

8 BLOKOVÉ PRVKY NA WEBOVEJ STRÁNKE A ICH ŠTÝLOVANIE

V tejto kapitole sa oboznámime s niekoľkými blokovými elementami na (logické) členenie textu a ukážeme si tri dôležité vlastnosti blokových elementov, a to orámovanie, vonkajšie a vnútorné okraje.

Blokové elementy

V kapitole 2 sme sa naučili, ako môžeme štruktúrovať text pomocou blokových elementov definujúcich nadpisy (`<h1>`, .., `<h6>`) a odsek (`<p>`). V kapitole 6 sme spoznali elementy `<section>` a `<nav>`. V tejto časti si predstavíme ďalšie blokové elementy, ktoré nám umožňujú rozčleniť dokument na väčšie či menšie ucelené časti, a to elementy `<header>`, `<footer>`, `<article>`, `<aside>` a `<div>`. Zoznam všetkých blokových elementov nájdete na https://www.w3schools.com/html/html_blocks.asp.

Element `<header>` používame na definovanie hlavičky dokumentu, teda akéhosi úvodného obsahu, resp. obsahu, ktorý sa opakuje v úvode každej stránky webového sídla. Zvyčajne obsahuje jeden alebo viac nadpisov, logo, prípadne nejakú navigáciu. Nemýľme si element `<header>` s elementom `<head>`. Element `<head>` definuje hlavičku po „technickej stránke“, v rámci neho definujeme napr. titulok stránky, či kódovanie (pozri *Kapitolu 1*), uvádza sa **pred** elementom `<body>`. Elementom `<header>` definujeme hlavičku obsahovej časti webového dokumentu, teda **je vnorený** v elemente `<body>`.

Elementom `<footer>` definujeme päť webového dokumentu. V päte zväčša uvádzame informácie o autoroch a vlastníckych právach, kontaktné informácie prípadne aj s mapami, mapu sídla, odkaz na úvod stránky. Podobne ako hlavička, aj päta zvykne byť spoločná pre všetky stránky webového sídla.

PRÍKLAD 8.1

Na stránke IT Pizza (súbor 08/index.html.) definujeme elementy `<header>` a `<footer>`. Do hlavičky umiestnime hlavný nadpis (`h1`) a logo stránky (obrázok `logo.gif`). Do päty zahrnieme všetko od kontaktov na obe prevádzky až po koniec (odkazy na Facebook, Twitter a @).

```
<body>
  <header>
    <h1>IT Pizza</h1>
    
  </header>
  ...
  <footer id="kontakt">
    <h2>Kontakt</h2>
    <h3>Bratislava</h3>
    <address>Mlynská dolina, 842 48 Bratislava<br>
      tel. +421 999 123 456<br> email: ba@itpizza.sk </address>
    <h3>Banská Bystrica</h3>
    <address>Tajovského 40, 974 01 Banská Bystrica<br>
      tel. +421 999 123 457<br> email: bb@itpizza.sk </address>
    <p>Sledujte nás na
      <a href="https://www.facebook.com/" title="Facebook" ...
```



```

    <a href="https://twitter.com/" title="Twitter" ...
  </p>
  <p>&copy; IT akadémia, 2018, Mlynská dolina, 842 48
    Bratislava</p>
</footer>
</body>

```

Element `<nav>` definuje množinu odkazov, ktoré tvoria tzv. (hlavnú) navigáciu stránky. Uvedomte si, že nie všetky odkazy na stránke, musia byť súčasťou navigácie.

Hlavný obsah stránky (zvyčajne medzi hlavičkou a päťou) môžeme ďalej členiť pomocou blokových elementov: `<section>`, `<article>`, `<aside>`. Medzi elementami `<section>` a `<article>` v podstate nie je žiaden rozdiel, obidvoma môžeme označiť nejakú časť dokumentu. My budeme používať `<section>` na definovanie väčších, obsahovo ucelených, častí (napr. kapitoly) a `<article>` na definovanie ich podčastí, menších celkov (napr. podkapitoly). Element `<aside>` sa používa na definovanie vedľajšieho obsahu (poznámka, vysvetlivka, odbočenie, teda niečo, čo nemusí byť priamou súčasťou hlavného obsahu, ale s týmto obsahom nejako súvisí).



ÚLOHA 8.2

Na stránke IT Pizza v časti *Ponuka* použite element `<article>` na definovanie informácií pre každú pizzu, t.j. nahradte všetky elementy `` v tejto časti elementom `<article>`. Nezabudnite zrušiť element ``, a tiež príslušný štýl pre ``.

Tiež nahradte zvýraznenie pizze pomocou elementu `` využitím nadpisu `<h3>` a zrušte element `
` za názvom pizze. Všetky nahradenia možno vidieť na *obrázku 8.1*. Odporúčame využívať automatické nahradenie textu.

```

<li><strong>Margherita</strong><br>
  paradajková omáčka, syr <sub>1, 7</sub><br>
  
  veľká 4,50 &euro;
</li>

```

↓

```

<article><h3>Margherita</h3>
  paradajková omáčka, syr <sub>1, 7</sub><br>
  
  veľká 4,50 &euro;
</article>

```

Obrázok 8.1 Nahradenie elementu `` elementom `<article>` na stránke IT Pizza.

Časť stránky o aktuálne prebiehajúcich akciách definujte pomocou elementu `<aside>` (príslušné `<section>` nahradte `<aside>`).

ODPOVEDZTE



Mali zmeny, ktoré sme spravili v príklade 8.1 a úlohe 8.2, vplyv na vzhľad stránky IT Pizza?

