

- význam, princip
 - výhody používání (5)
 - syntax, základní pojmy
 - umístění definice stylů
-
- třídy a identifikátory
 - kaskádování (překrývání), priorita, !important
 - kontextové selektory
 - pseudotřídy
-
- délkové jednotky (5), funkce
 - zápis barvy
-
- box model
 - extra display (4), position (5)
 - využití tagů div a span

17. CSS - použití, typy selektorů, příklady

Detailně

= **Cascading Style Sheets** - jazyk pro popis způsobu zobrazení webových stránek, napsaných převážně v HTML. V současnosti verze CSS3, která přinesla např. průhlednost, stíny, kulaté rohy a nové selektory.

výhody používání stylů

- Jednoduchý způsob, jak stránky hezky a moderně zformátovat, mnoho vlastností nejde v základním HTML měnit.
- Oddělují vzhled dokumentu od jeho struktury a obsahu.
- Používání tříd pro formáty, které se v dokumentu opakují, šetří čas.
- U složitějších stránek budou všechny vypadat podobně, když použijeme stejný stylový soubor.
- Umožňují responzivitu stránek přes rozmístění grid nebo flexbox a @media pravidla.
@media (min-width: 576px) { ... }

syntaxe stylů, možnosti umístění definice stylů, připojení css souboru k html stránce

selektor {vlastnost: hodnota; vlastnost2: hodnota;}

- selektor určuje, co má být formátováno, více selektorů se odděluje čárkou (můžeme formátovat více elementů najednou)
- ve složených závorkách se nachází vlastnosti a jejich hodnoty
- mezi vlastností a hodnotou je dvojtečka
- vlastnosti jsou oddělené středníkem

Umístění

1. **Inline style** jsou definovány uvnitř HTML tagů v atributu style a mají nejvyšší prioritu. Zhoršují čitelnost kódu, špatně se zpětně upravují. Vhodné na testování.
2. **Interní style** jsou definovány v hlavičce HTML dokumentu v tagu <style>. Mají nižší prioritu než inline style.

3. **Externí styly** jsou definovány v samostatném CSS souboru a do HTML se připojují tagem `<link rel="stylesheet" href="soubor.css" type="text/css">` v hlavičce. Obecně nejvhodnější způsob.

třídy a identifikátory

Třídy a identifikátory jsou způsob, jak označit prvky HTML dokumentu pro CSS nebo JS.

Atribut **class** (třída) se může použít u libovolného elementu (tagu). Tento element se poté zformátuje podle definice za selektorem `.navez-tridy` (tečka aby se odlišila třída od tagu). Element se stejnou třídou se v dokumentu může vyskytovat vícekrát a jeden element může mít i více tříd.

Atribut **id** slouží pro jednoznačný popis nějakého elementu - v jednom dokumentu by se měl určitý identifikátor vyskytnout pouze jednou. Jinak funguje stejně jako třída, akorát selektor nezačíná tečkou, ale dvojkřížkem `#`.

překrývání stylů a slučování definic

Na jeden element se může navrstvit mnoho deklarací stejné vlastnosti, někdy i protichůdných - styly se poté překrývají (**kaskádují**). Při překrývání stylů má prvek přidělenou vlastnost s nejvyšší prioritou. Priorita závisí v pořadí na:

- specifitě selektoru (`p.class > .class`)
- umístění (`inline > interní a externí CSS`)
- pořadí definice (pozdější styl přepisuje dřívější)

Výjimkou je deklarace obsahující řetězec **"!important"**. Taková potom nebude přepsána.

kontextové selektory, pseudotřídy

Kontextové selektory umožňují vybírat elementy na základě ostatních elementů nebo jejich pozice v dokumentu (**DOM**).

- `mezera` potomek
- `>` přímý potomek
- `+` element bezprostředně následující
- `~` sourozenec
- `tag[atr="hodnota"]` element s určitou hodnotou atributu

Pseudotřídy umožňují definovat styly na základě stavu nebo vlastnosti elementu. Např.: `:hover`, `:focus`, `:nth-of-type()`, `:not()`, `:link`, `:visited`, `:active`

délkové jednotky

Absolutní jednotky

- **px**
- **pt** (typografický bod)
- **mm, cm, in** - vhodné jen pro tisk
- **rem** (em kořenového prvku) - nemění se tedy podle nadřazených prvků, na jednom operačním systému/prohlížeči stejná (šířka M)

Relativní jednotky

- **em** - výška jednoho řádku ovlivněná nadřazenými prvky - mění se podle velikosti písma; v základu šířka "M"
- **procenta %** - většinou z šířky rodičovského elementu
- **procenta velikosti prohlížeče**
 - `vw` (1 % dostupné šířky prohlížeče), `vh`
 - `vmin` (1 % kratšího rozměru prohlížeče), `vmax`

