ale nemohou se překrývat; to znamená, že každý (ne kořenový) element musí být celý obsažen v jiném elementu.

Příklad XML

<? xml version = "1.0" encoding = "UTF-8"?>	XML declaration
< directory >	Root element
< person >	Nested element 1
< name > Adam < / name >	Nested Element 1.1
< phone > 777 777 777 < / phone >	Nested element 1.2
< email > adam@adam.com < / email >	Nested element 1.3
< / person >	Exiting the nested element 1
< person >	Nested element 2
< name > Bara < / name >	Nested element 2.1
< phone > 666 999 666 < / phone >	Nested element 2.2
< email > bara@bara.com < / email >	Nested Element 2.3
< / person >	Exiting nested element 2
< / directory >	Exiting the root element

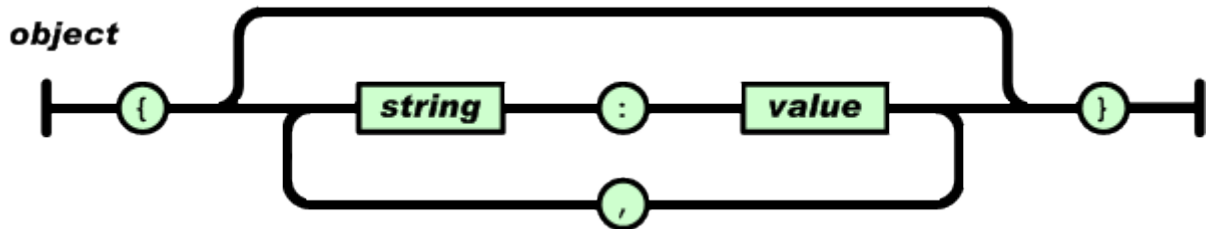
Json - JavaScript Object Notation – je odlehčený formát pro výměnu dat, jednoduše čitelný i zapisovatelný člověkem, zapsaný v textovém formátu a zcela nezávislý na jazyce.

Je založený na dvou strukturách

- Kolekce párů – název/hodnota
 - Realizace: Object, Record, Struct, Dictionary, Hash table, Keyed List, Associative array
- Seřazený seznam hodnot
 - Realizace: array, vector, list, sequence

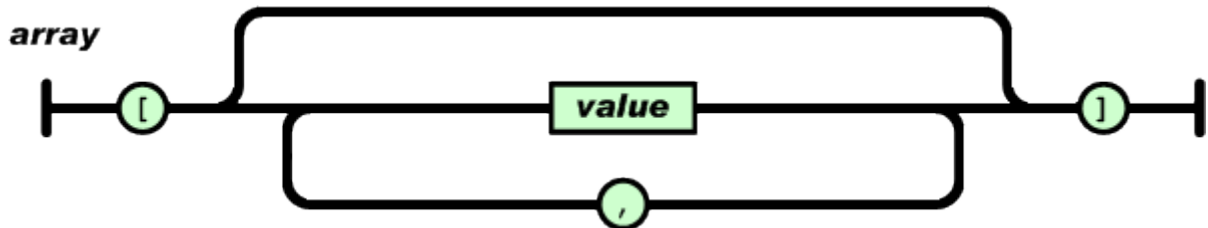
Objekt – neuspořádaná množina párů Název/Hodnota

Zápis: {název1: hodnota1, název2: hodnota2}



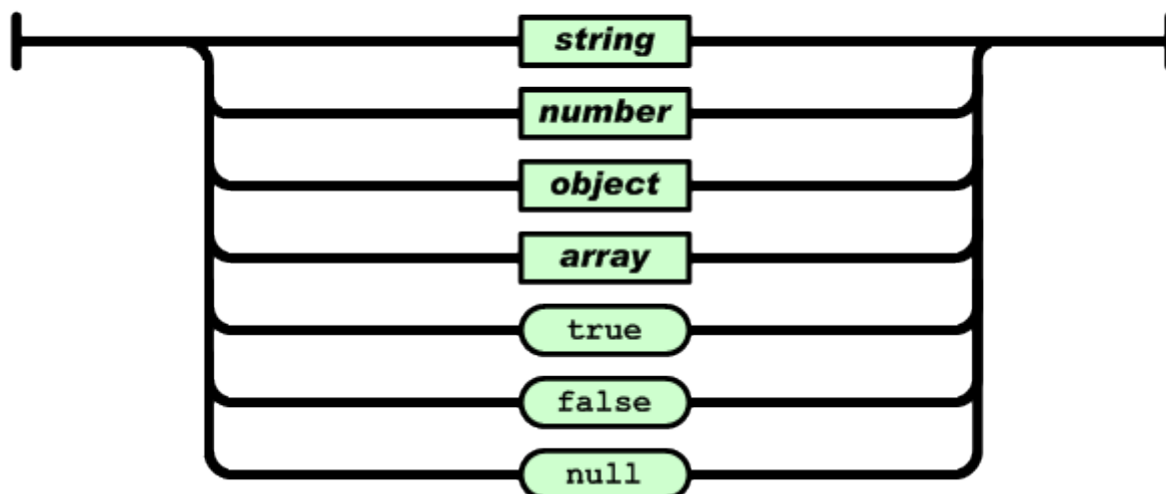
Pole – seřazená kolekce hodnot

Zápis: [hodnota1, hodnota2, hodnota3]



