

Webové technológie 4

Aplikácie Webových technológií

Peter Bednár

Vlastnosti písma (1)

- **font-family** – font písma
 - Je možné zadať všeobecný typ – pätkové **serif**, bezpätkové **sans-serif**, neproporcionálne (**monospaced**) alebo konkrétny názov písma, napr. *Arial*, *Times*, *Courier*, atď.
 - Viac alternatív oddelených čiarkou podľa preferencií, prehliadač zvolí najpodobnejší font, napr.:
font-family: Arial, Helvetica, sans-serif;
- **font-size** – veľkosť písma
 - Rozmerová jednotka (napr. **2em**) alebo preddefinované hodnoty **small**, **medium** (prednastavená hodnota), **large**
 - Nastavte referenčnú veľkosť písma v pixeloch pre koreňový element `<html>` a používajte relatívne rozmerové jednotky: **em**, resp. **rem**

Vlastnosti písma (2)

- `font-style` – štýl písma
 - `normal` alebo kurzíva `italic`
- `font-weight` – hrúbka písma
 - `lighter`, `normal`, `bold` atď. alebo číselná hodnota 100, 200, ..., 900 (`normal` zodpovedá 400, `bold` 700 atď.)
- `font-variant` – určuje, či sa malé písmená zobrazia normálne – `normal`, alebo ako kapitálky – `small-caps`

Vlastnosti textu (1)

- `color` – farba písma
- `text-align` – vodorovné zarovnanie textu v rámci nadradeného elementu
 - `left` – vľavo (prednastavená hodnota ak je smer textu zľava doprava), `right` – vpravo, `center` – centrovane, `justify` – do bloku
- `vertical-align` – zvislé zarovnanie textu
 - číselná jednotka pre posunutie hore (kladné číslo) alebo dole (záporné číslo) od základnej čiary, alebo `sub` – dolný index, `super` – horný index, `middle` – v strede nadradeného elementu
- `text-indent` – pred-sadenie (záporné hodnoty), resp. odsadenie prvého riadku textu – rozmerová jednotka (používajte pixely alebo relatívne jednotky em)

Vlastnosti textu (2)

- **text-decoration** – podčiarknutie **underline**, prečiarknutie **line-through**, čiara nad textom **overline**, blikanie **blink**, atď.
 - Niektoré dekorácie majú aj ďalšie podvlastnosti, napr. **text-decoration-color**
- **text-transform** – zmena zobrazenia všetkých písmen na malé **lowercase**, veľké **uppercase**, alebo prvé písmená slov veľké **capitalize**
- **white-space** – spôsob zobrazenia prázdnych znakov zapísaných v HTML dokumente
 - **normal** – zobrazia sa ako jedna medzera (prednastavené pre väčšinu elementov), alebo **pre** – zachovajú sa všetky medzery a konce riadkov ako v HTML dokumente (prednastavené pre element `<pre>`)

Vlastnosti zoznamov

- `list-style-type` – typ značky resp. číslovania
 - kruh `circle`, štvorec `square`, číslovanie arabskými číslicami `decimal`, číslovanie malými rímskymi číslicami `lower-latin`, atď.
 - `none` – nezobrazí sa žiadna značka
 - Umožňuje aj predefinovať zobrazenie usporiadaného zoznamu na neusporiadaný a naopak (nie je to vhodné kvôli významu obsahu)
- `list-style-image` – URL obrázka, ktorý sa zobrazí ako značka

