

# Možnosti pozicování na webových stránkách

Blokové a inline prvky, absolutní a relativní pozicování, obtékání, grid, flex, tabulka

## Blokové prvky

- Základními stavebními kameny obsahu v HTML jsou tzv. **blokové prvky**
- Základními stavebními kameny obsahu v HTML jsou tzv. **blokové prvky** nebo **inline prvky**
- Blokem se rozumí HTML prvky, který je ukončen znakem pro konec řádku (standardně nelze zobrazit více bloků na jeden řádek)
  - `<P>`
    - Odstavec – základní blok obsahující převážně text
    - Párová značka
    - Odstavce jsou od sebe oddáleny vertikální mezerou pro lepší přehlednost
    - `<p>Tento odstavec obsahuje jen jednu větu.</p>`
  - `<H1>`, `<H2>`, `<H3>`, ...
    - Nadpisy 1. až 6. úrovně – graficky odlišeny velikostí písma
    - Praktické využití mají jen nadpisy do 4. úrovně
    - Párová značka
    - Stejně jako odstavce jsou od sebe oddáleny vertikální mezerou
    - `<h1>Hlavní nadpis stránky (vel. 24 px)</h1>`  
`<h2>Podnadpis, kapitoly (vel. 18 px)</h2>`  
`<h3>Mezinadpisy (vel. 16 px)</h3>`  
`<h4>Důležité odstavce (vel. 14 px)</h4>`
  - `<DIV>`
    - Oddíl – zahrnuje libovolnou část HTML obsahu
    - Párová značka
    - Na rozdíl od odstavců a nadpisů nejsou nijak přeformátovány
    - Využívají se při rozdělování HTML dokumentu do několika nezávislých částí, které lze poté formátovat odděleně.
    - `<div>`  
`<h1>Nadpis</h1>`  
`<p>Tento odstavec obsahuje jen jednu větu.</p>`  
`</div>`
  - `<BR>`
    - Řádkový zlom – vloží zalomení řádku (tzv. měkký Enter)
    - Nepárová značka
    - Na rozdíl od odstavce není oddálen vertikální mezerou
    - `<p>Text se zobrazí na první řádce,<br>`  
`toto už bude na druhé řádce,<br>`  
`ale pořád je to jeden odstavec</p>`

## Inline prvky

- Řádkové prvky nevkládají na konec znak pro konec řádku, tudíž je lze zobrazovat v libovolném množství na jednom řádku
- Existuje jich celá řada, ale dnes se používají jen některé (jsou nahrazeny kaskádovými styly)
  - `<STRONG>`

- Silné zvýraznění – zvýrazní označenou část textu tučným písmem
- Párová značka
- Pokud chcete silně zvýraznit celý blokový prvek (celý nadpis nebo odstavec), nahradte tuto značku CSS vlastností **font-weight**
- `<p>Normální text <strong>silně zvýrazněný text</strong>.</p>`
- **<EM>**
  - Zvýraznění – zvýrazní text kurzívou
  - Párová značka
  - Pokud chcete zvýraznit kurzívou celý blokový prvek (celý nadpis nebo odstavec), nahradte tuto značku CSS vlastností **font-style**
  - `<p>Normální text <em>zvýrazněný text kurzívou</em>.</p>`
- **<B>**
  - Tučné písmo – označení části textu tučným písmem
  - Zastaralé, doporučuje se nahradit značkou `<strong>`
  - Párová značka
  - `<p>Normální text <b>tučný text</b>.</p>`
- **<I>**
  - Písmo kurzíva – označení části textu kurzívou
  - Zastaralé, doporučuje se nahradit značkou `<em>`
  - Párová značka
  - `<p>Normální text <i>zvýrazněný text kurzívou</i>.</p>`
- **<SUB>**
  - Dolní index (subscript) - zmenšení a snížení textu (chemické vzorce)
  - Párová značka
  - `<p>Kyselina sírová - H<sub>2</sub>SO<sub>4</sub>.</p>`
- **<SUP>**
  - horní index (superscript) - zmenšení a zvýšení textu (mocniny)
  - párová značka
  - `<p>Rozloha ČR je 78 866 km<sup>2</sup>.</p>`
- **<SPAN>**
  - Obálka – zahrnuje libovolnou část textu, obrázků, tabulek
  - Má stejné využití jako `div`, jen se řadí mezi řádkové prvky
  - Párová značka