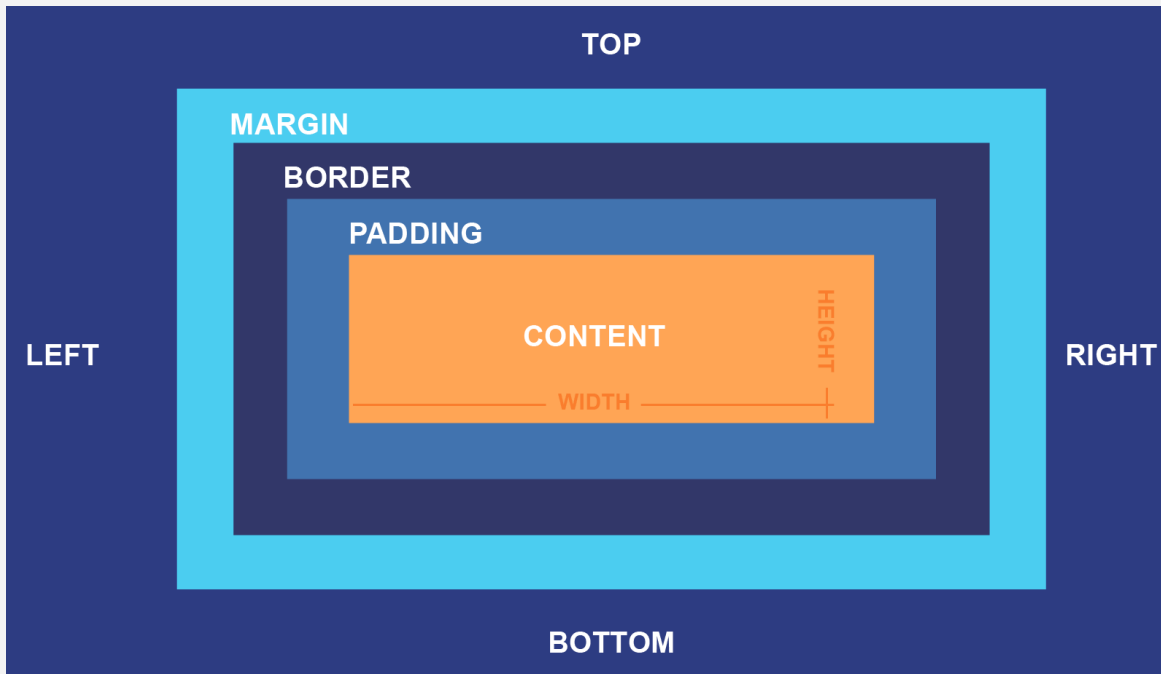
Mezi hodnotou a jednotkou se nedělá mezera. Lze použít různé kombinace ve funkcích `calc()`, `min()`, `max()`, `clamp()`

Zápis barvy

1. **slovně** - red, green, blue...
2. **hexadecimálně** - #ff0000
3. funkcemi rgb a rgba
 - a. **dekadicky** - rgb(255, 0, 0)
 - b. **procenty** - rgb(100%, 0%, 0%)

schéma blokového elementu

Box model:



display:

- **block** - prvek se zobrazí jako blok, to jest s řádkovým zlomem na konci
- **inline** - prvek se zobrazí jako kus řádku, to jest bez řádkového zlomu na konci
- **inline-block** - prvek se zobrazí jako kus řádku bez zlomu na konci, jdou mu nastavit rozměry
- **none** - skrytý prvek (jako by neexistoval)

position:

- **static** - defaultní hodnota
- **relative** - můžeme posunout relativně k originální pozici a okolní text počítá s jeho originální pozicí.
- **absolute** - můžeme umístit relativně k prvnímu umístěnému předkovi (pokud takový není tak k oknu) a okolní text se chová, jako by takového objektu vůbec nebylo.
- **fixed** - objekt zůstává na stejném místě v okně, i když se se stránkou roluje. Základ souřadného systému je vždy okno prohlížeče, bez ohledu na to, kam je prvek v kódu umístěn.

využití tagů div a span

Div a span jsou neutrální tagy, které nenesou význam ani formát. Byly přidány a jsou užitečné v kombinaci s CSS a pro organizaci kódu. Využívají se s třídami a identifikátory.

Div je blokový - na obalení logických sekcí stránky, jejichž prvky mohou mít společný formát.

Span je řádkový - často se používá jako paragraf bez otravných mezer nad a pod nebo při úpravě jen určité části textu.