

Otázka: HTML, CSS, JS, PHP

Předmět: Informační technologie

Přidal(a): Katyz

HTML (Hyper Text Markup Language)

Značkovací jazyk, který **definuje strukturu webových stránek a tvoří tak základ při jejich tvorbě**. Značkovací proto, že obsahuje množinu značek (**tagy**), kde většina z nich je párová, tzn., že obsahují počáteční a koncový tag. **Párový tag** definuje význam obsahu, který se v něm nachází.

Počáteční tag, obsah a koncový tag se souhrnně označuje jako element. Element p definuje na webové stránce odstavec - `<p>` je počáteční tag, `</p>` je koncový tag a textový obsah, který se nachází mezi nimi je poté definován jako odstavec.

Množina tagů HTML však obsahuje i **tagy nepárové**. Takové tagy neobsahují koncový tag a nelze tak mezi ně vložit textový obsah. Jako příklad lze uvést jedny z nejpoužívanějších nepárových tagů - tag `
`, který značí konec řádku a tag ``, pomocí kterého se vkládají do stránky obrázky.

Počáteční tagy v sobě mohou nést další informace o elementu. Tyto informace se definují pomocí tzv. **atributů**. ``

Element **header** představuje hlavičku stránky. Typicky uvnitř něj najdeme nadpis, navigaci, logo či vyhledávání. Tento element se v některých případech používá na stránce i vícekrát, například jako hlavička dané sekce.

```
<header>
  <h1>nadpis</h1>
  <nav>
    <ul>
      <li>položka1</li>
      <li>položka2</li>
    </ul>
  </nav>
</header>
```

Element **footer** představuje patičku stránky. Tradičně uvnitř něj najdeme autorské údaje a odkazy na související stránky. Stejně jako u elementu header platí, že může být použit na stránce vícekrát.

Element **section** reprezentuje určitý úsek stránky - tematická seskupení obsahu (typicky s vlastním nadpisem).

Element **article** tvoří tu část stránky, která je chápána jako nezávislý a samostatný celek. Jeho obsah by tak měl dívat smysl i sám o sobě.

Element **nav** představuje tu část stránky, ve které se odkazuje na další stránky, anebo na určitou část aktuální stránky.

CSS

Dalo by se označit jako sbírka metod pro grafickou úpravu webových stránek. Je to zkratka z **Cascading Style Sheets**, což se česky překládá jako **kaskádové styly**. Kaskáda je vlastnost, která říká, že definice mohou vrstvit jedna na druhou, platí vždy ta poslední. Její

syntaxe se skládá ze dvou částí - selektor a blok deklarácí. Selektor ukazuje na HTML element, který má být formátován. Každý blok deklarácí je ohraničen složenými závorkami a skládá se z vlastnosti a její hodnoty (popř. i více hodnot), mezi nimiž je dvojtečka. Každá dvojice vlastnost-hodnota je od sebe oddělena středníkem a příslušný blok deklarácí takovýchto dvojic může obsahovat libovolné množství.

```
<p style="color: red">odstavec</p>
```

Do hlavičky dokumentu můžeme zapsat **style** uzavřený mezi tagy `<style> </style>`.

```
<style>
```

```
p {color: red}
```

```
</style>
```

JavaScript

Přidává stránkám dynamickou funkcionalitu a interaktivitu. Běží na straně klienta v prohlížeči a v dnešní době se s ním setkáme v nějaké formě téměř na každé větší webové stránce. Javascript pohání jedny z největších a nejpoužívanějších webových aplikací jako jsou Google Maps, Gmail a Facebook. Výsledný program je tvořen sekvencí příkazů a každý příkaz se v JavaScriptu odděluje středníkem. Všechny identifikátory v JavaScriptu jsou case sensitive, to znamená, že záleží na velikosti písmem. Tak například proměnná `krestnijmeno` a `krestnijmeno` jsou v daném skriptu považovány jako dvě různé proměnné. Při deklaraci proměnných se používá klíčové slovo `var`.

```
<script>

document.write(„<a href=\http://www.seznam.cz\“>odkaz na
Seznam</a>“)

</script>
```

PHP

Flexibilní jazyk, jenž pracuje na straně serveru. Základy má v jazycích C a Perl, i když často připomíná spíše Javu. PHP skript začíná `<?php` a končí `?>`. Výchozí koncovka pro PHP soubory je `.php`. Klíčovou roli v syntaxi hrají ještě znaménka dolaru a středníku.

Znak dolaru využijeme vždy, když deklarujeme proměnnou. Každá proměnná v PHP musí začínat právě znakem dolaru. Díky tomu může PHP parser pracovat rychleji, protože ihned ví, že se jedná o proměnnou. Každý PHP příkaz musí končit znakem středníku.

```
<?php

$cislo = 1;

$retezec = „Ahoj“;

$pole = [];

?>
```