Vlastnosti tabuliek

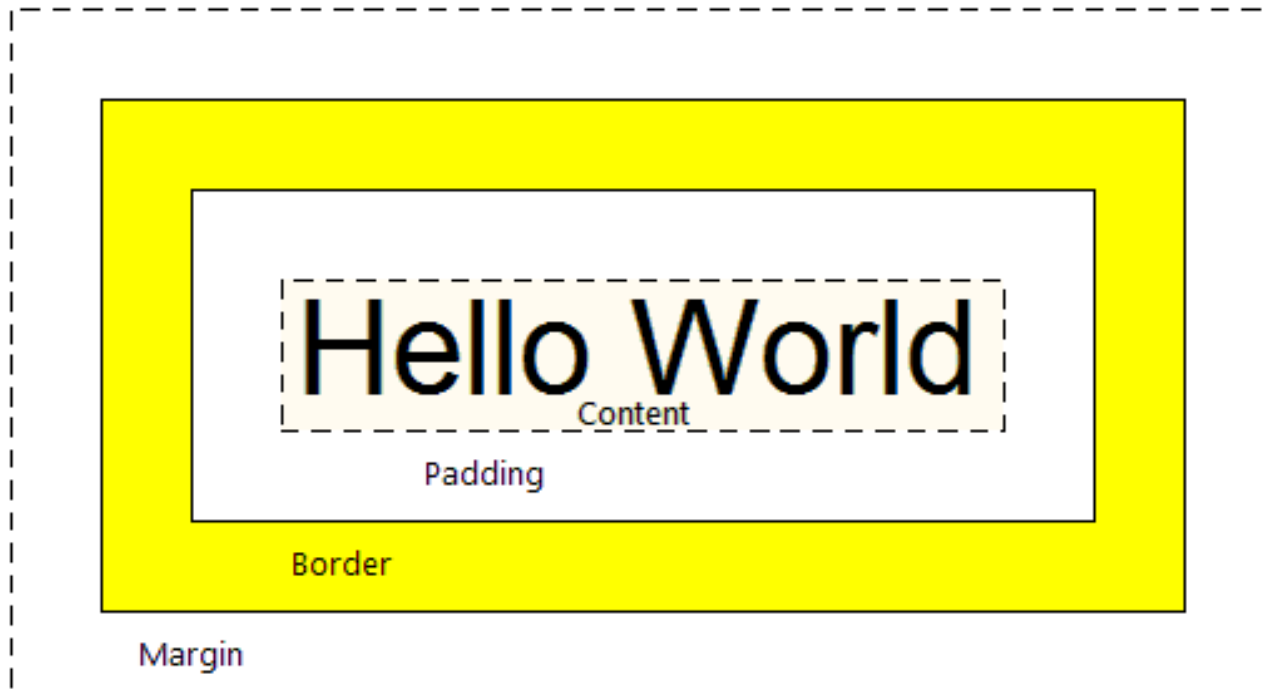
- **border-collapse** – udáva, či sa orámovanie zobrazí samostatne pre každú bunku (**separate** – prednastavená hodnota) alebo splynie do jedného okraja (**collapse**)

bunka 1	bunka 2
bunka 3	bunka 4

bunka 1	bunka 2
bunka 3	bunka 4

- **border-spacing** – udáva vzdialenosť medzi orámovaním buniek ak je zobrazené samostatne, rozmerové hodnoty X a Y (ak sú obe rovnaké, stačí zadať iba X, ako jednotky používajte pixely)
- **caption-side** – umiestnenie popisu tabuľky (element `<caption>`) nad tabuľkou **top** (preddefinovaná hodnota), alebo pod **bottom**

Ohraničenie a okraje elementov



Margin – vonkajšie odsadenie (okraj),
Border – ohraničenie,
Padding – vnútorné odsadenie

Veľkosť ohraničenia a okrajov

- `padding`-strana
- `margin`-strana
- `border`-strana-`width`
 - Nastavenie veľkosti pre jednotlivé strany `top`, `right`, `bottom`, `left`, hodnota je rozmerová jednotka, napr.:

```
padding-left: 15px;
```

- Skrátený zápis v poradí horná, pravá, dolná, ľavá
 - stačí uviesť jednu hodnotu ak je rovnaká pre všetky strany, alebo dve pre rovnaké hodnoty `top/bottom` a `left/right`

```
padding: 0 10px 5px 15px;
```