Okrem už uvedených blokových elementov sa veľmi často pri tvorbe stránok používa element `<div>`. Umožňuje nám deliť stránku na logické celky bez toho, aby sme špecifikovali obsahový význam týchto celkov (ako to robia napr. elementy `header` alebo `footer`). Podľa w3schools `div` sa veľmi často používa ako kontajner pre iné HTML elementy v spojení s CSS na definovanie vzhľadu časti webového dokumentu alebo v spojení s JavaScriptom na definovanie nejakého správania. Elementy `<div>` môžeme do seba vnárať do ľubovoľnej hĺbky.

PRÍKLAD 8.3



Na stránke IT Pizza dáme do hlavičky stránky (`<header>`) na pozadie obrázok zo súboru `header-bg.jpg`. Do štýlov pridáme štýl pre `header`.

```
<style>
...
header {
  background-image: url (obrazky/header-bg.jpg) ;
}
</style>
```

ÚLOHA 8.4



Na stránke IT Pizza definujte vhodné štýly tak, aby:

- navigácia mala farbu pozadia `#FFE4C4`,
- päta stránky mala farbu pozadia `#FF9900` a farbu textu čiernu,
- element `<aside>` mal text zarovnaný na stred a pozadie farby `#F5F6F7`,
- nadpis `<h1>` nemal žiadne pozadie.

Rôzne štýly pre rovnaký typ elementu

Štýly, ktoré sme definovali doteraz, sa vždy aplikovali na všetky výskyty elementu, pre ktorý boli zadefinované. Napr. ak sme definovali štýl `h2 {font-family: Verdana}`, nastavili sme tým typ písma pre všetky nadpisy úrovne 2 na stránke. Niekedy by sme však potrebovali mať rôzne štýly pre ten istý druh elementu, napr. jeden štýl pre odkazy v navigácii a iný štýl pre ostatné odkazy.

PRÍKLAD 8.5



V kóde na obrázku 8.2 máme tri výskyty elementu `h1`, z toho jeden je v hlavičke. Chceli by sme vizuálne odlíšiť nadpis `h1`, ktorý je v hlavičke od ostatných nadpisov `h1` (teda tých, ktoré nie sú v hlavičke).

```

<body>
  <header><h1>Vitamíny</h1></header>

  <section id="tuky">
    <h1>Vitamíny rozpustné v tukoch</h1>
    <h2>Vitamín A</h2>
    <p>Hlavným zdrojom je plnotučné mlieko, <a href="https://sk.wikipedia.org/wiki/Vajce">
vajcia</a> a pečeň.</p>
    <h2>Vitamín D</h2>
    <p>Za normálnych okolností sa vitamín D tvorí v koži pôsobením slnečného žiarenia.</p>
  </section>

  <section id="voda">
    <h1>Vitamíny rozpustné vo vode</h1>
    <h2>Vitamín B1 - Tiamín</h2>
    <p>Hlavným zdrojom je droždie, <a href="https://sk.wikipedia.org/wiki/Obilie">obilné
klíčky</a> a syr.</p>
    <h2>Vitamín C</h2>
    <p>Jeho hlavným zdrojom sú ovocie a zelenina.</p>
  </section>

  <footer>Zdroj: <a href="https://sk.wikipedia.org/wiki/Vitam%C3%ADn">Wikipédia</a></footer>
</body>

```

Obrázok 8.2 Kód s viacerými výskytmi elementu h1.

V editore JSFiddle pomocou štýlov zmeníme zobrazenie stránky s kódom z *obrázku 8.2* (kód nájdete v súbore 08/vitaminy.html, skopírujte ho do časti HTML). V časti CSS definujeme potrebné štýly tak, aby:

- všetky nadpisy úrovne 1 boli písmom `Verdana`,
- nadpisy úrovne 1, ktoré sú v hlavičke stránky (element `<header>`) mali pozadie farby `lightgreen` (*obrázok 8.3*).

```

h1 {font-family: Verdana;}
header h1 {
  background-color: lightgreen;
}

```

Vitamíny

Vitamíny rozpustné v tukoch

Obrázok 8.3 Rôzne oštylované nadpisy h1.

Zápisom `header h1` definujeme štýl len pre tie elementy `h1`, ktoré sú vnorené v elemente `<header>` (elementy `h1`, ktoré nie sú v `header`, sa nezmenia). Vlastnosti, ktoré definujeme v štýle `h1`, sa aplikujú na všetky výskyty elementu `h1`, vrátane tých, ktoré sú v hlavičke.



ÚLOHA 8.6

V kóde na *obrázku 8.2* máme tri odkazy (elementy `a`): jeden z nich je v päte (`footer`) a dva v sekcíach. Vizualne rozlíšte odkazy, ktoré sú v päte od odkazov, ktoré sú mimo nej. Odkazy v päte nech sú farby `#990033`, ostatné odkazy nech sú farby `darkblue`.

Ak chceme definovať **štýl len pre jeden konkrétny výskyt elementu**, môžeme využiť identifikátor elementu.

