



Přednáška 1: Základy HTML a CSS

Historie a vývoj HTML (1 / 3)

HTML (Hypertext Markup Language) je základní **značkovací jazyk**, umožňující publikování informací na internetu. Jazyk byl **do verze HTML 4** aplikací dříve vyvinutého univerzálního značkovacího jazyka **SGML** (Standard Generalized Markup Language). Aplikací proto, že meta-jazyk SGML umožňuje pomocí schémat **DTD** (Document Type Definition - Doctype) definovat, které značky jsou přípustné a jak a kde je používat. Definováním vhodné **podmnožiny značek** a **vytvořením** příslušného **DTD** tedy vzniklo HTML.

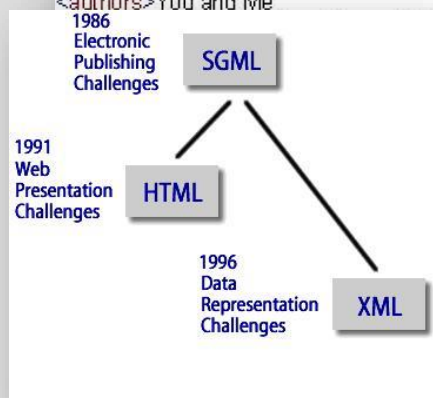
Historie HTML

Na začátku bylo SGML

V roce 1986 byla vydána norma **ISO 8879**, která si kladla za cíl vytvořit formát, který by umožňoval uložení textů v elektronické podobě a přitom byl **nezávislý jak na softwarové, tak na hardwarové platformě** a poskytoval dostatečnou flexibilitu. Jazyk definovaný touto normou dostal název

Standard Generalized Markup Language (SGML).

```
<!DOCTYPE article [
<!ELEMENT article      -- (header,body)>
<!ELEMENT header      - O ((title & authors) & abstract?)>
<!ELEMENT body        - O ((block*, ack)>
<!ELEMENT title       - O (#PCDATA)>
<!ELEMENT authors     - O (#PCDATA)>
<!ELEMENT abstract    - O (#PCDATA)>
<!ELEMENT block       - O ((#PCDATA| link) *)>
<!ELEMENT link        -- (#PCDATA)>
<!ATTLIST link        url CDATA #REQUIRED>
<!ELEMENT ack         - O (#PCDATA)>]>
<article>
<header>
<title>Article on SGML, HTML and XML
<authors>You and Me
```



of electronic publishing

actually the Document Type Definition (DTD). It defines how any article like

like the "block" tag need not be closed but some such as the "article" tag

has <link url="http://www.myurl.com/epubstuff">A link to

of SGML



Přednáška 1: Základy HTML a CSS

Historie a vývoj HTML (2 / 3)

Tim Berners-Lee a zrození HTML

V roce **1989** britský informatik **Tim Berners-Lee** vytvořil návrh na **hypertextový dokumentový systém** určený pro použití v rámci komunity **CERN** (Evropská organizace pro jaderný výzkum, [CERN Data Centre](#)). Ačkoliv má **CERN** sídlo ve Švýcarsku, jeho členové byli rozptýleni po celém světě a projekty se často střídaly. Požadavek **spolupráce na dlouhé vzdálenosti, rychlé začleňování** nových členů do týmů a **uchování informací** v souvislosti s rychlou obměnou týmů byly hnací silou ve vývoji navrhovaného systému. Tento systém, který Berners-Lee, později nazval termínem **"World-Wide Web"** (v říjnu 1990), definoval několik důležitých podmínek nutných k realizaci této vize, a které v kostce, definují i povahu dnešního WWW:



- Musí být **multiplatformní**, tj. nezávislý na konkrétním počítači, operačním systému, atd.
- Musí být schopen **používat mnoho stávajících informačních zdrojů**, systémů a zároveň umožňovat snadné **přidávání nových informací**.
- Vyžaduje **transportní mechanismus** pro přesunování dokumentů po sítích (vyvinul se do HTTP)
- Je potřeba **identifikační schéma pro adresování** lokálních i vzdálených hypertextových dokumentů (vyvinul se v URL řešení)
- Vyžaduje **jazyk pro formátování hypertextových dokumentů**. Toto nebylo výslovně uvedeno, ale bylo to nedílnou součástí prezentace získaných informací (vyvinul se do HTML).