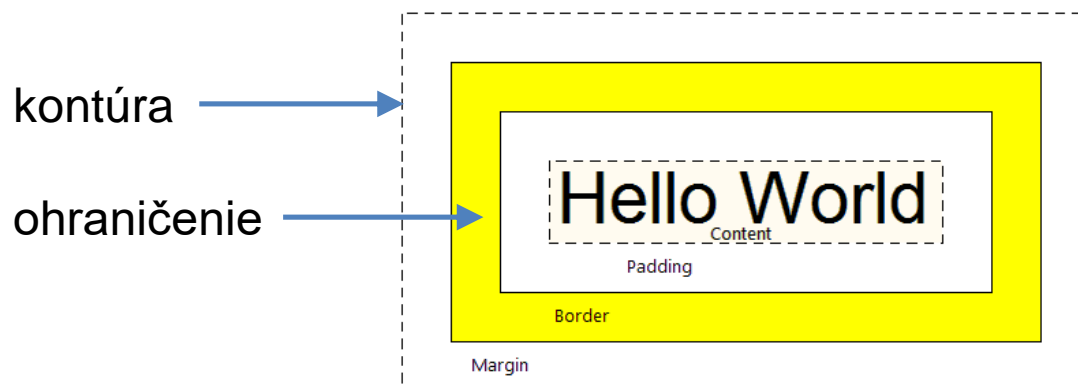
```
border-width: 1px;
```

Vlastnosti ohraničenia

- `border-strana-style`
 - žiadne ohraničenie – `none`, plná čiara – `solid`, bodkovaná čiara – `dotted`, čiarkovaná čiara – `dashed`, atď.
 - skrátенý zápis `border-style`
- `border-strana-color`
 - farba ohraničenia
 - skrátенý zápis `border-color`
- `border-roh-radius`
 - zobrazia sa oblé rohy s danou veľkosťou (polomerom), resp. vnútorný obsah elementu sa oreže podľa oblého ohraničenia (označenie rohov: `top-left`, `top-right`, `bottom-right`, `bottom-left`)
 - skrátенý zápis `border-radius: 10px;`

Zobrazenie kontúry

- Orámovanie ktoré sa zobrazí okolo elementu mimo ohraničenia cez vonkajší okraj alebo aj cez ďalšie elementy
- Používa sa napr. pri ladení zobrazenia stránky, aby bolo možné zobrazit' aj vonkajší okraj
- Vlastnosti:
 - `outline-width`, `outline-color`, `outline-style`
 - `outline-offset` – odsadenie od ohraničenia



Rozmery elementu

- **width**, **height** – šírka a výška elementu
 - hodnota je rozmerová jednotka (absolútna alebo relatívna k veľkosti nadradeného elementu, alebo *viewportu*), alebo prednastavená hodnota **auto** (tzn. veľkosť sa prispôsobí podľa obsahu)
- Pri hodnote **auto**, alebo relatívnych jednotkách sa môže veľkosť meniť podľa zmeny *viewportu* (pri zmene veľkosti okna, alebo orientácii zariadenia), pre elementy je možné definovať minimálnu a maximálnu veľkosť:
 - **width-min**, **height-min** – prednastavená hodnota 0
 - **width-max**, **height-max** – prednastavená hodnota bez obmedzenia **none**

Čo je zahrnuté do rozmerov

- Vlastnosť **box-sizing**
- **content-box** – (prednastavená hodnota), vlastnosti **height** a **width** (resp. **min**, **max-height**, **-width**) určujú veľkosť vnútorného obsahu (*padding* a *border* nie sú zahrnuté do nastavenia rozmerov)
 - Zobrazia sa väčšie elementy (pretože sa pridá šírka *padding* a *border*)
- **border-box** – nastavenie rozmerov zahŕňa vnútorné odsadenie a ohraničenie (*padding* a *border*)
 - Nastaví sa priamo celková veľkosť – vhodnejšie pre rozvrhovanie stránky