PRÍKLAD 8.7



V html dokumente z predchádzajúceho príkladu máme dve sekcie, každej sme dali identifikátor. Sekcii s identifikátorom `voda` dáme svetlomodré pozadie (`#CCFFFF`), sekciu s identifikátorom `tuky` svetložlté pozadie (`#FFF5CC`).

```
#voda {
  background-color: #CCFFFF;
}
#tuky {
  background-color: #FFF5CC;
}
```

ÚLOHA 8.8



Na stránke IT Pizza upravte použitím štýlov:

- odkazy v navigácii: farba `#333333`, tučné, nepodčiarknuté a pri nadídení myšou sa farba pozadia zmení na `#FF9900`,
- ostatné odkazy: farba `#CC6600`, nepodčiarknuté, pri nadídení myšou sa podčiarknu.

ponuka **fotogaléria** kontakt



Našu pizzu si môžete vychutnať aj v stravovacích zariadeniach **Free Food**.
Objednať si môžete cez náš objednávkový formulár alebo prostredníctvom **Bistro.sk**.

Obrázok 8.4 Odkazy v navigácii (hore) a ostatné odkazy (dolu) na stránke IT Pizza.

Zopakujme si, akými spôsobmi už vieme definovať štýly:

- `p {...}` - tento štýl definuje vlastnosti pre všetky výskyty elementu `p` na stránke,
- `header h1 {...}` - tento štýl definuje vlastnosti pre tie výskyty elementu `h1`, ktoré sú vnorené v elemente `header`,
- `#id {...}` - tento štýl definuje vlastnosti elementu s daným `id`.

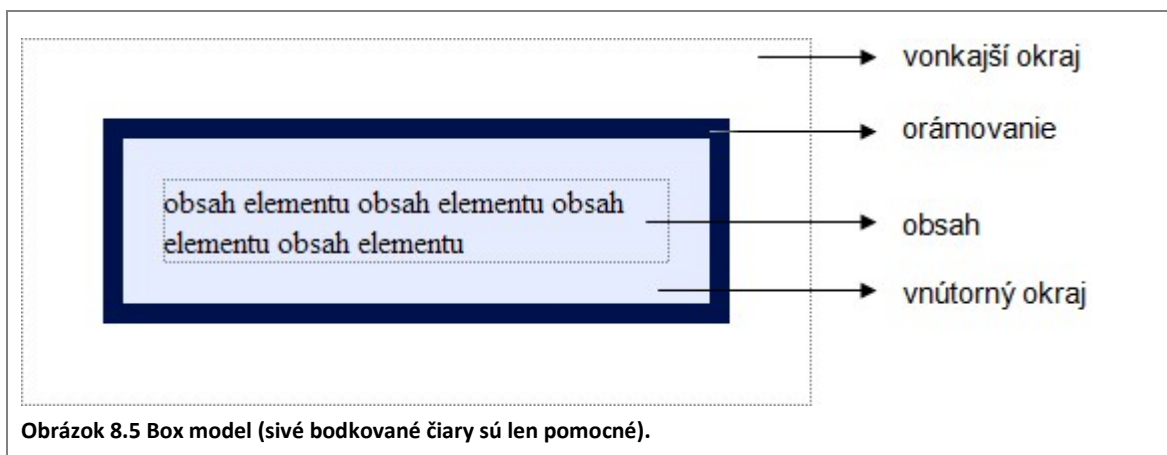
CSS – okraje elementu

Pre každý element sa automaticky generuje tzv. **box**. Box sa skladá z **obsahu**, **vnútorného okraja**, **orámovania** a **vonkajšieho okraja** (Obrázok 8.5).

Orámovanie elementu je tmavý rámik na *obrázku 8.5*. Orámovanie nemusí byť vždy viditeľné.

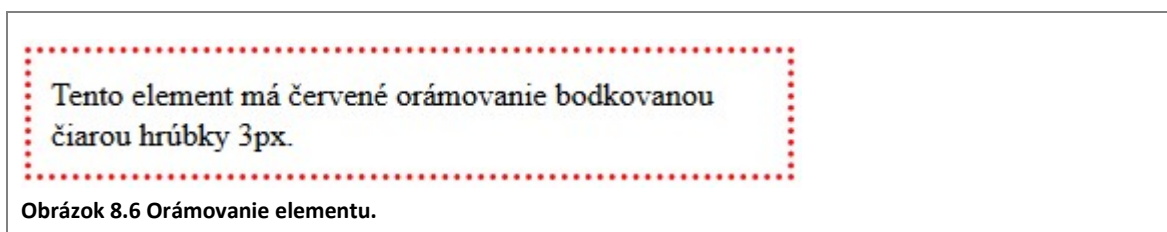
Priestor medzi obsahom elementu a jeho orámovaním nazývame **vnútorný okraj**.

Priestor medzi orámovaním elementu a okolitými elementmi nazývame **vonkajší okraj**. Na obrázku je to vzdialenosť medzi orámovaním elementu (modrý rámik) a bodkovaným rámikom (ľubovoľný element okolo).



Orámovanie

Pomocou vlastností `border-color`, `border-width` a `border-style` môžeme nastaviť farbu, hrúbku a štýl orámovania elementu.



PRÍKLAD 8.9

V editore JSFiddle definujeme v časti HTML jeden odsek (element `<p>`) s ľubovoľným textom (napr. zo súboru `08/vitamin.txt`). Pomocou kaskádových štýlov orámujeme tento odsek modrou súvislou čiarou hrúbky `3px` (obrázok 8.7).

```
p {
  border-width: 3px;
  border-style: solid;
  border-color: #0000FF;
}
```

Obrázok 8.7 Orámovanie elementu.



ZAPAMÄTAJTE SI

Orámovanie elementu sa zobrazí len vtedy, ak sú nastavené farba, hrúbka aj štýl.

Okrem farby, hrúbky a štýlu vieme orámovaniu nastaviť aj mieru zaoblenia rohov. Význam všetkých vlastností orámovania a ich možné hodnoty uvádzame v *tabuľke 8.1*.