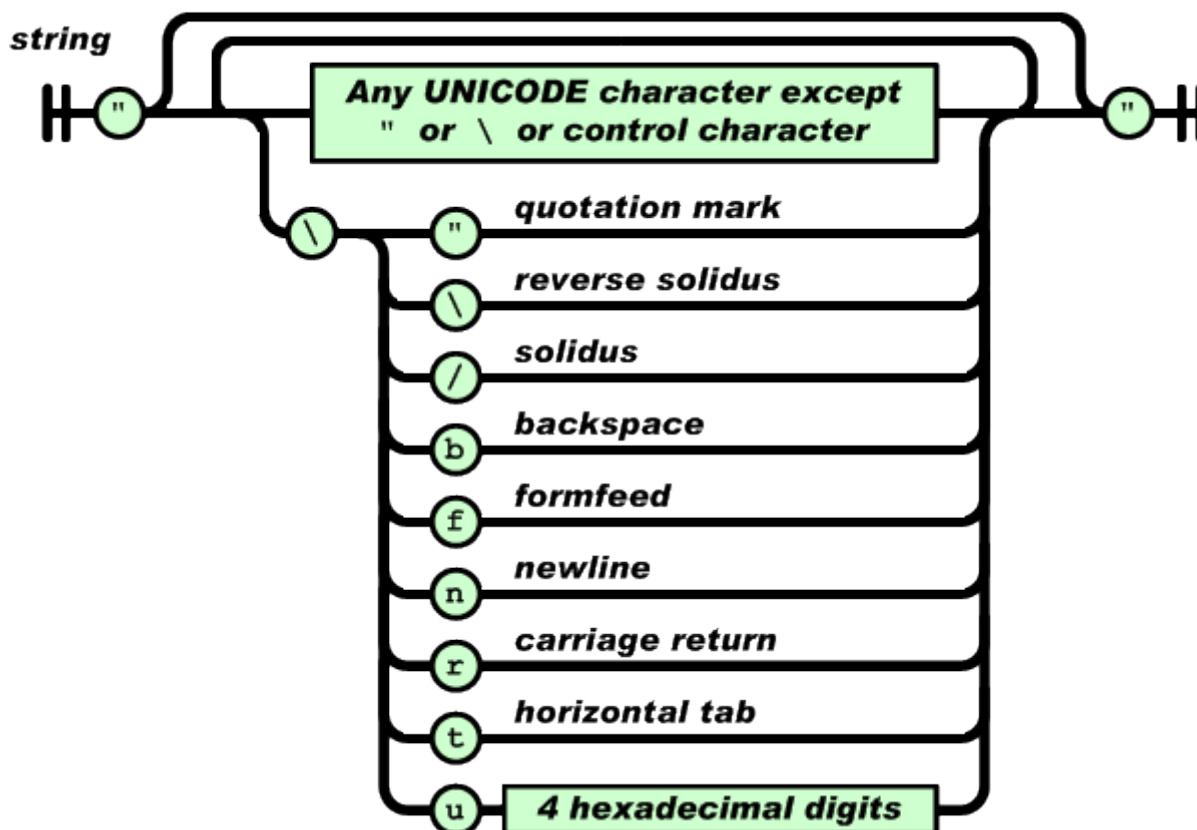
Hodnota – řetězec uzavřený do dvojitých uvozovek, číslo, true, false, null, objekt nebo pole. Tyto struktury mohou být vnořovány