Přednáška 1: Základy HTML a CSS

Historie a vývoj HTML (3 / 3)

V roce **1990** Tim Berners-Lee napsal první webový prohlížeč, který nazval **WorldWideWeb**. Prohlížeč běžel na platformě **NeXT**, využíval základní stylové předpisy (stylesheets) a umožňoval stahování a otevírání souborů, které podporoval mateřský OS **NeXTSTEP** (Postscript, videa a zvuky). Obrázky zpočátku otevíral v samostatných oknech. Využíval pro tento účel vytvořený protokol HTTP a jazyk HTML.

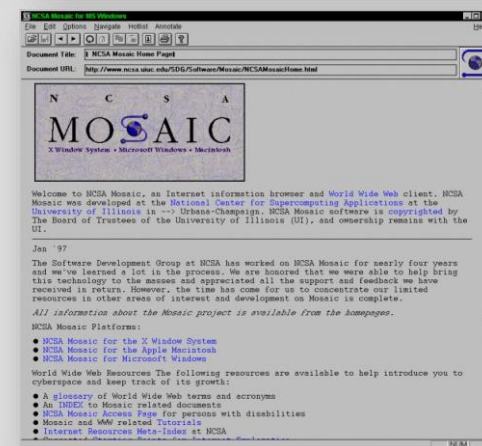
První www stránka – 6. srpna 1991:

<http://info.cern.ch/hypertext/WWW/TheProject.html>

V roce **1991** CERN sprovoznil svůj **web**, resp. **diskuzní skupinu** na téma dalšího vývoje HTML. Krátce poté, v roce **1992**, organizace **NCSA** (National Center for Supercomputer Applications) vybídla Marca Andreessena a Erica Binu k vytvoření prohlížeče **Mosaic**; ten vznikl v roce **1993** ve verzích pro počítače IBM PC a Macintosh a měl obrovský úspěch. Byl to první prohlížeč **s grafickým uživatelským rozhraním**. Následoval prohlížeč **Netscape Navigator** (1994) a **Internet explorer** (1995).

Vývoj webových prohlížečů také zpětně ovlivňoval definici jazyka HTML.

Přehled zapomenutých prohlížečů z počátku 90. let



Mosaic



Přednáška 1: Základy HTML a CSS

HTTP a URL

HTTP a URL

HTTP (Hypertext Transfer Protocol) je internetový protokol určený původně pro **výměnu hypertextových dokumentů ve formátu HTML**. Používá obvykle port TCP/80, existují dvě základní verze a to verze 1.0 a 1.1. Druhá zmíněná verze umí použít stejné spojení pro stažení textu a následně i obrázků. Tento protokol je spolu s elektronickou poštou tím nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech.

V současné době je používán i pro přenos dalších informací. Pomocí rozšíření **MIME** umí **přenášet jakýkoli soubor** (podobně jako e-mail), používá se společně s formátem XML pro tzv. **webové služby** (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP.

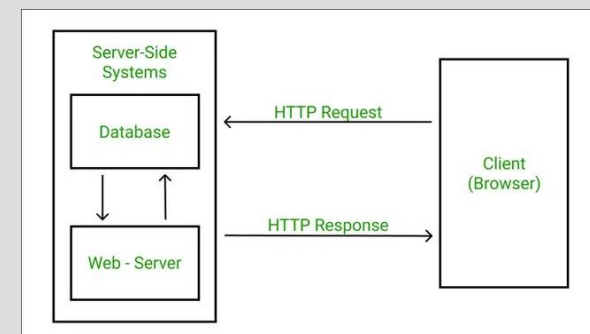
K protokolu HTTP existuje také jeho bezpečnější verze **HTTPS**, která umožňuje přenášet data šifrovat a tím chránit před odposlechem či jiným narušením.

URL

URL, celým názvem **Uniform Resource Locator** („jednotný lokátor zdrojů“) je řetězec znaků s definovanou strukturou, který slouží k přesné **specifikaci umístění zdrojů informací** (ve smyslu dokument nebo služba) na Internetu.