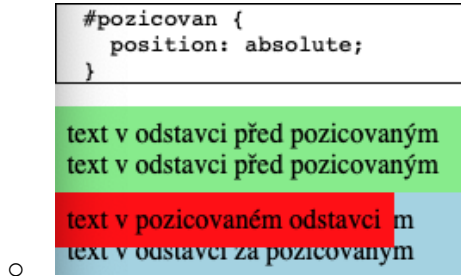
## Statické pozicování

- Je to přirozené pozicování elementu (umístění normálně v dokumentu)
- Výchozí hodnota všech elementů (smysl pouze pro přebíjení dříve přenastavených hodnot)
- Vlastnosti `top`, `left`, `right`, a `bottom` jsou ignorovány

## Absolutní pozicování

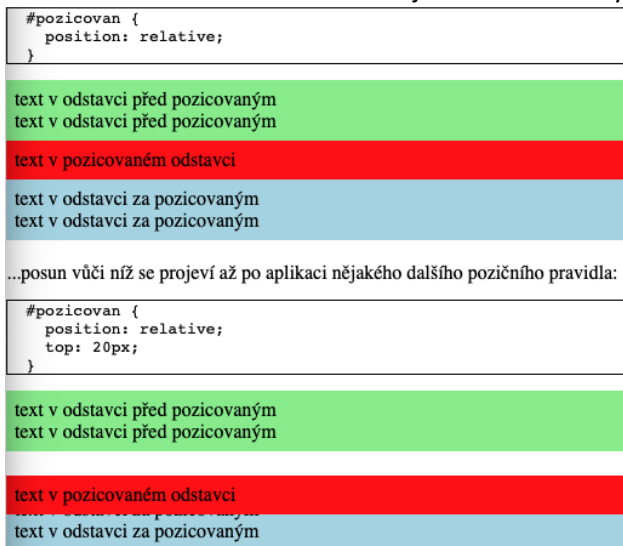
- Automaticky vyjmut z posloupnosti elementů ve stránce

- Na výstupu se to projeví tak, že elementy následující po něj zaujmou jeho místo v základním vykreslování
- Přitom samotný takto zapozicovaný element zůstane jakoby na svém původním místě, pouze se přesune nad všechny ostatní elementy ve stránce (jeho z-souřadnice je zvýšená nad všechny ostatní dosud zavedené)
- Navíc se „smrskne“ na šířku a místo základního místa od kraje ke kraji zabírá pouze tolik, kolik jeho obsah skutečně potřebuje.



## Relativní pozicování

- Zvětšení z-ové souřadnice nad všechny ostatní a na aktuální pozici (na které je zobrazen při základním vykreslování) se mu přiřadí nová, relativní soustava souřadnic (svázaná s levým horním rohem tohoto elementu v jeho základním vykreslení)



- **Fixed:**
  - To způsobí, že objekt zůstává na stejném místě v okně, i když se stránka roluje. Základ souřadného systému je vždy okno prohlížeče, bez ohledu na to, kam je prvek v kódu umístěn
- **Sticky:**
  - Pokud je na stránce, tak se chová normálně, ale poté co zmizí z viewpotu (z dohledu), tak se nalepí na vršek okna