value



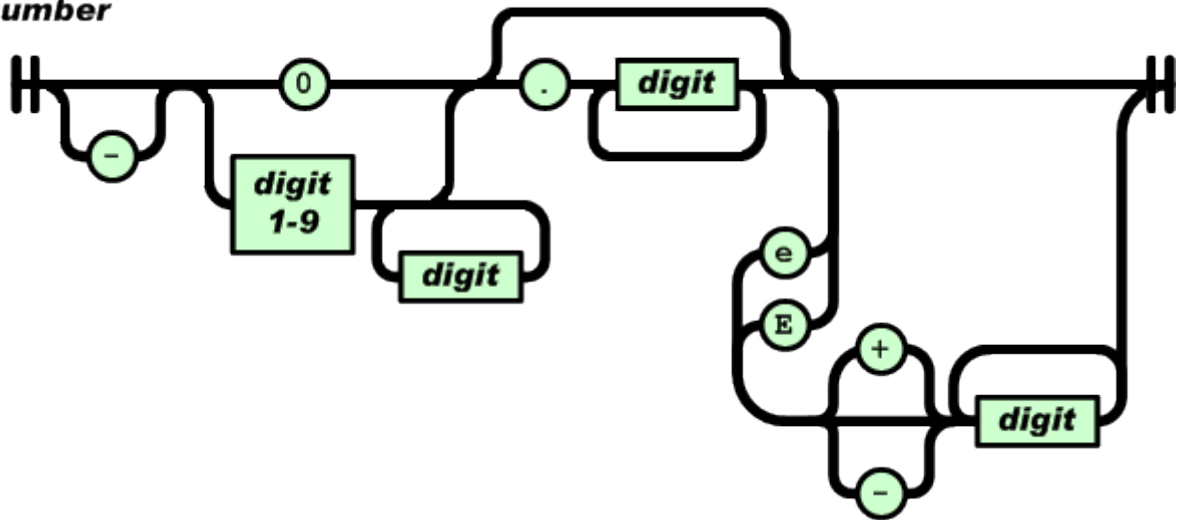
Řetězec – nula nebo více znaků Unicode, uzavřených do dvojitých uvozovek a využívající escape sequence s použitím zpětného lomítka.

Znak – řetězec s jediným znakem



Číslo – podobné číslům z jazyků C a Java, není používán oktalový ani hexadecimální zápis

number



7. SERVEROVÉ ČÁSTI WEBOVÝCH TECHNOLOGIÍ

Nejprve je nutné říci, co je to webový server.

Za webový server lze označit počítač, který je odpovědný za vyřizování požadavků HTTP(S) od klientů (nejčastěji webových prohlížečů). Vyřízením požadavků se rozumí odeslání cíle specifikovaného URL (typicky webová stránka, ale též statický text, obrázek či jiný soubor). Webové stránky jsou obvykle dokumenty v jazyku HTML. Nebo počítačový program, který provádí činnosti popsané výše (démon).

HTTP je internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML, používá tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), který specifikuje jednoznačné umístění nějakého zdroje v Internetu. HTTP neumožňuje šifrování ani zabezpečení integrity dat, k tomu slouží HTTPS.

HTTPS je protokol umožňující zabezpečenou komunikaci v počítačové síti, využívá protokol HTTP spolu s protokolem SSL nebo TLS.

Výhody HTTPS jsou

- ověření identity
- důvěrnost přenášených dat
- integrita obsahu
- možnost využití HTTP/2 protokolu
- zvýhodnění ve vyhledávači Google

Nevýhody:

- pokles výkonu u staršího hardwaru
- potřeba certifikátu a jeho obnovování
- neumožňuje blokování konkrétních URL pouze blokování celého webu
- mírně složitější konfigurace webového serveru
- možné komplikace u starších webových prohlížečů

Každý webový server je připojen k počítačové síti a přijímá požadavky ve tvaru HTTP. Požadavky vyřizuje a počítači, vrací odpověď, obvykle nějaký HTML dokument. (případně text, obrázek apod). Odpověď serveru je ve tvaru HTTP, je uvozena hlavičkou obsahující stavový kód, za níž následuje samotný obsah.

Odpovědi webového serveru (stavový kód)

- 2xx – úspěšné vyřízení požadavku
- 3xx – problémy spojené s přesměrováním

- 4xx – chyby související s vyřízením požadavku (stránka není dostupná, apod.)
- 5xx – interní chyby serveru

Zdrojem informací serveru může být:

- Statický obsah – předem připravené datové soubory (HTML stránky), výrazně rychlejší než dynamický.
- Dynamický obsah – na základě požadavku jsou shromážděna data (přečtena ze souboru, databáze, nebo nějakého koncového zařízení), zformátována a připravena k prezentaci ve formátu HTML a poskytnuta webovému prohlížeči
- Dynamické vytváření obsahu – řada různých technologií (Perl, PHP, ASP, ASP.NET, JSP, Python apod.), lze poskytovat mnohem větší obsah informací a lze reagovat i na různé „ad hoc“ dotazy klientů

V praxi se kombinují oba přístupy – například cachování, node.js,...

V reálném provozu může docházet k přetížení webového serveru.