URL definuje schéma (typ zdroje), doménovou adresu serveru (**IP adresu**), port, umístění zdroje na serveru, parametry a kotvu (anchor).

Jednotlivá pole v URL: ***schéma://doména:port/cesta?parametry#kotva***
Existují absolutní a relativní URL.





Přednáška 1: Základy HTML a CSS

HTML

HTML do verze 4.01

Jak bylo již dříve zmíněno, HTML jazyk prošel od počátku 90. let dlouhým vývojem. Do verze 4.01 byly uveřejňovány jednotlivé verze (standards) jako tzv. doporučení (recommendations) organizace **W3C** (World Wide Web Consortium, vzniklo v roce 1994). Mezi základní milníky tohoto období patří uveřejnění **HTML 3.2** (1996), **HTML 4.0** (1998) a **HTML 4.01** (1999).

HTML 5

V roce **2004** založili reprezentanti firem Mozilla, Opera a Apple vlastní organizaci nazvanou **Web Hypertext Application Technology Working Group** ([WHATWG](#)). Cílem organizace bylo **podpořit vývoj HTML** směrem k začleňování nových technologií (Web Storage, Web Socket. ad.) Založení organizace bylo rovněž **reakcí na plánovaný odklon organizace W3C od dalšího vývoje HTML směrem k podpoře XHTML**.

V roce **2019** došlo k [dohodě](#) o vzájemné spolupráci mezi organizacemi W3 a WHATWG.

WHATWG uveřejňuje svůj, tzv. [HTML Living Standard](#), což je průběžně publikovaná specifikace HTML 5.



Specifikace HTML 5.x z webu W3C jsou nyní označeny jako „Retired“ a [návštěvníci jsou přesměrováni na WHATWG HTML Living Standard](#). Pro vývojáře je tedy stránka WHATWG relevantním zdrojem.

HTML 5.3 publication history

Date	Status
28 January 2021	Retired
18 October 2018	Working Draft
9 August 2018	Working Draft
3 July 2018	Working Draft
26 April 2018	Working Draft
15 March 2018	Working Draft
6 February 2018	Working Draft
14 December 2017	First Public Working Draft



Přednáška 1: Základy HTML a CSS

HTML - koncepce jazyka

Koncepce jazyka HTML

Jazyk HTML je od verze 2.0 aplikací **SGML** (verze **HTML5** ale již závislost na SGML neobsahuje). Je charakterizován množinou **značek** a jejich **atributů** definovaných pro danou verzi. Mezi značky se **uzavírají části textu dokumentu** a tím se určuje **význam** (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi **úhlové závorky** (< a >). Část dokumentu tvořená otevírací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. **element (prvek) dokumentu**. Například je otevírací **značka** pro zvýraznění textu a text je **element** obsahující zvýrazněný text. Součástí obsahu elementu mohou být další **vnořené elementy**. **Atributy** jsou doplňující informace, které upřesňují **vlastnosti elementu**.

Značky (anglicky **tags** (tagy)) jsou buď **párové** (v XHTML jsou párové všechny), kde koncové značka je shodná se značkou počáteční, jen má před názvem znak lomítko, nebo **nepárové**.

Příklady:

<p>odstavec</p>...párový tag

<hr>...nepárový tag

...nepárový tag s atributem src

Pro každou verzi HTML (**kromě HTML5**) existuje definice pravidel **DTD** (Document Type Definition). DTD definuje pro určitou verzi elementy a atributy, které lze používat.