Dynamický výpočet rozmerov

- Pri nastavení rozmerov je možné využiť funkciu `calc()` ktorá dynamicky vypočíta rozmer podľa zadaného výrazu

- Príklady:

```
width: calc(80vw + 5em);
```

```
height: calc(100% / 6);
```

- Operácie `+`, `-`, `*`, `/`, pri násobení alebo delení musí byť jeden z operandov číslo bez jednotky, zložitejšie výrazy je možné uzatvoriť do zátvoriek, napr.

```
max-width: calc(100vh - (2rem + 0.1vw) * 4);
```

Presah obsahu

- Vlastnosť **overflow**
- **visible** – (prednastavená hodnota) ak je obsah väčší než rozmery elementu, zobrazí sa cez susedné elementy
- **hidden** – zobrazený obsah sa oreže podľa rozmerov elementu
- **scroll** – vždy sa pridajú posuvné lišty a používateľ môže zobrazit' obsah skrolovaním
- **auto** – posuvné lišty sa zobrazia podľa potreby ak bude obsah presahovať mimo aktuálne rozmery elementu

Viditeľnosť

- **visibility**
 - **visible** – (prednastavená hodnota) element sa zobrazí
 - **hidden** – element nie je zobrazený, ale vyhradí sa pre neho na stránke prázdne miesto
 - **collapse** – rovnako ako hidden pre riadky a stĺpce tabuliek alebo ich zoskupenia (`<colgroup>`, `<thead>`, atď.)
- **display: none;**
 - Element sa nezobrazí a nie je pre neho ani vyhradené miesto

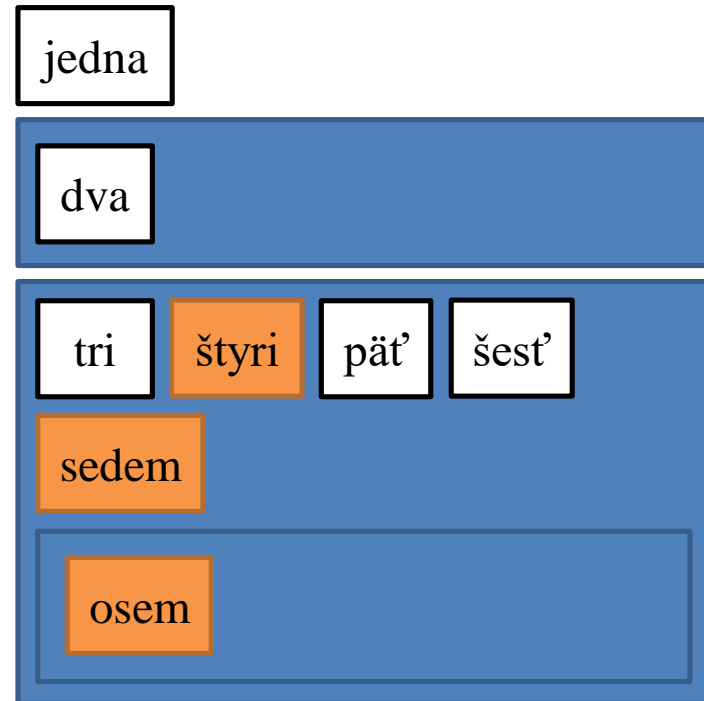
Umiestnenie elementov (1)

- Štandardne sa elementy rozdeľujú na dva základné typy:
 - **blokové**: <div>, <p>, <h1>-<h6>, <pre>, , , , <table> ...
 - **riadkové**: , <i>, , <a>, ...
- Pri zobrazovaní umiestni prehliadač elementy v poradí v akom sú uvedené v HTML dokumente, pričom:
 - Blokové elementy sa umiestnia samostatne vždy na nový riadok
 - Riadkové elementy sa umiestnia vedľa seba v riadku zľava doprava, ich výška a šírka sa prispôsobia obsahu a nie je možné ich predefinovať (ak sa už nezmestia na riadok, umiestnia sa na začiatok nového riadku)

Umiestnenie elementov (2)