Příznaky přetížení:

- pomalá odezva serveru (od jednotek po stovky s)
- Chyby 500, 502, 503, 504
- TCP spojení je nuceno se restartovat ještě před tím, než přijde odpověď
- server odešle nekompletní obsah (toto chování je většinou způsobeno chybou)

Důvody přetížení:

- Klasické přetížení (příliš mnoho lidí se připojí ve stejný čas, ale ne z důvodu útoku)
- DDoS útok, Počítačový vir, který napadne mnoho počítačů a donutí je se připojit
- Internetový bot
- Přetížení fyzické sítě
- Obsah je rozložený na více serverech a některý z nich není dostupný. Všechny dotazy musí obsloužit jen jeden server

Techniky pro zamezení přetížení:

- kontrola síťového provozu pomocí firewallů, HTTP traffic managerů a traffic shapingu
- použití webových cache
- použití rozdílných doménových jmen pro statické a dynamické dotazy
- použití rozdílných doménových jmen a/nebo počítačů pro oddělení velkých souborů, aby ty malé mohly být uloženy v cache
- použití více webových serverů na jednom počítači, každý s vlastní síťovou kartou
- použití více počítačů propojených dohromady a navenek se jevící jako jeden velký server

- přidání více hardware (RAM,CPU)
- vyladění použitého software

8. SERVEROVÉ ČÁSTI WEBOVÝCH TECHNOLOGIÍ II

Webových serverů existuje několik typů. Nejčastějšími jsou:

- Apache HTTP server
- IIS – Internet Information Service
- nginx
- GWS – Google Web Server

Apache HTTP server je softwarový webový server s otevřeným kódem pro GNU/Linux, BSD, Solaris, Mac OS X, Microsoft Windows a další platformy.

Podporuje velké množství funkcí – kompilované moduly rozšiřující jádro, podpora programovacích jazyků na straně serveru (Perl, Python, Tcl, PHP...), různá autentizační schémata (mod_access, mod_auth, mod_digest, a mod_auth_digest), podpora SSL, TLS (mod_ssl), proxy modul (mod_proxy), URL rewriter známý jako rewrite engine z modulu mod_rewrite, konfigurace souborů logu (mod_log_config) a filtrace (mod_include a mod_ext_filter)

Obsahuje externí modul pro kompresi dat webových stránek (mod_gzip), open source modul pro ochranu a prevenci webových aplikací před napadením (mod_security)

Logy mohou být analyzovány pomocí browseru a skriptů jako AWStats/W3Perl nebo Visitors.

Podporuje mnoho grafických prostředí (GUI) a virtuální hosting – jedna instalace Apache na jednom fyzickém počítači obsluhuje více webových stránek

IIS – Internet Information Service je softwarový webový server s kolekcí rozšiřujících modulů, vytvořený společností Microsoft pro operační systém Windows.

Podporuje řadu protokolů – HTTP, HTTPS, FTP, FTPS, SMTP a NNTP

Moduly pro IIS 7.5

- FTP Publishing Service – Publikování obsahu bezpečně na IIS 7 servery s SSL ověřování a přenos dat.
- Administration Pack – Podporu pro správu UI funkcí správy ve službě IIS 7, včetně oprávnění ASP.NET, vlastní chyby, FastCGI konfiguraci a filtrování požadavků.
- Application Request Routing – Poskytuje proxy-směrovací modul, který předá HTTP požadavky na obsahové servery, založené na HTTP hlavičce serverových proměnných a vyrovnávacích algoritmech.

- Database Manager – Umožňuje snadnou správu místní a vzdálené databáze v rámci služby IIS Manager.
- Media Services – Spojuje mediální platformu s IIS k řízení a spravování multimédií a dalšího webového obsahu.
- URL Rewrite Module – Poskytuje přepisovací mechanismus, jenž změní URL žádost před tím, než jsou zpracovány na webový server.
- WebDAV – Umožňuje autorům webu publikovat obsah bezpečně na IIS 7 servery.
- Web Deployment Tool – Synchronizuje IIS 6.0 a IIS 7 servery. Mění IIS 6.0 na IIS 7

Nginx je softwarový webový server s load management a reverzní proxy s otevřeným zdrojovým kódem. Pracuje s protokoly HTTP (i HTTPS), SMTP, POP3, IMAP a SSL. Zaměřuje se především na vysoký výkon a nízké nároky na paměť. Je dostupný na Unixu, Linuxu a Unix-like systémech pod BSD, existují varianty pro Solaris, macOS i MS Windows.

Základním cílem Nginx je rychlá distribuce statického obsahu, možnost rozložení zátěže na další servery dle nastavené priority.

Systém umožňuje definovat záložní server, na který Nginx požadavek předá, pokud primární server neodpoví do stanoveného limitu

Příchozí požadavky Nginx asynchronně zpracovává a vyřizuje

Příchozí HTTP (nebo HTTPS) požadavek se nejprve pokusí vyhledat ve své cache (má konfigurovatelnou velikost a dobu uchovávání), pokud jej najde, rovnou odpoví. V opačném případě se obrátí na jeden z definované sady serverů (každý server má definovanou prioritu). Pokud mu server do definovaného času stihne odpovědět, předá odpověď; v opačném případě se obrátí na záložní server (samozřejmě je-li definován). Odpověď, pokud může, uloží do své cache a následující dotazy do vypršení časového limitu životnosti cache vyřizuje právě z cache.