Další prvky dokumentu:

- **Direktivy** – začínají znaky <!, jsou určeny pro zpracovatele dokumentu (prohlížeč). Např. „<!DOCTYPE...”
- **Komentáře** – pomocné texty pro programátora, nejsou součástí obsahu dokumentu a nezobrazují se (prohlížeč je ignoruje). Např. „<!-- text -->”
- **Kód skriptovacích jazyků**. Např. „<script type='text/javascript'>alert('ahoj svete!');</script>”
- **Definice událostí** a kód pro jejich obsluhu. Např. ''



Přednáška 1: Základy HTML a CSS

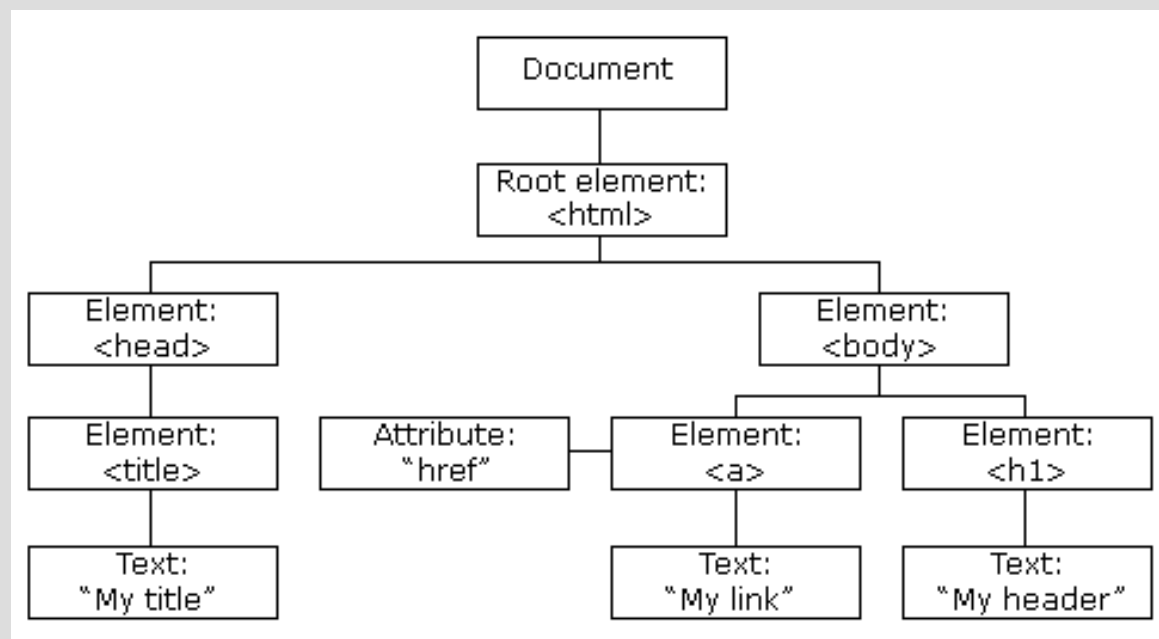
Objektový model dokumentu

Objektový model dokumentu (Document Object Model - **DOM**) je datová reprezentace objektů, které tvoří strukturu a obsah dokumentu na webu (webové stránky).

DOM je také **programovací rozhraní** (programming interface) pro webové dokumenty.

DOM tedy umožňuje programům **dynamicky přistupovat a aktualizovat obsah, strukturu a styl dokumentu**.

DOM je konstruován jako **strom** objektů:



DOM lze, jako objektově orientovanou reprezentaci webové stránky, upravovat pomocí skriptovacího jazyka, jako je např. **JavaScript**.

DOM standard na stránce WHATWG:

<https://dom.spec.whatwg.org/>



Přednáška 1: Základy HTML a CSS

Struktura dokumentu (1 / 2)

Struktura dokumentu (HTML 4.01)

Dokument v jazyce HTML 4.0.1 má předepsanou strukturu:

- **Deklarace DTD** – je povinná až ve verzi 4.01, je uvedena direktivou <!DOCTYPE.

HTML 4.01 Strict DTD

HTML 4.01 Transitional DTD

HTML 4.01 Frameset DTD

Viz. <http://www.w3.org/TR/html401/struct/global.html#version-info>

- **HTML** (kořenový) **element** – nepovinný element, který vymezuje celý dokument po deklaraci DTD.

- **Hlavička dokumentu** – obsahuje metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení, apod. Hlavička je vymezena nepovinným elementem **head**.

