

# 11 RÔZNE ŠTÝLY PRE RÔZNE ZOBRAZOVACIE ZARIADENIA, ŠTÝL PRE TLAČ

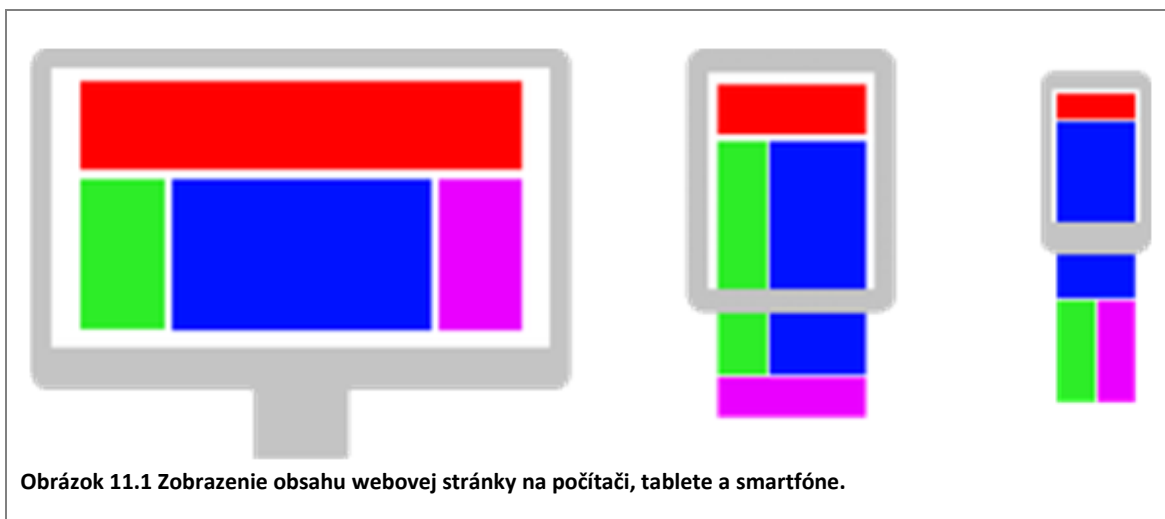
Webové stránky si pozeráme na rôznych zariadeniach: počítačoch, notebookoch, tabletoch či mobiloch. Každé z týchto zariadení má inú veľkosť zobrazovacej plochy - to, čo sa zmestí na šírku na obrazovku počítača, sa v rovnakom tvare určite nezmesť na obrazovku mobilu. Preto by tvorcovia stránok mali stránky vytvárať tak, aby sa správne zobrazovali na všetkých zariadeniach. Nazývame to responzívny web (resp. responzívny dizajn) - t.j. taký web (dizajn), ktorý sa automaticky prispôsobí zariadeniu, na ktorom je zobrazený.

Viete, že Google od roku 2015 umiestňuje responzívne webstránky vyššie vo vyhľadávaní?

## ÚLOHA 11.1

Nájdite príklady stránok, ktoré sa inak zobrazujú na počítači, inak v tablete či v mobile.

V čom sa líši počítačová verzia stránky od tej mobilnej? Aké je rozloženie jednotlivých prvkov? Sú tieto verzie obsahovo rovnaké?



Obrázok 11.1 Zobrazenie obsahu webovej stránky na počítači, tablete a smartfóne.

Ako vytvárať responzívne stránky? Máme dve možnosti:

- použiť niektorý z existujúcich CSS frameworkov, ktoré už majú predpripravené responzívne prvky a umožňujú škálovanie. Medzi najznámejšie patria Bootstrap, Skeleton, Foundation.
- vytvárať rozloženie stránky „od nuly“, t.j. písať vlastný kód, v ktorom využívame len HTML a CSS. Responzivnosť dosahujeme v podstate tým, že vytvárame viacero verzií štýlov: verziu pre mobilné zariadenia, verziu pre tablety, verziu pre počítače.

Pri tvorbe webu by sme už od začiatku mali myslieť na to, aby bol náš web responzívny. V súčasnosti sa za optimálny prístup považuje tzv. **mobile first** prístup, teda že najskôr vytvoríme najjednoduchšiu verziu webu vhodnú pre mobily a postupne pridávame vlastnosti pre rôzne ostatné zariadenia.

## 11.1 Media queries

Pri vytváraní responzívneho webu využívame tzv. **media-queries**. Media queries sú niečo ako podmienky a pravidlá, pomocou ktorých môžeme určiť, čo a ako zobrazovať na ktorom zariadení v závislosti od jeho vlastností, napr. šírky, výšky.



### PRÍKLAD 11.2

V editore JSFiddle skopírujeme do časti JSFiddle obsah súboru `11/sutaze.txt`. Pomocou CSS upravíme stránku tak, že v prípade jej zobrazenia na obrazovke so šírkou minimálne 800px, sa zväčší font na 120% a zmení farba textu na tmavomodrú (*obrázok 11.1*), inak bude farba textu a veľkosť fontu štandardná (*obrázok 11.2*).

```
@media screen and (min-width: 800px) {  
  body {  
    font-size: 120%;  
    color: darkblue;  
  }  
}
```

Otestujme zmenou veľkosti okna so stránkou.

# Informatické súťaže

## Palma

zameranie: programovanie

Súťaž je určená študentom SŠ. Organizuje ju Prírodovedecká fakulta UPJŠ v Košiciach a združenie STROM.

Obrázok 11.2 Zobrazenie pri šírke aspoň 800px (zväčšená veľkosť písma, modrá farba textu).

# Informatické súťaže

## Palma

zameranie: programovanie

Súťaž je určená študentom SŠ. Organizuje ju Prírodovedecká fakulta UPJŠ v Košiciach a združenie STROM.

Obrázok 11.3 Zobrazenie pri šírke menšej ako 800px (štandardná veľkosť písma, štandardná farba textu).

Všimnime si v CSS kóde riadok `@media screen and (min-width: 800px)`. To je **media query** – akási podmienka, ktorá ak je splnená, tak sa na stránku aplikujú štýly definované v zátvorkách `{ }` za ňou, v našom príklade štýl pre element `body`.

My budeme zväčša využívať media queries v tvare:

```
@media typ_média and vlastnosť_média
```

Význam jednotlivých častí je nasledovný:

- pomocou `@media` povieme, že ideme definovať media query,
- `typ_média` určuje typ média, pre ktorý chceme definovať štýly. Najčastejšie sa stretne s hodnotou `screen` (obrazovka počítača, tabletu, smartfónu,...), ale môže mať aj hodnotu `print` (tlačiareň), `speech` (čítač obrazovky) alebo `all` (všetky uvedené),
- `vlastnosť_média` definuje nejakú vlastnosť média, pre ktorú sa má aplikovať daný štýl, napr.:
  - `min-width`,
  - `max-width`,
  - `min-height`,
  - `max-height`,
  - `orientation` – môže mať hodnoty `portrait` (prednastavená) a `landscape`.