Tabuľka 8.1 Vlastnosti orámovania.

vlastnosť	hodnoty	popis, význam
<code>border-color</code>	<i>názov farby alebo RGB farby</i>	definuje farbu orámovania (pre všetky štyri okraje)
<code>border-style</code>	<code>none</code> <code>hidden</code> <code>dotted</code> <code>dashed</code> <code>solid</code> <code>double</code> <code>groove</code> <code>ridge</code> <code>inset</code> <code>outset</code>	definuje štýl orámovania (pre všetky štyri okraje)
<code>border-width</code>	<i>hrúbka orámovania v %, px, em</i> <code>thin</code> <code>medium</code> <code>thick</code>	definuje hrúbku orámovania (pre všetky štyri okraje)
<code>border-radius</code>	polomer zaoblenia v <i>px</i> alebo % (<i>alebo ľubovoľná jednotka</i>)	definuje polomer zaoblenia rohov

ÚLOHA 8.10

V kóde z príkladu 8.9 experimentujte s vlastnosťou `border`:

- nastavte hodnotu `border-width` postupne na `1px`, `5px`, `10px`, `40px`, `1%`, `1em`, `1.5em`,
- vyskúšajte rôzne hodnoty pre `border-style`, napr. `dotted`, `dashed`, `double`, `groove`, `ridge`, `inset`, `outset`,
- zmeňte `border-color`.

Štýl pre element `p` z príkladu 8.9 môžeme definovať aj pomocou združenej vlastnosti `border`.

```
p {
  border: 3px solid #0000FF;
}
```

Pri použití združenej vlastnosti musíme hodnoty zadať v poradí: *hrúbka štýl farba* (oddelené medzerou).

V prípade, že použijeme vlastnosť `border`, nastavujeme vlastnosti orámovania na štyroch stranách: *hore*, *vpravo*, *dolu* a *vľavo*. Existujú však aj vlastnosti `border-top`, `border-bottom`, `border-left` a `border-right`. Pomocou týchto vlastností môžeme samostatne definovať farbu, hrúbku a štýl orámovania pre hornú, dolnú, ľavú a pravú stranu elementu.

PRÍKLAD 8.11

Do HTML kódu z príkladu 8.9 doplníme jeden nadpis úrovne 1 a nadpis aj odsek ešte vnoríme do elementu `section` (aktualizovaný HTML kód skopírujeme zo súboru 08/vitamin2.txt). V CSS nastavíme:

- pre nadpis dolné a horné orámovanie hrúbky `3px`, dvojité, farby `#990000`,
- pre `section` tenké sivé súvislé orámovanie na všetkých štyroch stranách s mierne oblými rohmi (pozri obrázok 8.8).

```
h1 {
  border-bottom: 3px double #990000;
  border-top: 3px double #990000;
}
section {
  border: thin solid gray;
```



```
border-radius: 5px;
}
```

Vitamíny

Vitamín je látka, ktorú prijme organizmus vo veľmi malých množstvách, ale zároveň nevyhnutne, potrebuje na svoju existenciu, no nedokáže si ju sám syntetizovať a musí ju teda získavať v potrave.

Obrázok 8.8 Rôzne typy orámovania elementov.



ÚLOHA 8.12

Na stránke IT Pizza (pozri obrázok 8.9) definujte pomocou štýlov:

- orámovanie okolo časti akcia (elementu `aside`) – orámovanie by malo byť zľava, sprava a zdola, rovnakej farby, ako majú nadpisy `h2` – hrúbku a štýl orámovania si zvolíte,
- orámovanie okolo informácií o každej ponúkanej pizze (t.j. okolo elementu `article`, ktorý je v `section`) – zvolíte tenké súvislé orámovanie sivej farby,
- zmenu farby pozadia bloku informácií o jednej pizze (`article` v `section`) pri naďínení myšou – farba sa zmení na svetlosivú,
- orámovanie okolo obrázkov vo fotogalérii (nie iných!) – vzhľad orámovania si zvolíte.

Quattro Formaggi

paradajková omáčka, 4 druhy syra 1, 7



malá 4,50 €
veľká 6,00 €

Tonno

paradajková omáčka, mozzarella, tuniak, cibuľa 1, 4, 7



malá 4,00 €
veľká 5,50 €

Zoznam alergénov

Akcia

Každý pondelok pri kúpe troch pizz podľa ľubovoľného výberu máte najlacnejšiu z nich zadarmo.

Obrázok 8.9 Orámovanie na stránke IT Pizza (z úlohy 8.12).

Vonkajšie okraje

Vlastnosť `margin` určuje vzdialenosť orámovania elementu od okolitých elementov na stránke vo všetkých smeroch, tzv. vonkajší okraj elementu. V prípade, že element má definovanú farbu pozadia, tak voľný priestor definovaný vlastnosťou `margin` nebude podfarbený touto farbou.