- **Tělo dokumentu** – obsahuje vlastní text dokumentu. Vymezuje se elementem **body** nebo elementem **frameset**.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
```

```
  <head>
```

```
    <title>Můj první HTML dokument</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>Ahoj světe!</p>
```

```
  </body>
```

```
</html>
```




Přednáška 1: Základy HTML a CSS

Struktura dokumentu (2 / 2)

Struktura dokumentu (HTML5)

Dokument v jazyce HTML5 má předepsanou strukturu:

- **Deklarace DTD** – HTML5 není založena na SGML, proto dokument nevyžaduje odkaz na DTD.
- **HTML** (kořenový) **element** – nepovinný element, který vymezuje celý dokument.
- **Hlavička dokumentu** – obsahuje metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení, apod. Hlavička je vymezena nepovinným elementem **head**.
- **Tělo dokumentu** – obsahuje vlastní text dokumentu. Vymezuje se elementem **body**.

```
<!DOCTYPE html>
<html lang="cs-cz">
  <head></head>
  <body>
    <header></header>
    <section>
      <header>
        <h1></h1>
      </header>
      <article></article>
    </section>
  </body>
</html>
```



Přednáška 1: Základy HTML a CSS

HTML soubor

HTML soubor

- prostý textový (plain ASCII) soubor s příponou *.html, případně *.htm
- startovní soubor, stránka: index.htm (index.html) - načte se jako první

Pravidla pro názvy html souborů

- vždy **bez mezer** (místo mezery použijte spodní podtržítko '_')
- **bez diakritických znamének** (ěščřýáíéúů)
- všechna **písmena malá** (většina webových serverů běží na systémech typu UNIX, které rozlišují malá a velká písmena a nepodporují češtinu)

Kódování češtiny

- **UTF-8**
- Windows-1250 (CP1250)
- ISO-8859-2

Zásady zpracování dokumentu

- **mezery**
- větší počet mezer se **redukuje na jednu**
- konec řádku (enter) se **převádí na mezeru**
- **prázdné řádky** (entery) se **ignorují**
- **řádky** – **zalamují se dynamicky** při vykreslování dokumentu (pokud to není implicitně upraveno)



Přednáška 1: Základy HTML a CSS

HTML v prohlížečích

HTML v prohlížečích

Webové prohlížeče (Chrome, Firefox, Edge, Internet Explorer, Opera, Safari,...) jsou programy, jejichž účelem je prezentovat dokument na **zobrazovacím zařízení** – nejčastěji monitoru počítače. Dokument je prohlížečem načítán a je prováděn jeho rozklad (syntaktická analýza) na jednotlivé elementy. Prohlížeč obsahuje tabulku značek, které podporuje.

Každému elementu je poté přiřazen **styl** (způsob zobrazení). Styly bývají uvedeny ve **stylovém předpisu** (souboru s příponou CSS). Vlastnosti stylů, které nejsou předepsány, doplní prohlížeč podle **implicitního stylu**, který má zabudován.

Novější prohlížeče pracují obecně ve dvou základních režimech:

- **Standardní režim** – režim snažící se dodržovat definované standardy;
- **Quirk mód** – režim zaměřený na zpětnou kompatibilitu, i pokud takové chování není v souladu se standardy.

To, jaký režim (a chování) prohlížeč zvolí, závisí především na uvedení **direktivy <!DOCTYPE**, protože většina starších stránek ji vůbec neobsahuje. Později toto chování částečně přebrala Mozilla/Firefox i Opera, kde však rozdíly mezi režimy nejsou natolik markantní. Existence nestandardního režimu je výsledkem vývoje, kdy si výrobci jednotlivých prohlížečů přizpůsobovali definici HTML podle svých potřeb a prohlížeče podporovaly nestandardní elementy a syntaxi.



Přednáška 1: Základy HTML a CSS

HTML elementy (1 / 2)

HTML elementy – rozdělení podle významu

Elementy lze z hlediska významu rozdělit na tři základní skupiny:

Strukturální elementy

- Rozvrhují strukturu dokumentu, příkladem jsou odstavce (**<p>**), nadpisy (**<h1>**, **<h2>**). Dodávají dokumentu formu.

Popisné (sémantické) elementy

- Popisují povahu obsahu elementu, příkladem je nadpis (**<title>**), adresa (**<address>**) nebo článek (**<article>**). Současný trend je orientován právě na sémantické značky, které usnadňují automatizované zpracovávání dokumentů a vyhledávání informací v záplavě dokumentů na webu. Vyvrcholením této snahy je v současné době jazyk HTML5.

Stylistické elementy

- Určují vzhled elementu při zobrazení, typickým příkladem je značka pro tučné písmo (****). Od tohoto druhu značek se postupně upouští, trendem je používání **kaskádových stylů**, které vzhled popisují odděleně od obsahu dokumentu. Mezi důvody pro neužívání těchto značek patří především to, že tyto značky jsou orientovány na prohlížení na obrazovce počítače, příliš se však nepočítá s používáním dokumentu jiným způsobem – alternativní prohlížeče pro postižené (čtečky pro slepce), v mobilních zařízeních a podobně. Kaskádové styly umožňují definovat rozdílné zobrazení pro různá zařízení.



Přednáška 1: Základy HTML a CSS

HTML elementy (2 / 2)

HTML elementy – rozdělení podle typu formátování

Z hlediska formátování dokumentu existují tři základní druhy elementů: blokové, řádkové a nahrazované.

Blokové elementy

Jsou to elementy, které tvoří nějaký blok. Zjednodušeně to znamená, že po takovém elementu je text dokumentu zalomen, odřádkován. Blokovými elementy jsou například **<h1>** pro nadpis nebo **<p>** pro odstavec.

Řádkové (inline) elementy

Jsou to elementy, které se nacházejí uvnitř textu, nedochází po nich k zalomení. Obvykle plní funkci zvýraznění nějaké části textu. Je to například **<a>** pro hypertextový odkaz.

Nahrazované elementy

Ty jsou nahrazeny nějakým obsahem, pro začlenění dokumentu jsou důležité jejich rozměry. Například **** pro obrázek.



Přednáška 1: Základy HTML a CSS

CSS úvod

CSS (kaskádové styly) umožňují oddělit obsah webu uložený v HTML od jeho způsobu zobrazení (formátu, stylu). Obsah zůstává vložen pomocí HTML kódu, veškerá jeho prezentace (písma, barvy, pozadí, okraje, formátování textu, efekty pro odkazy apod.) však zajišťuje CSS.



Interní zápis CSS