Možnost nastavit limit počtu připojení z jedné IP adresy

Nginx je modulární systém

Jeden z modulů – GEO lokace – umožňuje například dle země předávat požadavky na definované servery, nebo zakázat přístup na stránky z některých zemí

Moduly pro přesměrování dle definovaných pravidel, zabezpečení stránky heslem, podpora komprese gzip, streaming (FLV, MP4)...

9. PHP: HYPERTEXT PREPROCESSOR

Programovací paradigma PHP - imperativní, objektivě orientované, procedurální, reflektivní

Vznikl v roce 1995, jeho autorem je Rasmus Lerdorf. První vydání proběhlo 8. června 1995 a zatím poslední verze je 7.2.0 (30. listopad 2017)

PHP se vyznačuje slabou a dynamickou typovou kontrolou.

Hlavní implementace PHP jsou Zend Engine, Phalanger, Quercus, Project Zero, HipHop

PHP je skriptovací programovací jazyk určený především pro programování dynamických internetových stránek a webových aplikací. Skripty jsou prováděny na straně serveru – k uživateli je přenášěn až výsledek jejich činnosti. Interpret PHP skriptu je možné volat pomocí příkazového řádku, dotazovacích metod HTTP nebo pomocí webových služeb. Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java)

Jazyk PHP je nezávislý na platformě, rozdíly v různých operačních systémech se omezují na několik systémově závislých funkcí a skripty lze většinou mezi operačními systémy přenášet bez jakýchkoli úprav. Podporuje mnoho knihoven pro různé účely – např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), řadu internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, ...).

PHP je nejrozšířenějším skriptovacím jazykem pro web

Vlastnosti jazyka PHP:

- Jazyk PHP je dynamicky typovaný – datový typ proměnné je vázán na hodnotu, nikoliv na proměnnou.
- Pole jsou asociativní
- Řetězce lze v PHP zapsat 2 různými způsoby:
 - uzavírat do uvozovek (při vyhodnocení se provede nahrazení proměnných uvnitř)
 - uzavírat do apostrofů (nahrazuje se jen escape sekvence \).
- Proměnné lze vytvářet i rušit
- Konstanty lze definovat, nelze je zrušit
- Proměnné mají své úrovně viditelnosti a pravidla pro jejich perzistenci.
- Podporuje reference, pomocí kterých lze do proměnných ukládat odkazy na libovolnou jinou proměnnou, nebo i prvek jejího pole

- Jako reference lze volat i parametry funkce - u každé proměnné eviduje, kolik na ni směřuje referencí, a podle toho se rozhoduje, kdy může kterou proměnnou zrušit.

Výhody:

- PHP je specializované na webové stránky.
- Rozsáhlý soubor funkcí v základní knihovně PHP (přes pět a půl tisíce), další funkce v PECL.
- Nativní podpora mnoha databázových systémů.
- Multiplatformnost (zejména Linux a Microsoft Windows)
- Možnost využití nativních funkcí operačního systému (možná nekompatibilita s jiným OS)
- Strmá křivka učení.
- Obrovská podpora na hostingových službách
- Obrovské množství projektů a kódů, které lze zdarma využít (WordPress, phpBB a další).
- Poměrně slušná dokumentace
- Velmi svobodná licence

Nevýhody:

- Jazyk PHP byl dlouho definován pouze svou implementací, oficiální specifikace jazyka byla oznámena na konci července 2014
- Nekonzistentní pojmenování funkcí
- `strpos()`, `strchr()`, ale `str_replace()`, `str_pad()`.
- Nejednotné názvosloví skupin funkcí
- `mysql_XXXX`, `imap_XXXX`, `json_XXXX` (s podtržítkem) versus `imageXXXX`, `bcXXXX`, `gzXXXX` (bez podtržítka).
- Nejednotné pořadí parametrů, např.: `array_map()` vs. `array_filter()`.
- Ač jazyk podporuje výjimky, jeho knihovna je používá jen zřídka.
- Slabší podpora Unicode, pouze přes PHP knihovnu (ve verzích po PHP 5 má být Unicode řetězec jako základní typ).
- Ve standardní distribuci chybí ladící (debugovací) nástroj.
- Po zpracování požadavku neudrží kontext aplikace, vytváří jej vždy znovu (oslabuje výkon).

Jazyk PHP není určen jen pro malé projekty a stránky, lze v něm napsat (naprogramovat) libovolnou aplikaci.