### PRÍKLAD 11.3

Nastavme stránke z predchádzajúceho príkladu sivé pozadie v prípade, že je zobrazená na výšku a pozadie farby `#FFCCCC`, ak je zobrazená na šírku. Taktiež nastavme font pre text na `Verdana`, a to pri ľubovoľnej orientácii stránky.

Štandardná orientácia stránky je na výšku, preto pre nastavenie sivého pozadia nemusíme definovať žiaden media-query, stačí pridať klasický štýl pre element `body` s vlastnosťou `background-color: gray;`. Ak pri orientácii stránky na šírku chceme nastaviť inú farbu pozadia, musíme zdefinovať media-query s podmienkou `screen and (orientation: landscape)` a v rámci neho definovať štýl pre `body` s nastavením `background-color: #FFCCCC;`. Nastavenie fonu `Verdana` stačí zdefinovať v štýle pre `body` mimo media-query. Tento štýl sa totiž použije na všetkých zariadeniach, aj tých s orientáciou na výšku aj tých s orientáciou na šírku.

```
body {
  background-color: gray;
  font-family: Verdana;
}
@media screen and (min-width: 600px) {
  body {
    font-size: 120%;
    color: darkblue;
  }
}
@media screen and (orientation: landscape) {
  body {
    background-color: #FFCCCC;
  }
}
```





## ÚLOHA 11.4

Doplňte CSS kód z predchádzajúceho príkladu tak, aby nadpisy úrovne 2 boli centrovane a pri šírke okna aspoň 800px zarovnané vpravo. Svoj kód otestujte – meňte veľkosť okna tak, aby bolo okno raz na šírku, raz na výšku.

Ak si stránky vytvorené v *príkladoch 11.2 až 11.4* prezeráme na počítači, vidíme, že na zmenu veľkosti okna reagujú tak, ako by sme očakávali. Ak by sme ich testovali na mobilnom telefóne, s najväčšou pravdepodobnosťou by sa zobrazili vo verzii pre šírku aspoň 800px. K verzii pod 800px by sme sa vôbec nedopracovali (možno na staršom type mobilu). Nebudeme sa zaoberať dôvodmi, vysvetlíme si radšej, ako zabezpečiť, aby sa stránky využívajúce media queries naozaj správne zobrazovali na všetkých obrazovkách pri akejkoľvek šírke.

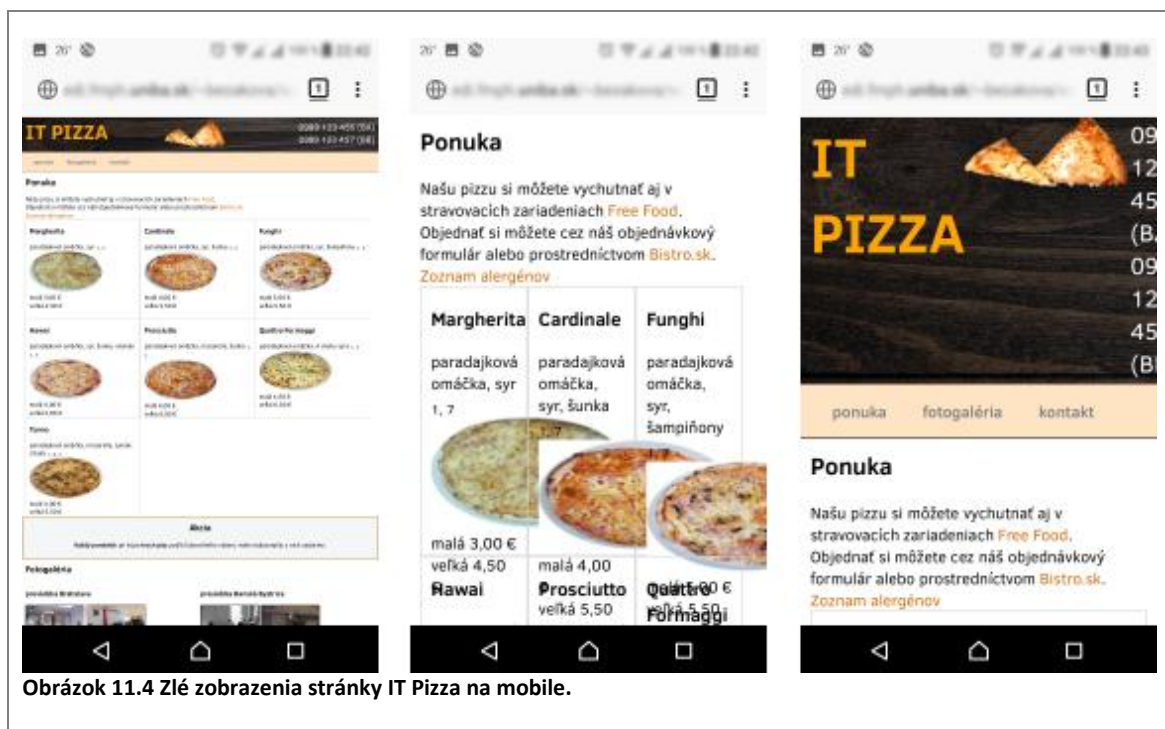
Nevyhnutnou súčasťou HTML kódu responzívnych stránok je element `<meta name="viewport" content="width=device-width, initial-scale=1.0">`, ktorý pridávame do elementu `head`. Tým povieme, že šírka stránky bude zodpovedať šírke zariadenia (nie rozlíšeniu!) a nastaví sa počítačová úroveň škálovania (zoom-u). Bez tohto meta elementu nebude zobrazovanie stránky na mobilných zariadeniach správne fungovať, aj napriek použitiu vhodných media queries.