```
<head>
<title>Titulek stránky</title>
<style type="text/css">
CSS obsah
</style>
</head>
<body>
```

Externí stylový soubor

```
<link rel="stylesheet"
type="text/css"
href="stylesheet.css" />
```

Inline zápis CSS

```
<p style="color:
#ff0000;">Červený
text</p>
```

Řazení kaskádování

1. Inline stylový zápis (uvnitř HTML elementu)
2. Interní zápis CSS (uvnitř značky <head>)
3. Externí stylový soubor

Jazyk byl navržen standardizační organizací [W3C](https://www.w3.org/), autorem prvotního návrhu byl [Håkon Wium Lie](#). Byly vydány zatím dvě úrovně specifikace CSS1 a CSS2, dne 7. června 2011 byla dokončena revize CSS 2.1 a pracuje se na verzi CSS3 (některé části jsou již standardizovány, viz.

<https://www.w3.org/Style/CSS/current-work.en.html>).



Přednáška 1: Základy HTML a CSS

CSS syntaxe

Syntaxe CSS se odlišuje od syntaxe HTML. Skládá se za tři částí:

```
selektor {vlastnost: hodnota;}
```

Příklad:

```
body {  
  background: #eeeeee;  
  font-family: "Trebuchet MS", Verdana, Arial, serif;  
}
```

Dědění

Pokud je element součástí jiného (má svůj nadřazený (rodičovský) element), vnořený element zdědí vlastnosti přiřazené nadřazenému elementu. Do doby, než jsou vnořenému elementu vlastnosti cíleně změněny. Dědí se pouze vlastnosti, které může vnořený element použít (blokové vs. řádkové elementy).

Kombinování selektorů

```
h1, h2, h3, h4, h5, h6 {  
  color: #009900;  
  font-family: Georgia, sans-serif;  
}
```

Komentáře