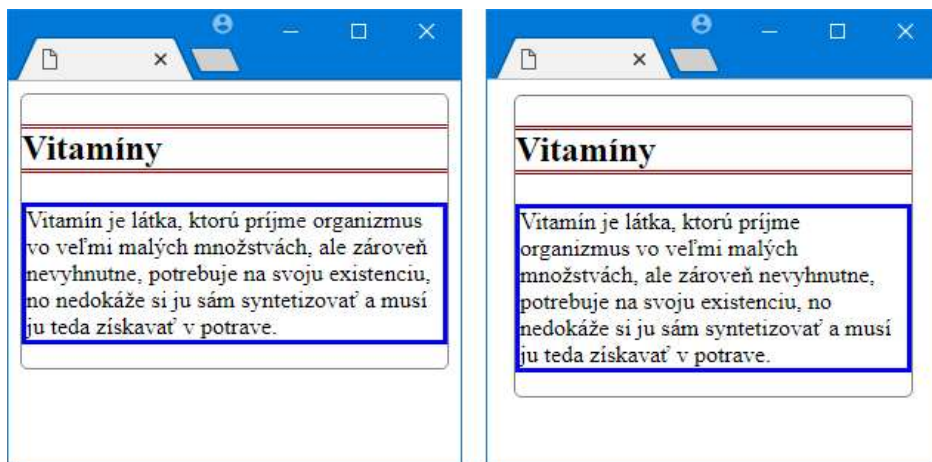
Tento element má vonkajší okraj 30px.

Obrázok 8.10 Vonkajší okraj elementu.

PRÍKLAD 8.13

Elementu `section` z príkladu 8.11 nastavíme vonkajší okraj na `10px`. Rozdiel medzi `section` s nenastaveným vonkajším okrajom a s nastaveným vonkajším okrajom je na obrázku 8.11.

```
section {  
  border: thin solid gray;  
  border-radius: 5px;  
  margin: 10px;  
}
```



Obrázok 8.11 Vľavo bez vonkajšieho okraja, vpravo s vonkajším okrajom 10px pre `section`.

ÚLOHA 8.14

V kóde z príkladu 8.13:

- meňte hodnoty `margin` postupne na `20px`, `30px`, `40px`, `60px`,
- meňte hodnoty `margin` postupne na `10%`, `5%`, `2%`,
- meňte hodnoty `margin` postupne na `1em`, `2em`, `2.5em`,
- vyskúšajte, čo sa stane, ak použijete záporné číslo (napr. `-10px`),
- pridajte `margin` pre element `h1`, skúšajte rôzne hodnoty,
- pridajte `margin` pre element `p`, skúšajte rôzne hodnoty.

Podobne ako pri orámovaní, aj pri vlastnosti `margin` máme možnosť samostatne nastaviť horný, pravý, dolný, či ľavý vonkajší okraj elementu. Môžeme to spraviť pomocou samostatných vlastností: `margin-top`, `margin-right`, `margin-bottom`, `margin-left`.

Na nastavenie rôznych hodnôt vonkajších okrajov na jednotlivých stranách môžeme tiež použiť združenú vlastnosť `margin` tak, že namiesto jedného čísla zadáme štvoricu čísel, ktoré určujú horný, pravý, dolný a ľavý okraj (v tomto poradí). Hodnoty oddeľujeme medzerou.



PRÍKLAD 8.15

V JSFiddle doplníme kód z príkladu 8.13. Odseku nastavíme ľavý a pravý vonkajší okraj na `20px`, spodný vonkajší okraj na `10px`. Nadpisu nastavíme ľavý vonkajší okraj na `20px`.

```
p {
  border: 3px solid #0000FF;
  margin: 0px 20px 10px 20px;
}
h1 {
  border-bottom: 3px double #990000;
  border-top: 3px double #990000;
  margin-left: 20px;
}
```



Obrázok 8.12 Nastavenie okrajov.



ÚLOHA 8.16

Experimentujte s vlastnosťou `margin`.

- Nastavte elementu `h1` horný vonkajší okraj na 0.
- Ktoré vonkajšie okraje a na akú hodnotu treba v našom príklade nastaviť elementom `h1` a `p`, aby boli tieto dva elementy nalepené na seba?
- Pozrite si vaše pokusy aj bez nastavenia orámovania (stačí, ak vlastnosť `border` v definícii štýlu zakomentujete pomocou `/* ...*/`, napr. `/*border: 3px solid #0000FF; */`).
- Experimentovaním zistite, ktoré vonkajšie okraje majú prednastavené elementy `h1`, `p` a `section` (t.j. s hodnotou inou ako 0).

- Porozmýšľajte a experimentovaním overte, ktoré iné elementy majú prednastavené nejaké nenulové vonkajšie okraje.
- Experimentujte s vlastnosťou `margin` pre element `body`.

ÚLOHA 8.17

Na stránke IT Pizza využite vlastnosť `margin` na to, aby ste:

- prilepili celý obsah stránky k okrajom okna prehliadača,
- aby sa jednotlivé sekcie k sebe prilepili.

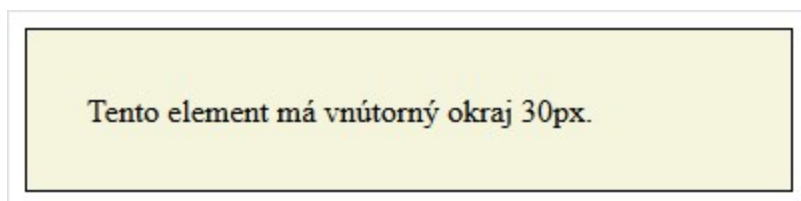
Pomôcka: Pouvažujte, ktorým elementom treba prestaviť vonkajšie okraje, či všetky alebo len na niektorej strane a na akú hodnotu. Zakaždým zvážte, či tak treba urobiť pre všetky výskyty daného elementu, alebo iba pre niektoré výskyty, ktoré sú napr. vnorené v inom elemente.



Obrázok 8.13 Využitie vlastnosti `margin` na stránke IT Pizza.