### [Responzívny dizajn stránky IT Pizza](#)

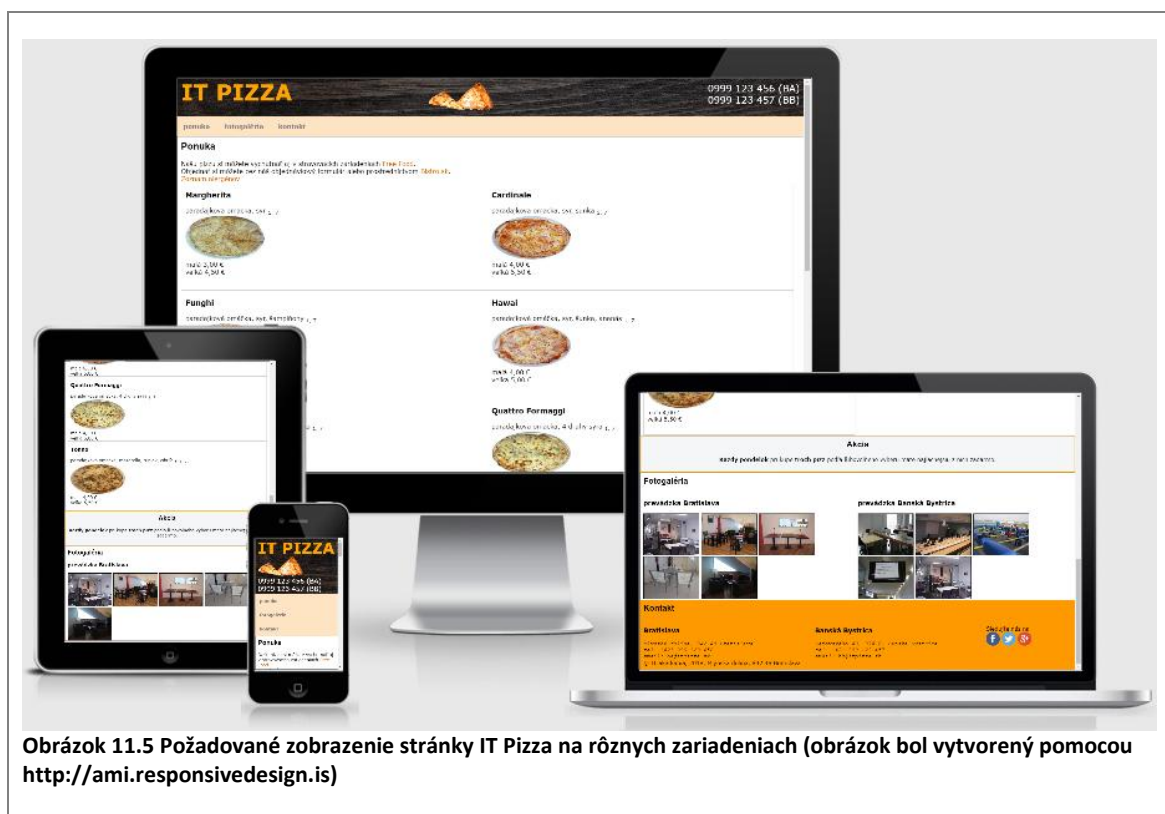
Až po 9. kapitole (vrátane) mala stránka IT Pizza v podstate jednostĺpcový dizajn. Keby sme do tejto verzie stránky pridali uvedený `meta` element `viewport`, zobrazovala by sa na mobiloch správne. Dá sa teda povedať, že sme stránku vytvárali prístupom *mobile first*.

V 10. kapitole sme pre stránku vytvorili viacstĺpcové rozloženie objektov: hlavičku, ponuku píz a päť sme rozdelili do troch stĺpcov, galériu do dvoch stĺpcov. Toto rozloženie sme realizovali dvoma spôsobmi, pomocou plávajúcich objektov a pomocou flexibilných boxov. Obe tieto riešenia, tak ako sú zatiaľ vytvorené, nie sú vhodné pre zobrazovanie stránky v mobile (*obrázok 11.4*). V lepšom prípade by prehliadač v mobile stránku zmenšil tak, že by texty boli nečitateľne malé (*obrázok 11.4 vľavo*). V horšom prípade by sa niektoré objekty prekryvali (pri riešení pomocou `float`, *obrázok 11.4 v strede*) alebo by niektoré prvky nebolo vidno (pri riešení pomocou flexibilných boxov, *obrázok 11.4 vpravo*) – to sme mohli vidieť aj na počítači, ak sme príliš zúžili okno prehliadača.

Upravíme stránku IT Pizza tak, aby bola responzívna: na zariadeniach s menšou zobrazovacou plochou sa objekty zobrazia v jednom stĺpci, na obrazovke so šírkou aspoň 800px sa rozložia do dvoch, resp. troch stĺpcov (*obrázok 11.5*). Budeme upravovať obe verzie stránky (`float` aj `flex`) kvôli tomu, že technológia flexibilných boxov je relatívne nová a staršími prehliadačmi nepodporovaná.



Obrázok 11.4 Zlé zobrazenia stránky IT Pizza na mobile.



Obrázok 11.5 Požadované zobrazenie stránky IT Pizza na rôznych zariadeniach (obrázok bol vytvorený pomocou <http://ami.responsivedesign.is>)

Teraz oceníme, že sme všetky vlastnosti, ktoré súvisia s umiestňovaním objektov (float, clear, nastavenia rozmerov, atď, resp. display:flex a s tým súvisiace vlastnosti) definovali samostatne. Vďaka tomu nebudeme musieť nastavovať žiadne nové štýly či pridávať vlastnosti, stačí len definície týchto “umiestňovacích” nastavení „obaliť” vhodným media-query.



## PRÍKLAD 11.5

Pomocou media queries upravíme stránku IT Pizza riešenú pomocou plávajúcich objektov (súbor 11/index-float.html) tak, aby sa len pri šírke aspoň 800px zobrazovala vo viacerých stĺpcoch. Všetky definície vlastností súvisiace s umiestňovaním (tie, čo sme vkladali za riadok /\* štýly súvisiace s viacstĺpcovým rozložením \*/) vložíme do media query @media screen and (min-width: 800px).

Najskôr pridajme do elementu head spomínaný dôležitý meta element:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>IT PIZZA</title>
  ...
```

Potom v elemente **style** definujme media query:

```
/* štýly súvisiace s viacstĺpcovým rozložením */
@media screen and (min-width: 800px) {
  header {
    height: 90px;
  }
  header h1 {
    float: left;
    width: 40%;
  }
  header img {
    float:left;
    width: 167px;
    padding-top: 10px;
  }
  header div {
    float:left;
    width: calc(60% - 167px);
    text-align: right;
    padding-top: 10px;
  }
  section article {
    float: left;
    width: calc((100% - 20px - 2*20px) / 2);
    height: 270px;
  }
  #galeria div {
    float: left;
    width: 50%;
  }
  .adresa {
    float: left;
    width: 40%;
  }
  #social {
    float:left;
    width: 20%;
  }
  nav, aside, footer, footer p {
    clear: left;
  }
}
```

Ako overíme, či je stránka naozaj responzívna? Ak máme stránku len na lokálnom počítači, testujeme ju ako doteraz pri rôznych veľkostiach okna prehliadača. V prípade, že už máme stránku umiestnenú na webovom serveri, môžeme použiť niektorý dostupný testovač responzivity, napr. Am I Responsive? (<http://ami.responsivedesign.is/>) alebo Mobile Friendly Test (<https://search.google.com/test/mobile-friendly>).

### ÚLOHA 11.6



Pomocou media queries upravte stránku IT Pizza riešenú pomocou flexibilných boxov (súbor 11/index-flex.html), aby sa pri šírke aspoň 800px zobrazovala vo viacerých stĺpcoch, inak v jednom. Nezapudnite na meta element!

Všimnite si, že v oboch spôsoboch riešenia (float aj flex) sa pri malej šírke odkazy nezobrazujú dobre (obrázok 11.6). Chceli by sme, aby sa zobrazovali pekne vedľa seba pri väčšej šírke a pod sebou pri menšej.



Obrázok 11.6 Zlé zobrazenie odkazov.

### PRÍKLAD 11.7



Upravme navigáciu na stránke IT Pizza tak, aby sa odkazy zobrazovali pod sebou pri šírke do 800px a vedľa seba pri šírke aspoň 800px (obrázok 11.7). Tentokrát si ukážeme riešenie pre verziu s flexibilnými boxami. Budeme vychádzať zo súboru z úlohy 11.6.

Znova využijeme flexibilný box. Jeho prvkom totiž dokážeme jednoducho nastaviť, či sa majú zobrazovať pod sebou alebo vedľa seba – pomocou vlastnosti `flex-direction`. Flexibilným boxom bude navigácia (z elementu `nav`). Jej prvky, teda odkazy, sa budú štandardne zobrazovať pod sebou (`flex-direction: column`) a pri šírke aspoň 800px vedľa seba (`flex-direction: row`). Kvôli krajšiemu vzhľadu ešte zrušíme `padding` pre element `nav`.

```
nav {
  background-color: #FFE4C4;
  border-bottom: 1px solid black;
  padding: 1em;
  display: flex;
  flex-direction: column;
}
...
/* štýly súvisiace s viacstĺpcovým rozložením */
@media screen and (min-width: 800px) {
  ...
  nav {
    flex-direction: row;
  }
}
```



Obrázok 11.7 Zvislá alebo vodorovná navigácia v závislosti od šírky stránky.

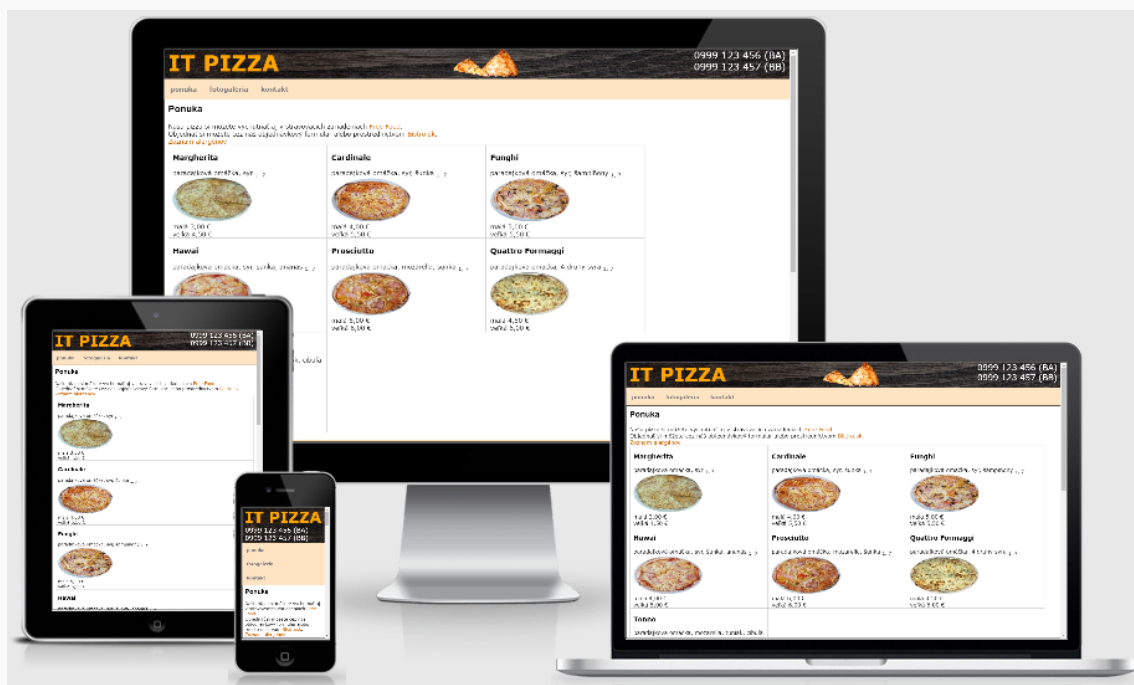
Zatiaľ sme rozlišovali dve rôzne zobrazenia stránky IT Pizza, pri šírke do 800px a pri šírke aspoň 800px. Pomocou media queries však môžeme definovať aj viacero rôznych verzií, napr. úzku, strednú a širokú.



### PRÍKLAD 11.8

Upravíme stránku IT Pizza z predchádzajúceho príkladu tak, aby:

- sa odkazy zobrazovali do šírky 400px pod sebou a od šírky 400px vedľa seba,
- sa logo stránky zobrazovalo až od šírky 800px,
- nadpis a telefónne čísla boli pod sebou pri šírke do 400px, v jednom riadku pri šírke aspoň 400px, pričom od šírky 400px sa medzi nimi zobrazí aj logo (obrázok 11.8).



Obrázok 11.8 Zvislá alebo vodorovná navigácia v závislosti od šírky stránky.

Zhrnieme, ako sa majú zobrazovať časti stránky pri vymedzených šírkach. Pomôže nám to pri rozhodovaní, čo dať do ktorého media query a do ktorého štýlu.



časť stránky/ šírka	ľubovoľná šírka	šírka aspoň 400px	šírka aspoň 800px
hlavička	bez loga, nadpis a telefónne čísla pod sebou	bez loga, nadpis a telefónne čísla v jednom riadku	nadpis, logo a telefónne čísla v jednom riadku
odkazy	pod sebou	vedľa seba	vedľa seba
zvyšok stránky	jednostĺpcový vzhľad	jednostĺpcový vzhľad	viacstĺpcový vzhľad

V časti s CSS kódom (v elemente `style`) budeme mať teda skupinu „základných štýlov“ pre zobrazovanie stránky v ľubovoľnej šírke a ďalšie dve skupiny štýlov v dvoch media queries, jednu pre zobrazovanie stránky v šírke aspoň 400px a druhú pre zobrazovanie stránky v šírke aspoň 800px. Skrytie loga vyriešime pomocou nastavenia `display: none` a jeho zobrazenie pomocou `display: inline`, pretože obrázok je štandardne riadkový element. V nasledujúcom kóde si všimnite aj komentáre vpravo:

```

...
header img {
  display: none;      /* v základnom štýle obrázok skryjeme */
}
nav {
  background-color: #FFE4C4;
  border-bottom: 1px solid black;
  display: flex;      /* navigácia bude flexibilným boxom */
  flex-direction: column; /* odkazy budú pod sebou v stĺpci */
}
...
/* štýly súvisiace s viacstĺpcovým rozložením */
@media screen and (min-width: 400px) {
  nav {
    flex-direction: row;      /* odkazy budú vedľa seba v riadku */
  }
  header {
    display: flex;           /* hlavička bude flexibilným boxom */
    justify-content: space-between;
    /* časti hlavičky budú vedľa seba oddelené medzerou */
  }
}
@media screen and (min-width: 800px) {
  header img {
    display: inline;        /* zobrazíme obrázok */
  }
  .flex {
    display: flex;
    justify-content: space-around;
  }
  #pizze {
    display: flex;
    flex-wrap: wrap;
  }
  section article {
    width: 380px;
  }
}

```



### ÚLOHA 11.9

Upravte stránku IT Pizza z *príkladu 11.5* (float verzia) tak, aby sa odkazy zobrazovali vedľa seba len pri šírke aspoň 800px, inak pod sebou. Nepoužívajte flexibilné boxy, ale vlastnosť `display` použiť môžete.

## 11.2 Štýl pre tlač

Webové stránky čítame zväčša priamo v mobile, či v počítači. Niekedy si však chceme stránku vytlačiť, napr. cestovný poriadok, recept na koláč, text piesne.



### DISKUTUJTE

Už ste niekedy tlačili webovú stránku? Zhodovala sa tlačaná verzia stránky s elektronickou? Myslíte, že je potrebné, aby sa stránka vytlačila v rovnakej podobe, ako vyzerá na počítači alebo v mobile? Ktoré časti stránky IT Pizza by podľa vás mali byť v jej tlačenej verzii?

V predchádzajúcej podkapitole sme definovali štýly pre obrazovky (screen) rôznych elektronických zariadení. Pomocou media queries dokážeme definovať aj tzv. **štýl pre tlač**, t.j. vieme ovplyvniť, ako bude vyzeráť stránka vytlačená na papieri.

Štýl pre tlač definujeme pomocou media-query: `@media print {}`, pričom v zátvorkách `{}` definujeme potrebné vlastnosti elementov. Treba si uvedomiť, že ak máme na stránke definovaný aj iný štýl, ktorý je platný pre všetky zariadenia, tak sa vlastnosti štýlov zlučujú, čo môže mať vplyv na konečné zobrazenie stránky.



### PRÍKLAD 11.10

Vytvorme štýl pre tlač stránky IT Pizza z *úlohy 11.9* (`11/index-float.html`). Najdôležitejšími informáciami na stránke je ponuka píz a kontakty. Pri tlači môžeme vynechať logo stránky, navigáciu, akciu, fotogalériu a kontakty na sociálne siete. Elementy prislúchajúce týmto častiam či prvkom teda v štýle pre tlač nezobrazíme, čo nám umožní nastavenie `display: none`. Textom v hlavičke (nadpis, telefónne čísla) nastavíme farbu na čiernu. Odkazom nastavíme farbu písma na sivú a podčiarkneme ich.

Do elementu style doplníme nasledujúci kód:

```
@media print {
  header img, nav, #galeria, aside, #social {
    display: none;
  }
  header h1, header div {
    color: black;
  }
  a {
    color: gray;
    text-decoration: underline;
  }
}
```

Štýl pre tlač otestujeme tak, že si zobrazíme ukážku stránky pred tlačou.

# IT PIZZA

0999 123 456 (BA)

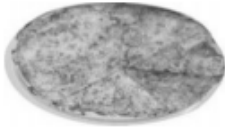
0999 123 457 (BB)

## Ponuka

Našu pizzu si môžete vychutnať aj v stravovacích zariadeniach [Free Food](#).  
Objednať si môžete cez náš objednávkový formulár alebo prostredníctvom [Bistro.sk](#).  
[Zoznam alergénov](#)

### Margherita

paradajková omáčka, syr 1, 7



malá 3,00 €

Obrázok 11.9 Stránka IT Pizza vo verzii pre tlač.

## ÚLOHA 11.11

Pre stránku `11/vitaminy.html` vytvorte štýl pre tlač, v ktorom iba skryjete navigáciu.

- Pozrite si ukážku pred tlačou.
- Diskutujte a skúmajte, prečo text v ukážke pred tlačou z úlohy 11.11 nevyužil celú šírku papiera (nešiel až po pravý kraj)?
- Čo treba zmeniť, aby bola využitá celá šírka papiera?

## ZAPAMÄTAJTE SI

V štýle pre tlač zvykneme:

- skrývať objekty, ktoré pre tlač nie sú dôležité, napríklad navigáciu, niektoré obrázky dekoratívneho či doplňujúceho charakteru, rôzne reklamy, ...
- nastaviť všetkým textom čiernu farbu písma,
- upraviť šírky potrebných elementov tak, aby sa pri tlači využil aj priestor, ktorý sa vytvoril skrytím iných elementov.

## 11.3 Externé štýly

Kaskádové štýly, ktoré sme zatiaľ definovali, by sme mohli označiť ako **interné kaskádové štýly**. Interné preto, lebo sme ich definovali priamo v tom súbore, v ktorom sme ich aj použili. Tento spôsob je vhodný vtedy, ak definovanú skupinu kaskádových štýlov používa len jediná stránka, ako napríklad naša stránka IT Pizza.

V prípade, že webové sídlo tvorí viacero stránok, ktoré majú mať rovnaký vzhľad, je použitie interných štýlov neefektívne. Všetky zmeny, ktoré by sme spravili v štýloch na jednej stránke, by sme totiž museli kopírovať aj na ostatné stránky. Preto sa pri tvorbe webových sídiel s viacerými stránkami využívajú **externé kaskádové štýly**. Externé znamená len to, že sú umiestnené v inom (externom) súbore, nie priamo v súbore s HTML kódom.

Externé kaskádové štýly môžeme písať v ľubovoľnom textovom editore, zväčša používame ten istý editor, v ktorom píšeme HTML kód. Definície externých štýlov, jednotlivé vlastnosti a ich hodnoty, zapisujeme úplne rovnako ako pri interných štýloch. Súbor so štýlmi musíme uložiť s príponou `.css`. Tento súbor nesmie obsahovať žiadne HTML značky.

Súbor s kaskádovým štýlom musíme „pripojiť“ ku všetkým HTML súborom, ktoré ho majú používať. To znamená, že v HTML kóde stránky musíme zadefinovať, že chceme, aby stránka používala daný súbor so štýlmi. Na to slúži element `<link>`, ktorý sa vkladá do elementu `<head>` v tvare: `<link href="subor.css" rel="stylesheet">`. Pomocou atribútu `href` určíme, aký súbor chceme pripojiť k webovej stránke.



### PRÍKLAD 11.12

Pre stránku so súťažami (`11/sutaze.html`) vytvoríme externý štýl `styl.css`, v ktorom nastavíme písmo pre celú stránku na `Arial` a farbu písma pre nadpisy úrovne 2 na tmavomodrú. Pripojíme štýl k stránke.

V editore, ktorý zvyčajne používame na písanie HTML kódu, vytvoríme nový súbor a napíšeme doň nasledujúci kód:

```
new 1 x
1 body {
2   font-family: Arial;
3 }
4
5 h2 {
6   color: darkblue;
7 }
```

Súbor uložíme ako `styl.css` do toho istého priečinka, v ktorom je umiestnený súbor `sutaze.html`. Do HTML kódu pridáme element `link` na pripojenie štýlu:

```
<head>
  <meta charset="utf-8">
  <title>Informatické súťaže</title>
  <link href="styl.css" rel="stylesheet">
</head>
}
```



### ÚLOHA 11.13

V priečinku `11/ucebnica` máme pripravené stránky s piatimi kapitolami učebnice (`kap1.html`, `kap2.html`, ..., `kap5.html`). a súbor `styly-uceb.css` so štýlmi pre učebnicu.

- Pripojte štýl zo súboru `styly-uceb.css` do všetkých kapitol učebnice.
- Skontrolujte, či všetky kapitoly učebnice vyzerajú rovnako.
- V súbore `styly-uceb.css` zmeňte nastavenie písma pre nadpisy na `Verdana` a súbor uložte.
- Overte, či sa písmo zmenilo vo všetkých kapitolách učebnice.

V predchádzajúcich častiach sme sa naučili vytvárať štýly pre rôzne zariadenia. Aj tieto môžeme vytvoriť ako externé štýly. V elemente `link` môžeme totiž určiť, pre aké zariadenie sa má uvedený štýl použiť, a to pomocou atribútu `media`. Atribút `media` môže nadobúdať hodnoty:

- `all` – štýly pre všetky zariadenia (prednastavená hodnota, ak atribút nevedieme),
- `screen` – štýly pre obrazovku,
- `projection, tv` – štýly pre projekciu a televízor,
- `print` – štýly pre tlač.

Napríklad:

- štýl pre ľubovoľnú obrazovku pripojíme pomocou elementu `<link href="styl.css" rel="stylesheet" media="screen">`,
- štýl pre tlač pripojíme pomocou elementu `<link href="styl.css" rel="stylesheet" media="print">`.

Okrem typu zariadenia môžeme v atribúte `media` definovať aj jeho vlastnosti, podobne ako v media queries. Napríklad:

- štýl pre obrazovky so šírkou aspoň `800px` pripojíme elementom `<link href="styl.css" rel="stylesheet" media="screen and (min-width: 800px)">`
- štýl pre obrazovky so šírkou najviac `1000px` pripojíme elementom `<link href="styl.css" rel="stylesheet" media="screen and (max-width: 1000px)">`

Ak používame na stránke viacero druhov štýlov pre rôzne zariadenia rôznymi vlastnosťami, ukladáme každý do samostatného súboru a k HTML súboru ich pripájame pomocou viacerých elementov `link`.

Vráťme sa k stránke IT Pizza. Jej zdrojový kód je už poriadne dlhý a začína byť neprehľadný. V nasledujúcich úlohách a príkladoch oddelíme CSS kód od HTML kódu, samostatne uložíme štýly pre jednotlivé zariadenia a šírky a pripojíme ich k HTML súboru.



### ÚLOHA 11.14

Štýly (t.j. obsah elementu `style`) zo stránky IT Pizza z príkladu 11.10 rozdeľte a uložte do štyroch samostatných súborov:

- základný štýl (to, čo nie je v žiadnom media query) uložte do súboru `stylyfloat.css`, do toho istého priečinka, v ktorom je súbor `index-float.html`,
- štýl pre obrazovky so šírkou aspoň 800px uložte do súboru `stylyfloat-min800.css`,
- štýl pre tlač uložte do súboru `stylyfloat-print.css`.
- V súbore `index-float.html` vymažte element `<style> </style>` s celým jeho obsahom.



### PRÍKLAD 11.15

V súbore `index-float.html` pripojíme CSS súbory vytvorené v úlohe 11.14.

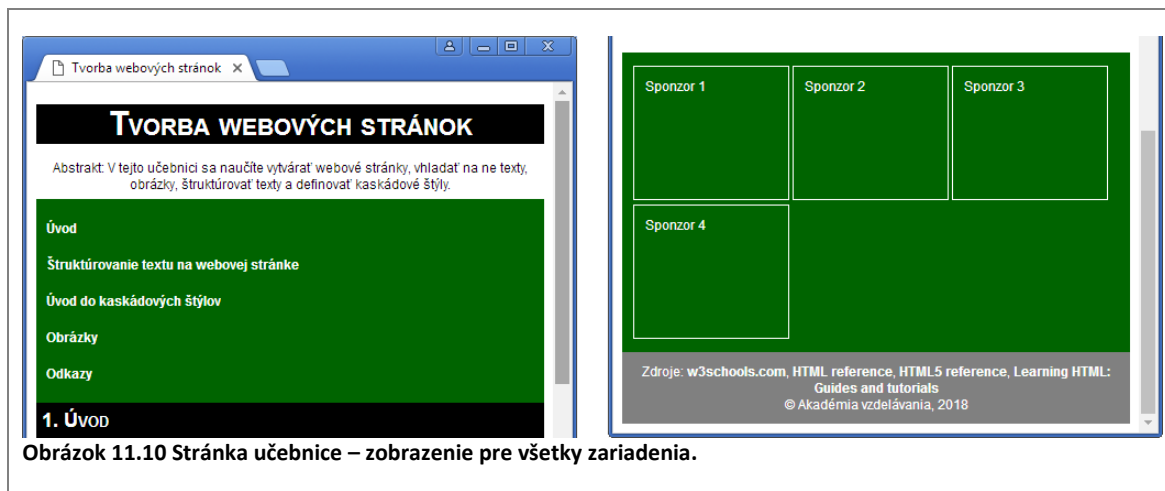
```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link href="stylyfloat.css" rel="stylesheet" media="screen">
  <link href="stylyfloat-min800.css" rel="stylesheet" media="screen
and (min-width: 800px)">
  <link href="stylyfloat-print.css" rel="stylesheet" media="print">
  <title>IT PIZZA</title>
</head>
```



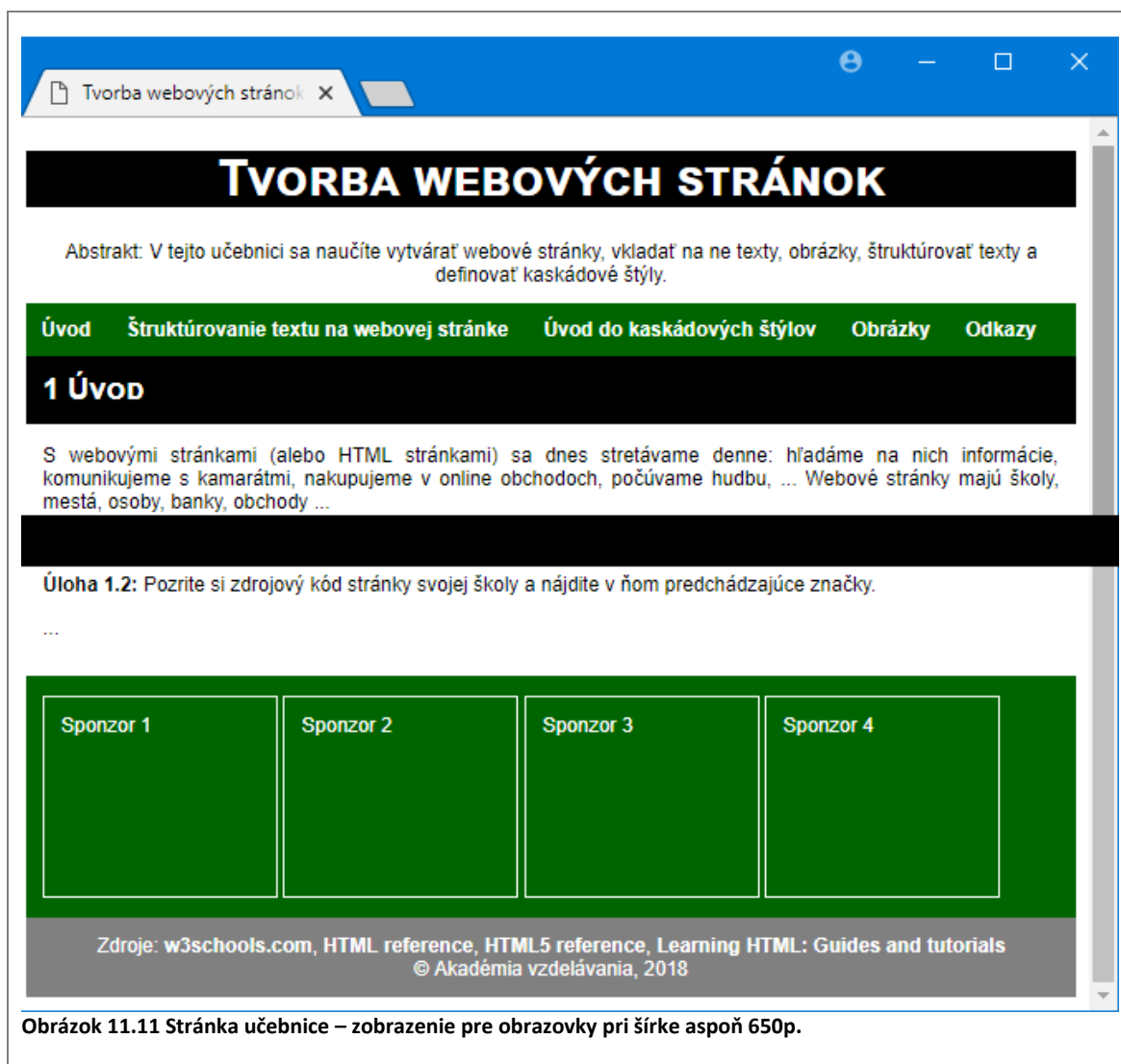
### ÚLOHA 11.16

Upravte stránku s učebnicou (súbory v priečinku 11/učebnica) tak, aby bola responzívna. Definujte tiež štýl pre tlač. Všetky štýly definujte ako externé súbory.

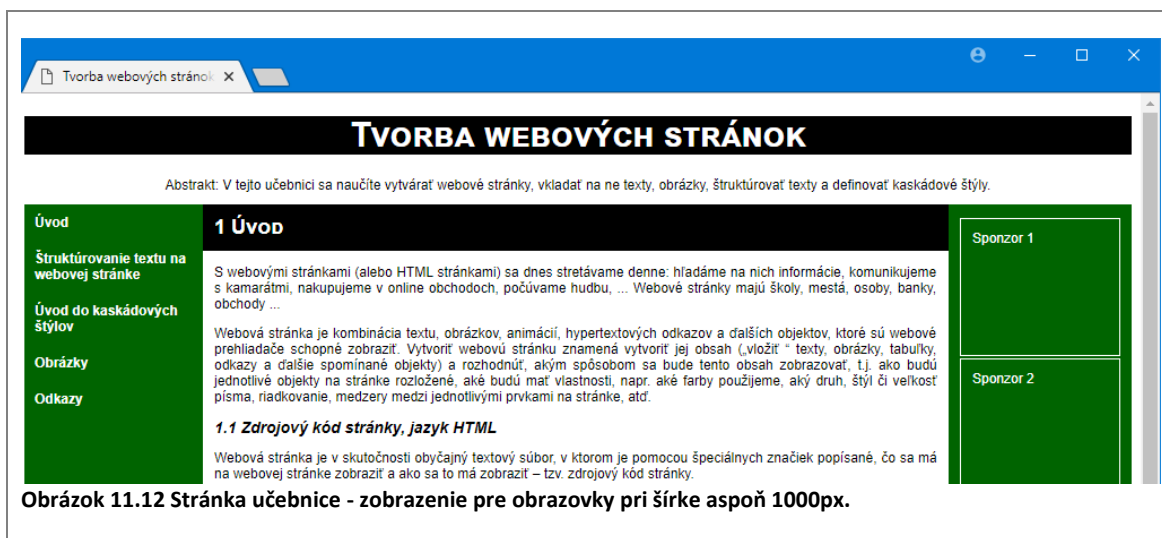
- základné zobrazenie bude v jednom stĺpci, odkazy v navigácii budú pod sebou, sponzori vedľa seba koľko sa zmestí do riadka (*obrázok 11.10*),
- pri šírke aspoň 650px: vzhľad je stále jednostĺpcový, sponzori aj odkazy sa zobrazujú vedľa seba (*obrázok 11.11*),
- pri šírke aspoň 1000px: vzhľad je trojstĺpcový, vľavo sú odkazy pod sebou, v strede je text kapitoly, vpravo sú sponzori pod sebou (*obrázok 11.12*),
- tlačíť sa bude len hlavný nadpis, text kapitoly a päta. Nadpisy a texty v päte budú čierne, odkazy sivé, podčiarknuté a nie tučné (*obrázok 11.13*).



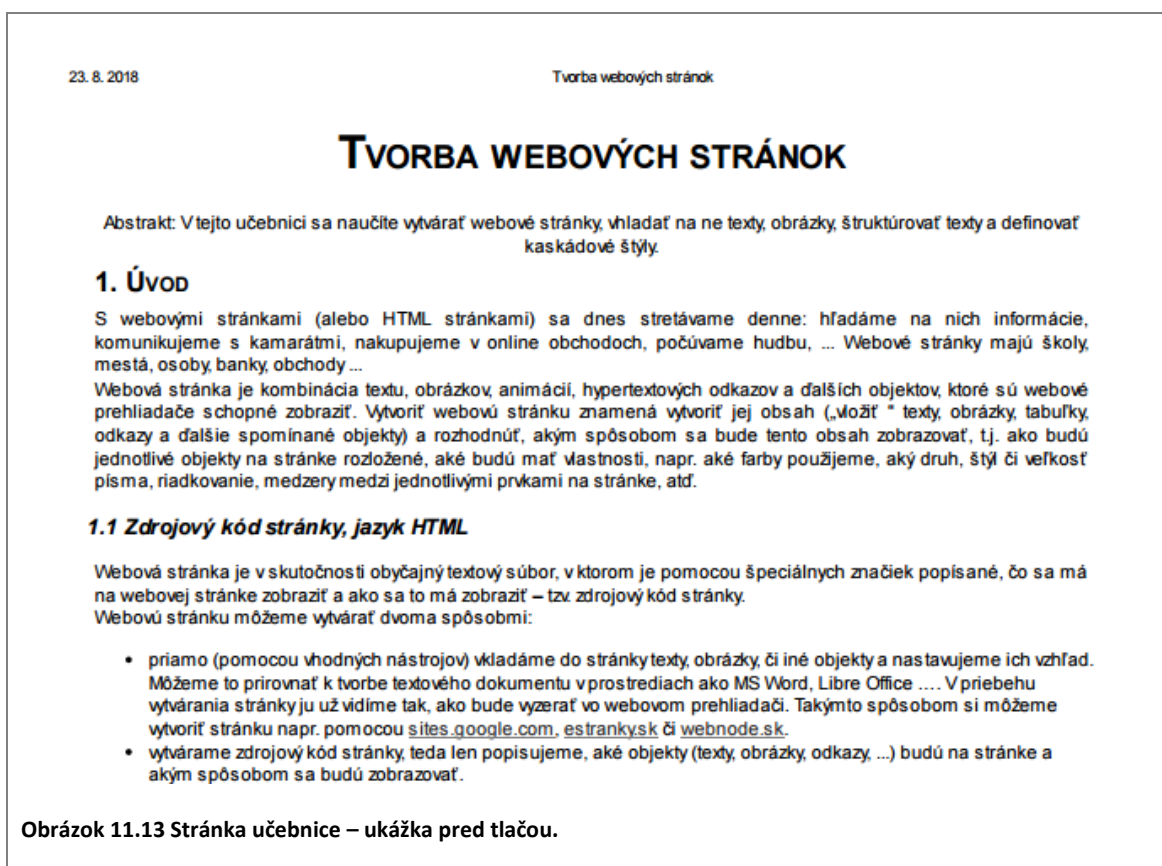
Obrázok 11.10 Stránka učebnice – zobrazenie pre všetky zariadenia.



Obrázok 11.11 Stránka učebnice – zobrazenie pre obrazovky pri šírke aspoň 650p.



Obrázok 11.12 Stránka učebnice - zobrazenie pre obrazovky pri šírke aspoň 1000px.



Obrázok 11.13 Stránka učebnice – ukážka pred tlačou.



## 11.4 Metodika pre učiteľa

### CIEĽ

Cieľom je pochopiť pojem responzívny web a oboznámiť sa so spôsobmi, ako ho tvoriť.

**Nadväznosť na predchádzajúce kapitoly:** Kapitola vyžaduje kapitoly 1-8 a 10.

Je to pomerne náročná kapitola. Isté princípy sa dajú ukázať na príkladoch 1 až 4 a časti 11.2, ktoré sú jednoduchšie a nie sú viazané na kap. 10. Zvyšok odporúčame skôr len pre pokročilých.

**K syntaxi media queries:** Syntax media queries môže byť aj zložitejšia, pozrite na [https://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](https://www.w3schools.com/cssref/css3_pr_mediaquery.asp)

**Riešenie úlohy 11.6:**