- Text sa zobrazuje ako riadkový element

```
jedna  
<div>  
  dva  
</div>  
<div>  
  tri  
  <span>štyri</span>  
  päť šesť<span>sedem</span>  
  <div>  
    <span>osem</span>  
  </div>  
</div>
```



Vlastnosť `display`

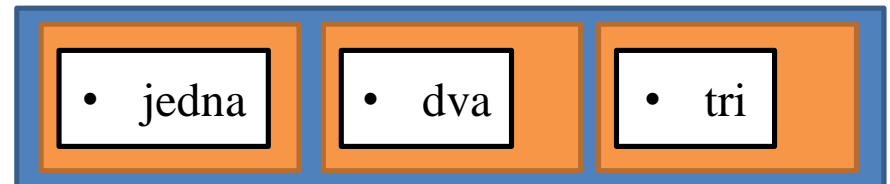
- Štandardné umiestnenie je možné zmeniť predefinovaním vlastnosti `display` pre daný element
- Základné hodnoty:
 - `block` – umiestnenie ako blokový element
 - `inline` – umiestnenie ako riadkový element (rozmery sa prispôbia obsahu a nie je možné ich zmeniť v CSS)
 - `inline-block` – umiestnenie ako riadkový element, ale je možné zmeniť rozmery elementu (šírku a výšku)

Príklad - `display: inline-block`

```
<style>
  li {
    display: inline-block;
  }
</style>
```

```
...
<ul>
  <li>jedna</li>
  <li>dva</li>
  <li>tri</li>
</ul>
```

Štandardne sú `` elementy blokové a preto sa zobrazia v zozname pod seba, po predefinovaní sa zobrazia v riadku



Rozvrhovanie stránky - Flexbox

- Flexibilnejší spôsob rozvrhovania elementov na stránke
- Je potrebné vytvoriť **flex kontajner** – nadradený element, pričom bude možné flexibilne usporiadať potomkov
 - Kontajner je vytvorený nastavením vlastnosti **display: flex;**
- Základná štruktúra:

```
<div style="display: flex;">
  <div>text</div>
  <div>text</div>
  ...
</div>
```

————— Flex kontajner

|————— Vnorené elementy
(potomkovia), ktoré
možno flexibilne
umiestniť

Vlastnosti Flex kontajnera (1)

- **flex-direction**
 - **row** (prednastavená hodnota) – umiestnenie potomkov v riadku (podobne ako pri `inline-block`)
 - **column** – umiestnenie pod sebou na nových riadkoch
 - **row-reverse**, **column-reverse** – opačné poradie než je usporiadanie elementov v HTML dokumente
- **flex-wrap**
 - **wrap** – ak sa potomkovia nezmestia na riadok, umiestnia sa na nový (zachová sa veľkosť potomkov)
 - **no-wrap** (prednastavená hodnota) – veľkosť potomkov sa zmenší tak aby sa zmestili na jeden riadok

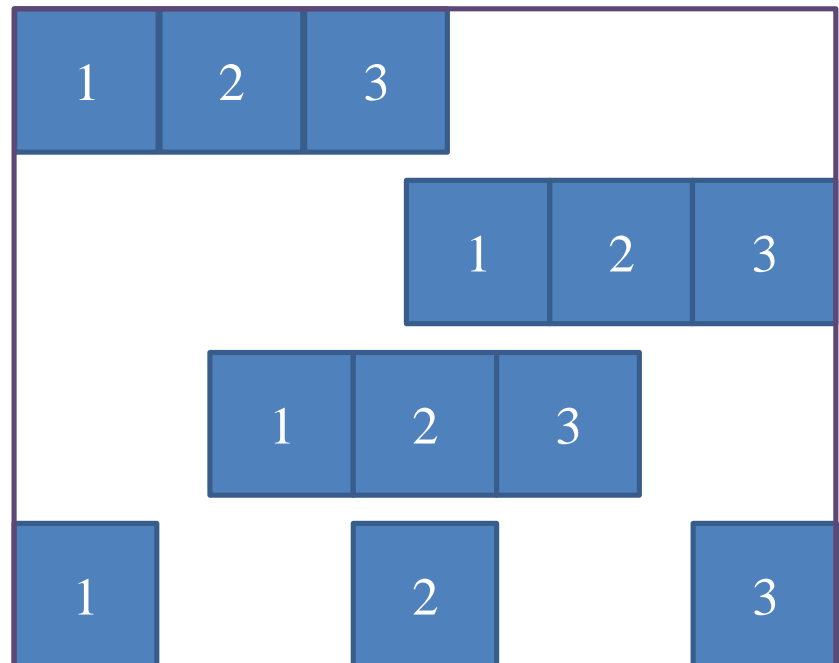
Vlastnosti Flex kontajnera (2)