```
/* toto je komentar*/
```




Přednáška 1: Základy HTML a CSS

CSS třídy a identifikátory

Třídní selektor umožňuje stylovat stejné HTML elementy různě. Název třídy je v CSS vždy uvozen tečkou.

Příklad:

```
.greenboldtext {  
  font-size: small;  
  color: green;  
  font-weight: bold;  
}
```

HTML kód:

```
<p>  
Lorem ipsum dolor sit amet, <span  
class="greenboldtext">consectetur</span> adipiscing  
elit.  
</p>
```

Identifikátory jsou podobné třídám s tím rozdílem, že pokud je jednou identifikátor v HTML dokumentu deklarovaný, **nemůže být již v tomtéž dokumentu deklarován znovu**. Jménu identifikátoru vždy v CSS předchází znak `#`;

Příklad:

```
#container {  
  width: 80%;  
  margin: auto;  
  padding: 20px;  
  border: 1px solid #666;  
  background: #ffffff;  
}
```

HTML kód:

```
<div id="container">  
Veškerý obsah  
stránky.  
</div>
```



Přednáška 1: Základy HTML a CSS

CSS vnější a vnitřní okraje, rámeček

Vnější okraje = vlastnost **margin**
Vnitřní okraje = vlastnost **padding**
Rámeček = vlastnost **border**

Příklad:

```
p {  
    padding: 10px; border: 1px solid black;  
    margin: 20px;  
}
```





Přednáška 1: Základy HTML a CSS

CSS vybrané vlastnosti textu a písma

Barva (color)

- Názvy (red, black...)
- Hexadecimální čísla (#ff0000, #000000)
- RGB kódování barev (rgb(255, 0, 0), rgb(0, 0, 0))

Příklad:

```
span {color: blue;}
```

Zarovnání textu (text-align)

- Hodnoty: left, right, center, justify

Příklad:

```
p {text-align: justify;}
```

Písmo

Rodina písma (font-family)

- Hodnoty: název rodiny písma, obecná rodina písem

Příklad:

```
h1 {  
    font-family: Arial,  
    Verdana, sans-serif;  
}
```

Velikost písma (font-size)

- Hodnoty: názvy velikostí, číslo + jednotky (px, pt, em, %)

Příklad:

```
* {font-size: 12px;}
```

Tučnost písma (font-weight)

- Hodnoty: názvy, číslo

Příklad:

```
div#text {  
    font-weight: bold;  
}
```



Přednáška 1: Základy HTML a CSS

CSS vybrané složitější typy selektorů

Univerzální selektor (universal selector)

- vztahuje se na všechny elementy, lze kombinovat s kontextovým selektorem

Příklad:

```
* {font-family: Arial;}  
p * {color: blue;}
```

Kontextový selektor (potomek, descendant selector)

- všechny elementy B, které jsou uvnitř elementu A

Příklad:

```
p span {  
    color: green;  
}
```

Selektor dítěte (child selector)

- element B je přímým potomkem elementu A (dítětem)

Příklad:

```
.left > p {  
    font-weight: bold;  
}
```

Selektor sousedících sourozenců (následník, adjacent selector)

- vybere element pouze pokud se vyskytuje bezprostředně za jiným elementem

Příklad:

```
h1 + h2 {font-family: Arial;}
```

Vícenásobné třídy

- dva a více názvů tříd

Příklad:

```
.left.text {  
    font-size: 14px;  
}
```



Přednáška 1: Základy HTML a CSS

CSS pravidla kaskádování

Pravidla kaskádování

1. Přednastavený vzhled

„co styly výslovně neupravují, to zůstává“

2. Dědičnost

- podřízený element přejímá formátování nadřazeného element

Příklad:

```
body {color: blue;}
```

3. Pozdější vyhrává

- platí ten zápis, který je uveden nejpozději

Příklad:

```
H1 {color: black;}
```

```
H1 {color: red;}
```

4. Priorita vnitřního elementu

- vyhrává ten element, který je "vnitřnější", je blíže formátovanému obsahu

Příklad:

```
<span style="color: red;">  
  <span style="color: green;">  
    obsah</span>  
</span>
```