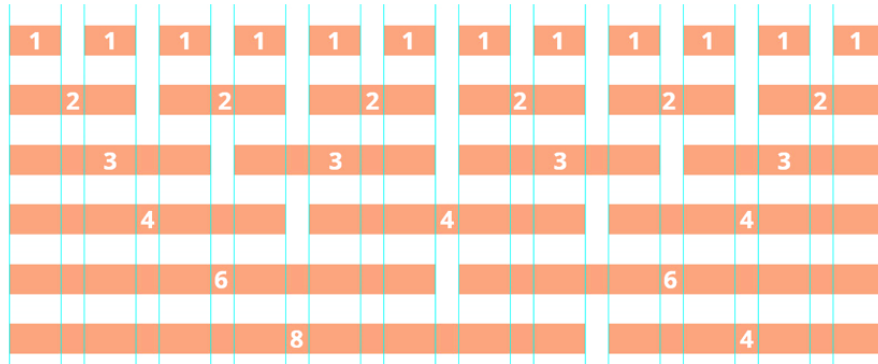
## Obtékání

- Lokální úprava textu – obtékání obrázku

- Globální layout (uspořádání stránky) - rozmístění celých bloků, bez použití tabulek, či pozicování
- Vždycky obsahuje:
  - Obtékaný objekt
  - Obtékající obsah
- Obtékaný objekt musí mít nastavenou šířku, protože kdyby jí neměl, tak bude mít šířku 100% a okolní text nemá kudy téci
- **Zprava (float: right)**
  - Umístí objekt doprava, takže ho text zleva obtéká
- **Zleva (float: left)**
  - Umístí objekt doleva, takže je obtékaný zprava

## Grid

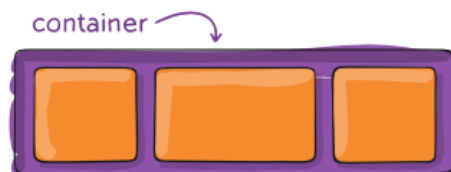
- Bylo vyvinuto z potřeby udělat mřížku bez elementu <table> (kvůli nedostatkům)
- Grid systémy:
  - Předpřipravené třídy, které přiřazujeme např. elementům <div> a ty následně získávají určitou šířku.
  - Systémy obvykle fungují tak, že rodičovský element rozdělí do 12 sloupců
  - Dříve se k tomu používala šířka zadaná v procentech, jeden sloupec tedy zabíral 8,33%
  - Dnes se využívá **flexbox**
  - Číslo 12 bylo vybráno kvůli jeho mnoha dělitelům (12, 6, 4, 3, 2, 1)



- 
- Pokud chceme systém používat, tak jsou dvě možnosti:
  - CSS Framework Bootstrap
  - Framework Flexbox Grid

## Flex

- **Flex container**



- 
- Display:
  - Definuje container (buď inline nebo block) pro všechny itemy

```
.container {
  display: flex; /* or inline-flex */
}
```

- Flex-direction

- Určuje, zda jsou itemy v řádcích nebo sloupcích

```
.container {
  flex-direction: row | row-reverse | column |
  column-reverse;
}
```

- `row` (default): left to right in `ltr`; right to left in `rtl`
- `row-reverse`: right to left in `ltr`; left to right in `rtl`
- `column`: same as `row` but top to bottom
- `column-reverse`: same as `row-reverse` but bottom to top