Vnútorne okraje

Vlastnosť `padding` určuje vzdialenosť obsahu elementu od jeho orámovania, tzv. vnútorný okraj elementu. V prípade, že element má definovanú farbu/obrázok pozadia, tak voľný priestor definovaný vlastnosťou `padding` bude tiež podfarbený.



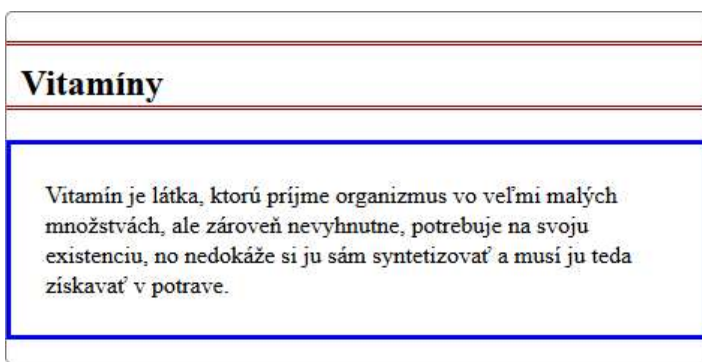
Obrázok 8.14 Vnútorný okraj.



PRÍKLAD 8.18

V editore JSFiddle pracujeme s HTML kódom z príkladov 8.13, 8.15. Z CSS kódu zrušíme všetky nastavenia vlastnosti `margin`. Definujeme elementu `<p>` vnútorný okraj na všetkých stranách na hodnotu `5%` a elementu `<h1>` ľavý a pravý vnútorný okraj na hodnotu `10px`.

```
p {
  border: 3px solid #0000FF;
  padding: 5%;
}
h1 {
  border-bottom: 3px double #990000;
  border-top: 3px double #990000;
  padding-left: 10px;
  padding-top: 10px;
}
```



Obrázok 8.15 Nastavenie vnútorných okrajov.

Pomocou vlastností `padding-top`, `padding-right`, `padding-bottom`, `padding-left` môžeme nastaviť vzdialenosť od horného, pravého, dolného či ľavého okraja elementu.

Vzdialenosti od všetkých štyroch strán môžeme tiež nastaviť pomocou združenej vlastnosti `padding`, ktorej hodnotou bude štvorica čísel oddelených medzerou v poradí: top right bottom left. Napr. `padding: 5px 10px 20px 10px` definuje vzdialenosť od horného okraja `5px`, vzdialenosť od ľavého a pravého okraja `10px` a vzdialenosť od dolného okraja `20px`. Ak zadáme iba jedno číslo, okraje budú rovnaké pre všetky štyri strany.



ÚLOHA 8.19

V CSS kóde z príkladu 8.18 experimentujte s vlastnosťou `padding`.

- Meňte hodnotu vnútorného okraja pre odsek. Skúste zmeniť jednotku `%` na `px`, či `em` a znova skúšajte rôzne hodnoty. Dá sa použiť aj záporná hodnota?
- Pre nadpis nastavte namiesto ľavého a horného vnútorného okraja pravý a dolný vnútorný okraj. Vyskúšajte rôzne hodnoty pre tieto dva okraje.
- Vyriešte predchádzajúcu úlohu pomocou združenej vlastnosti `padding` so štvoricou čísel.
- Pre element `<section>` nastavte nejaké jednoduché orámovanie a vonkajší okraj na `2%`. Pozrite si výsledok. Potom vonkajší okraj zmeňte na vnútorný a porovnajte.

ÚLOHA 8.20



Na stránke IT Pizza využite vlastnosť `padding` na odsadenie textu od okraja vo všetkých hlavných častiach stránky (hlavička, päta, jednotlivé sekcie), a taktiež v blokoch pre každú pizzu. Veľkosť vnútorného okraja si zvolte. Pre navigáciu a odkazy v navigácii nastavte vnútorný okraj na všetkých stranách na `1em`. Zrušte oranžové pozadie nadpisov úrovne 2.



Obrázok 8.16 Využitie vlastnosti padding na stránke IT Pizza.

Šírka a výška elementu

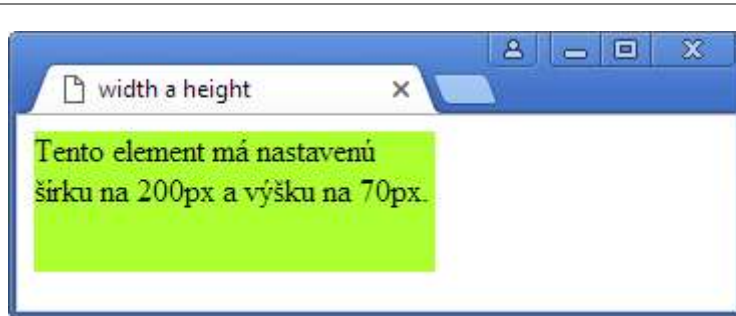
Kaskádové štýly nám umožňujú nastaviť rozmery elementu. Slúžia na to vlastnosti `width` a `height`.

PRÍKLAD 8.21



V súbore `08/rozmary.html` máme definovaný jeden element `div`. Definujme mu farbu pozadia, šírku `200px` a výšku `70px`.

```
...
<style>
  div {
    background-color: greenyellow;
    width: 200px;
    height: 70px;
  }
</style>
</head>
```



Obrázok 8.17 Element s nastavenou šírkou a výškou v px.

Hodnoty vlastností `width` a `height` môžeme zadávať v dĺžkových jednotkách (napr. `px`, `em`) alebo v percentách (vzhľadom na nadradený element).