Vybrané významné projekty v PHP:

- MediaWiki – software pro tvorbu webových projektů typu wiki,
- phpBB – balík pro provoz webového fóra
- WordPress – publikační systém pro provoz blogů a podobných aplikací
- Adminer – webová aplikace pro správu databázového systému MySQL
- phpMyAdmin – webová aplikace pro správu databázového systému MySQL
- Taxy! – překladač intuitivní syntaxe pro formátování textu na HTML
- Nette Framework – framework pro tvorbu webových aplikací v PHP
- Facebook

10. PHP II – SYNTAXE

PHP skript lze v HTML označit několika způsoby

- `<? [PHP kód] ?>`
- `<?php [PHP kód] ?>`
- `<SCRIPT LANGUAGE="php"> php [PHP kód] </SCRIPT>`

Jednotlivé instrukce (příkazy) se oddělují středníkem. Komentáře se uvozují dvojitým lomítkem nebo mřížkou. Víceřádkové komentáře jsou zapsány mezi lomítko hvězdička – hvězdička lomítko (`//`, `#`, `/*` víceřádkový text `*/`)

Typy proměnných v PHP

- Proměnné
- Logický typ – Boolean – hodnota pravda /nepravda, zapisuje se jako TRUE a FALSE (na velikosti nezáleží)
- Celočíslný typ – Integer – celá kladná i záporná čísla (a nulu) od nějakých -2 biliónů po + 2 bilióny
- Desetinné číslo – Float, Real - s přesností obvykle na 14 desetinných míst
- Řetězec – String – textové řetězce

Typ proměnné se určuje v okamžiku přiřazení hodnoty, během programu může proměnná svůj typ změnit, ať už díky instrukci v kódu nebo v důsledku nějakého výpočtu. Každá proměnná musí mít jednoznačný název, začíná znakem dolaru (\$) a bez mezery následuje pojmenováním. První znak toho pojmenování musí být buď písmeno a-z nebo podtržítka. Nesmí to být číslo ani nic jiného. Názvy proměnných rozlišují mezi malými a velkými písmeny. Pro přiřazení se používá znak rovná se (=)

PHP umožňuje definovat tři typy polí:

- Indexovaná
- Asociativní
- Vícerozměrná

Ta se využívají jako seznamy, simulace slovníků a kolekce prvků. S poli můžeme pracovat jako se zásobníky nebo frontami, mohou také představovat stromové struktury (prvkem pole totiž může být pole).

Pole mohou být vracena z funkcí PHP (databáze).

PHP patří mezi OOP jazyky.

```
Class ClassName
{
    var $ VariableName
    function Function Name (parameters)
    {
        body function
    }
}
```

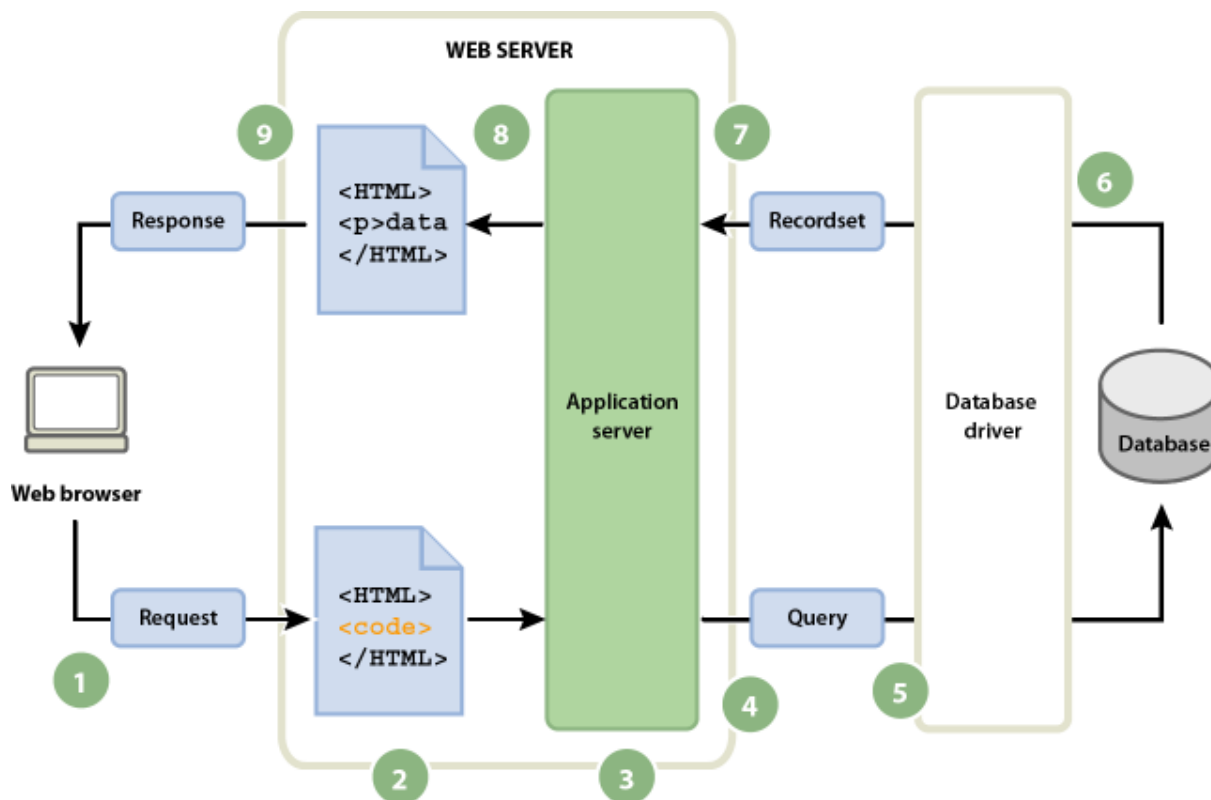
Pozor na dědičnost – PHP nemá privátní metody (funkce).