- Flex-wrap

- Defaultně se itemy snaží vejít na jeden řádek, ale dá se to díky `flex-wrap` změnit

```
.container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

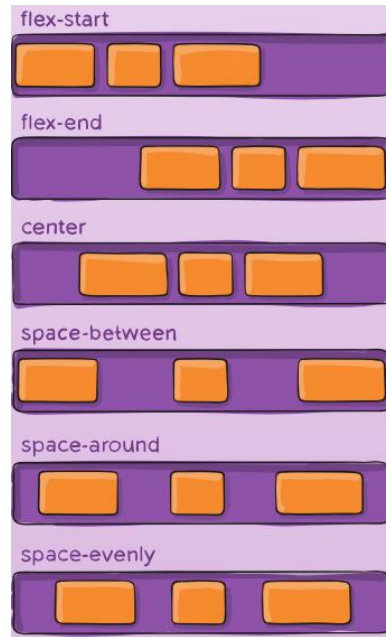
- `nowrap` (default): all flex items will be on one line
- `wrap`: flex items will wrap onto multiple lines, from top to bottom.
- `wrap-reverse`: flex items will wrap onto multiple lines from bottom to top.

- Flex-flow

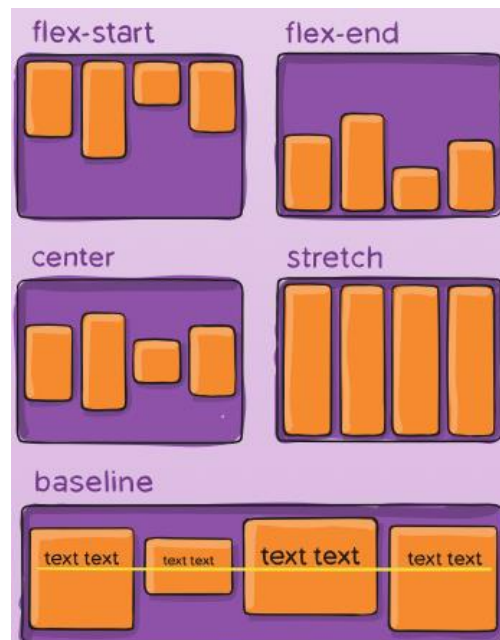
- Zkratka pro `flex-direction` a `flex-wrap`

```
.container {
  flex-flow: column wrap;
}
```

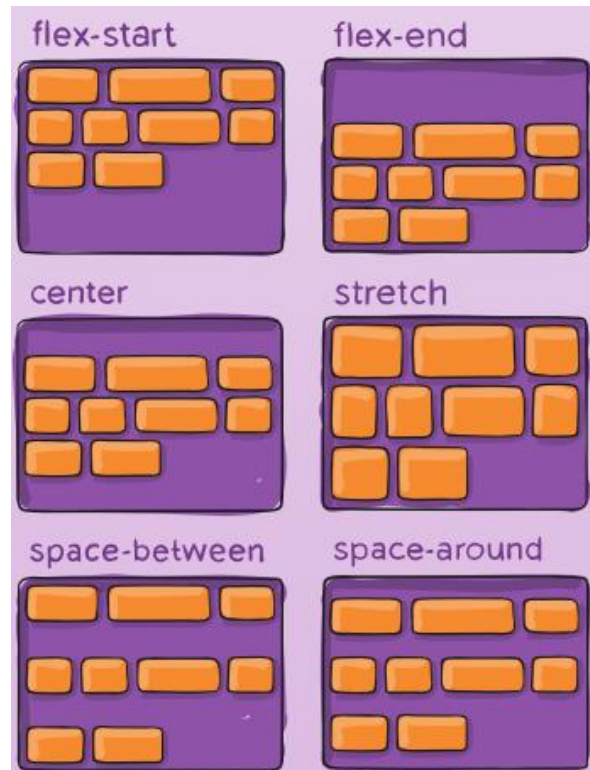
- Justify-content



- Align-items



- Align-content



- **Flex items**

- Order

By default, flex items are laid out in the source order. However, the `order` property controls the order in which they appear in the flex container.