ÚLOHA 8.22

V kóde z príkladu 8.21 experimentujte s hodnotami vlastností `width` a `height`:

- Zmeňte šírku na `300px`, `400px`, `100px` a pozorujte. Čo sa stane, ak zvolíte príliš malú hodnotu?
- Nastavte šírku na `200px` a meňte len výšku – na `100px`, `30px`, `20px`.
- Úplne zrušte nastavenie výšky, ponechajte len nastavenie šírky. Akú výšku (tým nemyslíme číselné vyjadrenie) bude mať element `div` teraz?

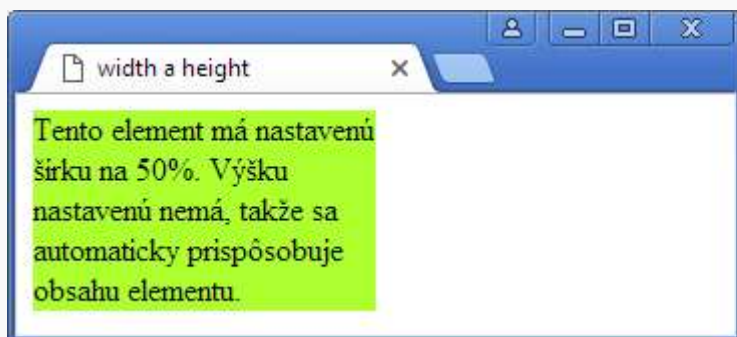
Štandardne majú vlastnosti `width` a `height` nastavenú hodnotu `auto`, čo znamená, že šírku a výšku elementu si vypočíta prehliadač, keď zobrazuje stránku.

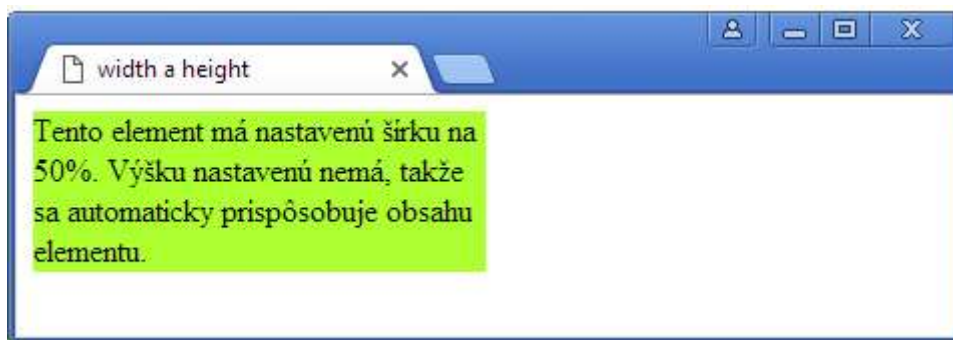


PRÍKLAD 8.23

Pokračujeme v príklade 8.21 (odporúčame pridať do elementu `<div>` viac textu). V CSS časti zmeníme elementu `div` šírku na `50%`.

```
div {
  background-color: greenyellow;
  width: 50%;
}
```





Obrázok 8.18 Element s nastavenou šírkou na 50% pri rôznych veľkostiach okna prehliadača.

ÚLOHA 8.24

V kóde z príkladu 8.23 experimentujte s nastavením šírky, resp. výšky elementu v %.

- Meňte šírku okna prehliadača a pozorujte, ako sa mení šírka elementu.
- Postupne meňte šírku elementu na 100%, 80%, 20%, 5% a pozorujte. Čo sa stane, keď zvolíte príliš malú hodnotu?
- Zopakujte predchádzajúce dva body pre výšku. Je správanie stránky rovnaké ako pri nastavení šírky?

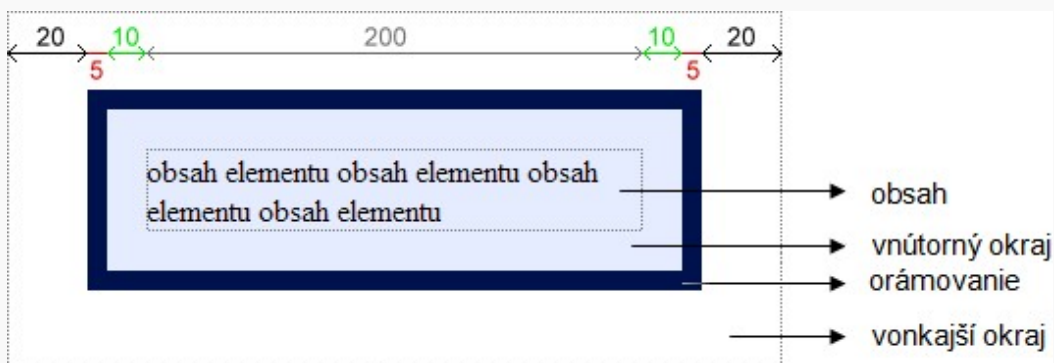
Vlastnosťami `width` a `height` nastavujeme šírku a výšku obsahu elementu, t. j. ich hodnoty nezahŕňajú veľkosť vonkajšieho okraja, orámovania a vnútorného okraja.

PRÍKLAD 8.25

Vypočítajme šírku, ktorú v prehliadači zaberá element `div` s nasledujúcimi nastaveniami:

```
div {
  width: 200px;
  padding: 10px;
  border: 5px solid darkblue;
  margin: 20px;
}
```

Jednotlivé rozmery si znázorníme na obrázku (Obrázok 8.19). Celková šírka boxu, ktorý náš element `div` zaberá v prehliadači, bude: 20px (ľavý vonkajší okraj) + 5px (ľavé orámovanie) + 10px (ľavý vnútorný okraj) + 200px (šírka obsahu) + 10px (pravý vnútorný okraj) + 5px (pravé orámovanie) + 20px (pravý vonkajší okraj) = 270px.