- Pri Flexbox zobrazení sa rozlišuje hlavná a vedľajšia os, pri prednastavenej hodnote `flex-direction: row` je hlavná os horizontálna a vedľajšia vertikálna, pri `flex-direction: column` naopak
- Nasledujúci popis zodpovedá horizontálnej hlavnej osi
- `justify-content`
 - Zarovná potomkov na hlavnej osi
 - `flex-start` naľavo, `flex-end` napravo, `center` centrovane, `space-between` rovnomerne vložení medzier
- `align-items`
 - Zarovná potomkov na vedľajšej osi
 - `flex-start` hore, `flex-end` dole, `center` centrovane, `stretch` rozťahne elementy na celú výšku kontajnera

Príklad – justify-content

```
<style>
  .kontajner {
    display: flex;
    flex-direction: row;
  }
  p {
    height: 100px;
    width: 100px;
  }
</style>
...
<div class="kontajner">
  <p>1</p>
  <p>2</p>
  <p>3</p>
</div>
```

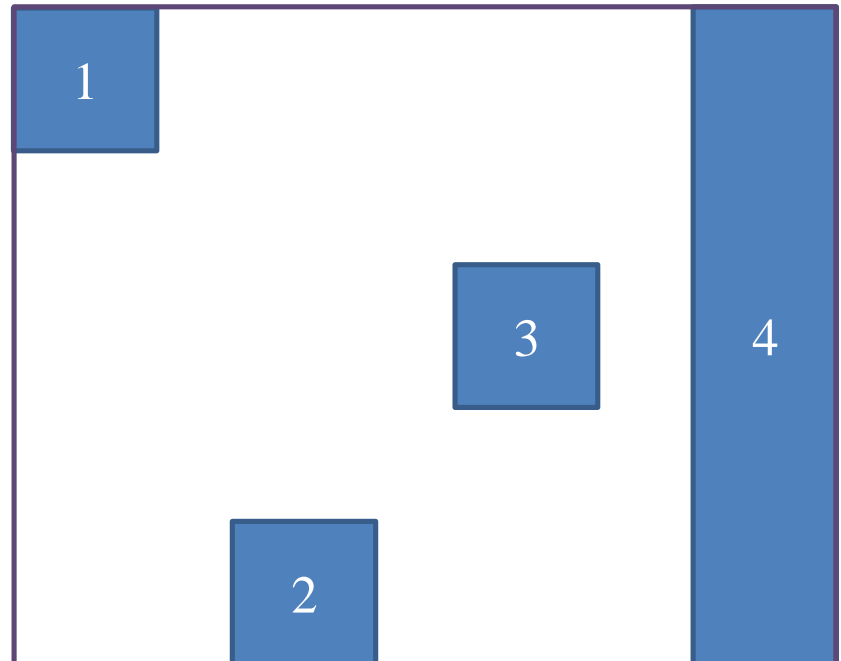
- flex-start
- flex-end
- center
- space-between



Príklad – align-items

```
<style>
  .kontajner {
    display: flex;
    flex-direction: row;
    justify-content:
      space-between;
  }
  p {
    height: 100px;
    width: 100px;
  }
</style>
...
<div class="kontajner">
  <p>1</p> ...
</div>
```