```

.item {
  order: 5; /* default is 0 */
}
  
```

- Flex-grow





This defines the ability for a flex item to grow if necessary. It accepts a unitless value that serves as a proportion. It dictates what amount of the available space inside the flex container the item should take up.

If all items have `flex-grow` set to 1, the remaining space in the container will be distributed equally to all children. If one of the children has a value of 2, the remaining space would take up twice as much space as the others (or it will try to, at least).

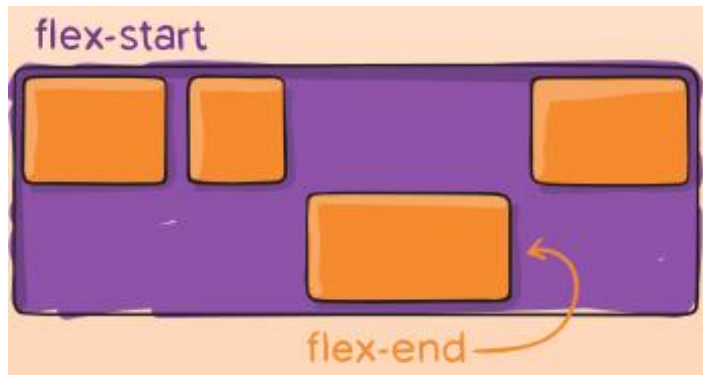
```
.item {
  flex-grow: 4; /* default 0 */
}
```

- 
- Flex-shrink
  - Možnost se scvrknout
- Flex-basis
  - Základní velikost
- Flex

This is the shorthand for `flex-grow`, `flex-shrink` and `flex-basis` combined. The second and third parameters (`flex-shrink` and `flex-basis`) are optional. The default is `0 1 auto`, but if you set it with a single number value, it's like `1 0`.

```
.item {
  flex: none | [ <'flex-grow'> <'flex-shrink'>
  ? || <'flex-basis'> ]
}
```

- 
- Align-self



■

## Tabulka

- Tabulka nám umožňuje do jejích buněk vkládat elementy, které jsem po tom hezky uspořádané
- Na rozdíl od odstavců, které se vždy skládají pod sebe, si můžeme text v tabulce skládat úhledně vedle sebe
- V buňkách mohou být kromě textu samozřejmě i obrázky a další elementy
- To se může hodit například k zobrazování nějakých výsledků, parametrů nebo statistik
  - Tabulka o 3 sloupcích a 2 řádcích

```
<table border="1">
  <tr>
    <td>Buňka 1</td>
    <td>Buňka 2</td>
    <td>Buňka 3</td>
  </tr>
  <tr>
    <td>Buňka 4</td>
    <td>Buňka 5</td>
    <td>Buňka 6</td>
  </tr>
</table>
```

Buňka 1	Buňka 2	Buňka 3
Buňka 4	Buňka 5	Buňka 6




- Tabulku uzavřeme mezi párový tag <table>
- Každý řádek uzavřeme do párového tagu <tr> (jako table row)
- Jednotlivé buňky jsou poté v řádku obalené tagem <td> (jako table data)
- Tabulka s hlavičkou
  - Může mít i složitější strukturu
  - Hlavička <thead> a potom <tbody>
  - Hlavička znamená první řádek, který označuje, co se ve sloupečku nachází
  - Za Hlavičkou může následovat tfoot (patička tabulky)
  - V hlavičce píšeme také <tr>, ale místo <td> píšeme <th> (table header)

```

<table border="1">
  <thead>
    <tr>
      <th>Náhled</th>
      <th>Typ</th>
      <th>Procesor</th>
      <th>Graf. karta</th>
      <th>Skladem</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td></td>
      <td>AB8AC9</td>
      <td>Intel Atom</td>
      <td>Nvidia</td>
      <td>Ano</td>
    </tr>
    <tr>
      <td></td>
      <td>GS8DGF</td>
      <td>AMD</td>
      <td>ATI</td>
      <td>Ano</td>
    </tr>
    <tr>
      <td></td>
      <td>KG1862A</td>
      <td>Neuvedeno</td>
      <td>Neuvedeno</td>
      <td>Ne</td>
    </tr>
  </tbody>
</table>

```

•

Náhled	Typ	Procesor	Graf. karta	Skladem
	AB8AC9	Intel Atom	Nvidia	Ano
	GS8DGF	AMD	ATI	Ano
	KG1862A	Neuvedeno	Neuvedeno	Ne

•