Navíc PHP umožní zadávat a měnit NEDEKLAROVANÉ atributy!

Mnoho hotových knihoven a částí kódů PHP objekty používá, můžete klidně mít jen jednu "úroveň" objektů bez dědičností. Objekt může existovat bez metod, jen s atributy. Ale někdy je pak lepší použít pole. Někdo zase v PHP pro "přiřazení" každého atributu vytváří metodu nebo metody. Při použití objektů platí více než kdy dříve, že byste si měli vymyslet a dodržovat konvence (třeba, někdo začíná metody, které by se neměly volat zvnějšku objektu podtržítkem).

11. WEBOVÉ APLIKACE SPOLUPRACUJÍCÍ S DATABÁZEMI

Postup zpracování webového požadavku do databáze.



1. Požadavek – request
2. HTML kód
3. Application server
4. Dotaz – Query
5. Database driver
6. Výstup z databáze
7. Recordset pro server
8. Překlad do HTML
9. Odpověď - Response

Základem pro práci s databázemi je jazyk SQL.

SQL – Structured Query Language – strukturovaný jazyk pro práci s relačními databázemi

Databáze je systém souborů s pevnou strukturou záznamů, kde soubory jsou mezi sebou navzájem propojeny pomocí klíčů.

Typy databází:

- Hierarchická databáze
- Síťová databáze
- Relační databáze
- Objektová databáze
- Objektově relační databáze

Databázové objekty

- TABLE - základní databázový objektů, který slouží k přímému uložení dat do paměťového prostoru relační databáze
- VIEW - databázový objekt, který uživateli poskytuje náhled na data obsažená v tabulce
- INDEX (KEY) – slouží ke zrychlení vyhledávacích a dotazovacích procesů, definování unikátní hodnoty sloupce tabulky, optimalizaci vyhledávání
- CONSTRAINT – umožňuje vytvořit omezení s podmínkami, jež musí být splněny pro hodnoty jejich sloupců při vkládání nebo změnu záznamů
- TRANSACTION – skupina příkazů, které převedou databázi z jednoho konzistentního stavu do druhého
- TRIGGER – definuje činnosti, které se mají provést v případě definované události nad databázovou tabulkou

SQL příkazy pro manipulaci s daty (DML)

- SELECT – vybírá data z databáze, umožňuje výběr podmnožiny a řazení dat.
- INSERT – vkládá do databáze nová data.
- UPDATE – mění data v databázi (editace).
- MERGE – kombinace INSERT a UPDATE – data buď vloží (pokud neexistuje odpovídající klíč), pokud existuje, pak je upraví ve stylu UPDATE.
- DELETE – odstraňuje data (záznamy) z databáze.
- EXPLAIN – speciální příkaz, který zobrazuje postup zpracování SQL příkazu. Pomáhá uživateli optimalizovat příkazy tak, aby byly rychlejší.
- SHOW - méně častý příkaz, umožňující zobrazit databáze, tabulky nebo jejich definice

SQL příkazy pro definici dat (DDL)

- CREATE – vytváření nových objektů.
- ALTER – změny existujících objektů.
- DROP – odstraňování objektů.

SQL příkazy pro řízení dat (DCL)

- GRANT – příkaz pro přidělení oprávnění uživateli k určitým objektům.
- REVOKE – příkaz pro odnětí práv uživateli.
- START TRANSACTION – zahájení transakce.
- COMMIT – potvrzení transakce.
- ROLLBACK – zrušení transakce, návrat do původního stavu.