Obrázok 8.19 Výpočet šírky boxu, ktorý zaberá element v prehliadači.

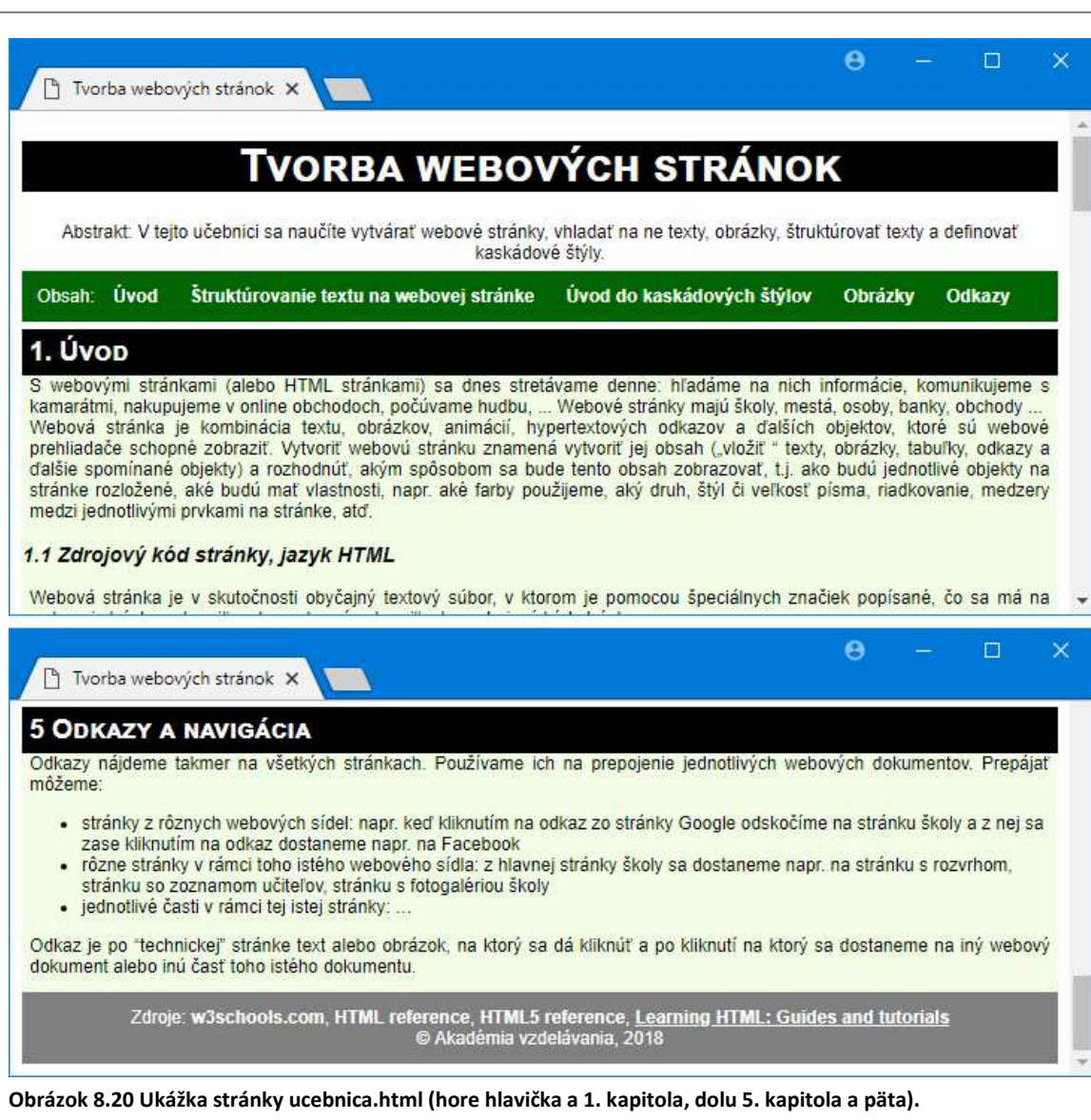
Celková šírka boxu, ktorý element zaberá v prehliadači, sa vypočíta ako ľavý margin + ľavý border + ľavý padding + šírka obsahu + pravý padding + pravý border + pravý margin. Analogický výpočet platí pre výšku.



ÚLOHA 8.26

Upravte štruktúru webovej stránky 08/ucebnica.html a oštyľujte ju:

- definujte na stránke časti `header` (nadpis a abstrakt), `nav` (odkazy na jednotlivé kapitoly), `footer` (odkazy na zdroje, autori),
- `header`, `nav` a `footer` nejakú pekne oštyľujte (inšpirácia na *obrázku 8.20*), samostatne oštyľujte odkazy v navigácii, odkazy v päte a ostatné odkazy,
- zabezpečte, aby sekcie pre 1., 3. a 5. kapitolu mali pozadie inej farby (napr. #F2FFE6) ako sekcie pre 2. a 4. kapitolu,
- skúste vo vašich štýloch využiť aj niektoré z vlastností `border`, `padding`, `margin`, `width`.



Obrázok 8.20 Ukážka stránky ucebnica.html (hore hlavička a 1. kapitola, dolu 5. kapitola a päta).