- flex-start
- flex-end
- center
- stretch



Pozícia elementov

- Elementy je možné umiestniť aj priamo nastavením vlastnosti `position` a definovaním vlastností `left`, `right`, `top`, alebo `bottom` ktoré určujú umiestnenie strán elementu:
- Hodnoty `position`:
 - `static` – štandardne podľa pravidiel zobrazenia
 - `relative` – relatívne posunutie voči štandardnému umiestneniu
 - `absolute` – relatívne umiestnenie voči prvému nadradenému nestatickému elementu
 - `fixed` – umiestnenie voči oknu prehliadača (tzn. element zostane na danej pozícii aj keď používateľ skroluje obsah okna)
- Hodnoty `left`, `right`, `top`, `bottom` sú rozmerové jednotky, alebo `auto` (štandardné umiestnenie)

Prekrývanie zobrazenia

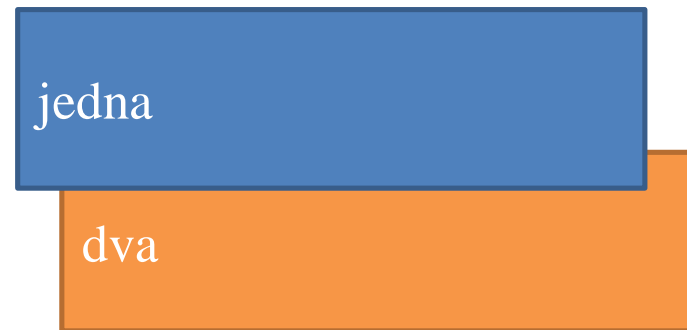
- Po zmene pozície, alebo pri pretečení obsahu sa môžu elementy prekrývať
- Vlastnosť **z-index** určuje poradie zobrazenia:
 - Číselná hodnota (môže byť nastavené aj záporné číslo)
 - Čím väčší index, tým viac bude element zobrazený v popredí a prekryje ostatné elementy
 - Prednastavená hodnota je **auto** – rovnaký index ako nadradený element od 0, elementy s rovnakým indexom sa zobrazia v poradí v akom sú definované v HTML (tzn. neskôr uvedené elementy prekryjú predchádzajúce)

Príklad - position, z-index

```
<style>
  #d2 {
    position: relative;
    left: 10px;
    top: -10px;
    z-index: -1;
  }
</style>

...
<div>jedna</div>
<div id="d2">dva</div>
```

Druhý element <div> posunieme hore a napravo o 10px, normálne by bol zobrazený nad prvým <div>, ale zmenšili sme **z-index**

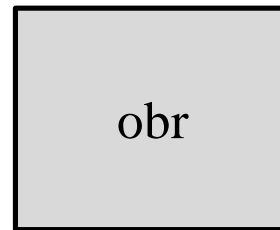


Plávajúce (*floating*) elementy

- **float**
 - Umiestni element čo najviac naľavo (**left**), alebo napravo (**right**) v rámci nadradeného elementu, obsah ostatných susedných elementov bude "obtekať" okolo plávajúceho elementu
 - **none** (predefinovaná hodnota) – štandardné umiestnenie bez obtekania
- **clear**
 - Zruší obtekania susedných plávajúcich elementov naľavo (**left**), alebo napravo (**right**), tzn. element sa zobrazí štandardne na samostatnom riadku
 - **none** (predvolená hodnota) – povolenie obtekania plávajúcich elementov z oboch strán

Príklad - plávajúce elementy

```
<a style="float: left;"  
  href="obr.png" />  
<p>  
  prvý prvý prvý prvý prvý  
  prvý prvý  
</p>  
<p style="clear: left;">  
  druhý druhý druhý druhý  
  druhý druhý druhý  
</p>
```



prvý prvý prvý prvý
prvý prvý prvý

druhý druhý druhý druhý druhý druhý
druhý