Klíčová slova SQL pro dotazování

- TOP – vrátí prvních N řádků
- LIMIT – omezení počtu řádků vrácených příkazem SELECT
- JOIN (FULL LEFT RIGHT INNER CROSS) ON – spojování výsledku dotazu SELECT ze dvou vstupních množin (typicky tabulek relační databáze)
- UNION – sjednocení výsledku dotazu ze dvou nebo více vstupních množin dotazu SELECT
- ORDER BY – seřazení záznamů vybíraných pomocí příkazu SELECT
- WHERE – omezuje výběr řádků z tabulek pomocí podmínek
- GROUP BY – agregace záznamů vybíraných pomocí příkazu SELECT
- WITH ROLLUP – za standardním výpisem se objeví řádek s hodnotou NULL na místo sloupce, podle kterého jsou data agregována (je-li uveden)
- HAVING – umožňuje omezit řádky, které jsou zpracovány agregační funkcí

Databáze a databázové servery

- Microsoft Access
- MySQL
- Oracle
- Microsoft SQL Server
- SQLite

12. DOM – DOCUMENT OBJECT MODEL

DOM je objektový model dokumentu, API (application programming interface – aplikační programové rozhraní), které definuje obecný standard pro přístup k jakémukoliv platnému HTML dokumentu nebo správně strukturovanému XML dokumentu, je zcela nezávislé na programovacím jazyku

Pomocí DOM lze je obsluhovat jednotlivé prvky (objekty) pomocí JavaScriptu.

Definice DOM popisují jednotlivé úrovně

- **Level 0**
 - Podpora Intermediate DOM, jenž existoval před vytvořením DOM Level 1. Například DHTML Object Model vyvinutý firmou Microsoft, nebo nepojmenovaný Intermediate DOM od Netscape. Level 0 není formální specifikací publikovanou W3C, ale používá se jako srozumitelná zkratka odkazující na věci existující před standardizačním procesem.
- **Level 1**
 - Navigace v DOMu (HTML a XML) dokumentu (resp. jeho stromové struktury) a manipulace s obsahem (včetně přidávání elementů). Specifické elementy HTML jsou obsaženy také.
- **Level 2**
 - Podpora jmenných prostorů, událostí a filtrovaných pohledů.
- **Level 3**
 - Standardizovaný mechanismus načítání a ukládání a podpora XML schémat. Umožňuje dynamické vkládání obsahu do dokumentu a přidává nové metody a vlastnosti.
- **Level 4**
 - Sloučení předchozích standardů DOM Level 3 Core, Element Traversal, Selectors API Level 2, DOM Level 3 Events a DOM Level 2 Traversal and Range a jejich zjednodušení a přiblížení již existujícím standardům, především specifikacím JavaScriptu a HTML5. Specifikace dále zjednoduší časté DOM operace.

Základní myšlenkou DOM je považovat HTML elementy za objekty. Pak každý objekt má vlastnosti – atributy – a může reagovat na události.

Každý objekt musí být identifikován, to je nezbytné pro změnu vlastnosti nebo obsahu pomocí skriptu. Identifikace probíhá pomocí id nebo name.

DOM – univerzální vlastnosti a metody pro procházení a čtení (použití: document.Nazev)

- documentElement - vrací kořenový (root) element dokumentu
- getElementsByTagName() - vrací pole všech elementů daného jména
- parentNode - vrací rodičovský objekt daného objektu
- nextSibling - vrací následujícího sourozence aktuálního objektu, pokud takový existuje, jinak vrací null
- previousSibling - vrací předchozího sourozence aktuálního objektu, pokud existuje
- firstChild - Vrací první dítě daného objektu
- lastChild - vrací poslední dítě daného objektu
- childNodes[] - vrací pole všech objektů (nodelist), které jsou dětmi daného objektu
- nodeName - vrací jméno objektu
- nodeValue - vrací hodnotu (obsah) daného objektu
- data - vrací hodnotu (obsah) objektu typu Text
- className - vrací hodnotu atributu class (pouze pro HTML)
- id - vrací hodnotu atributu id objektu, který musí být typu element, pouze pro HTML
- title - vrací hodnotu atributu title daného elementu, pouze HTML
- item() - vrací určitou položku pole objektů (nodelist) specifikovanou indexem

DOM – metody pro manipulaci s uzly

- createElement(jméno) - vytvoření nového elementu
- setAttribute(jméno, hodnota) - nastavení atributu
- createTextNode(hodnota) - Vytvoří textový uzel
- splitText(pozice) - rozdělení textového uzlu na uzly dva
- normalize() - Sloučí sourozenecké uzly typu Text do jednoho uzlu
- appendChild(objekt) - Přidá potomka uzlu
- insertBefore(objekt, objekt) - 2 parametry – první je uzel, který budeme vkládat, a druhý je uzel, před který budeme vkládat
- cloneNode(true | false) - vytvoří kopii daného objektu
- replaceChild(objekt, objekt) - 2 parametry – první parametr je uzel, který nahradí uzel uvedený jako druhý parametr
- removeChild(objekt) - aplikuje na rodiče uzlu, který má být odstraněn
- removeAttribute(jméno) - Odstraní atribut daného uzlu

DOM lze také využít pro efektivní tvorbu dynamických tabulek, případně pro dynamickou úpravu formátování stránky pomocí CSS.