```
@media screen and (max-width: 800px) {  
  ...  
  .adresa, #social, #galeria div {  
    float: none;  
    width: 100%;  
  }  
}
```

**Úloha 11.9:** **Riešenie:** Vo všeobecných štýloch spravíme z odkazov v navigácii bloky, `nav a {display: block}` a v media query pre väčšiu šírku zasa inline-bloky. `nav a {display: inline-block}`.

**Doplňujúca úloha (za úlohou 11.9):** Upraviť stránku index-float z úlohy 9 tak, aby mala až tri verzie: základnú jednotípcovú, strednú (pri šírke aspoň 400px), v ktorej by sa zobrazovali v ponuke dve pizze vedľa seba a širokú (pri šírke aspoň 800px), v ktorej by sa zobrazovali tri pizze vedľa seba.

**K testovaniu responzivity:**

Pre potreby učebnice sme využívali testovače:

- Am I responsive - (<http://ami.responsivedesign.is>) - výhodou je, že vie testovať aj lokálne stránky, ale len cez localhost
- a Mobile Friendly Test (<https://search.google.com/test/mobile-friendly>)

Prácu s týmito testovačmi v učebnici nepopisujeme, je intuitívna a jednoduchá.

Zoznam ďalších aplikácií na testovanie responzivity nájdete na stránke: <https://www.webdesignerdepot.com/2017/10/7-free-tools-for-testing-responsive-layouts/>

**Úloha 11.10:** V úlohe sme riešili iba skrytie niektorých objektov. Môžete sa ešte so žiakmi porozprávať, či je lepšie tlačiť úzke rozloženie stránky (ako pre mobily) alebo široké. Prípadne

experimentovať a pokúsiť sa vytvoriť také rozloženie objektov na IT Pizza stránke, aby ste spotrebovali čo najmenej papiera, ale aby sa potrebné informácie na papier zmestili.

**Úloha 11.11:** Stránka vitamíny sa skladá z dvoch stĺpcov, ktoré sú vytvorené pomocou plávajúcich objektov. Ľavý stĺpec s navigáciou, má nastavenú šírku 25%, pravý stĺpec s hlavným obsahom má šírku zvyšných 75%. Ak v štýle pre tlač vypneme navigáciu, hlavný obsah sa síce posunie vľavo, ale jeho šírka zostáva 75%, a preto text pri tlači nezaberá celú šírku papiera. Vyriešime to nastavením šírky 100% pre pravý stĺpec v štýle pre tlač.

**Príklad 11.12:** Upozorniť, že ak budú teraz chcieť žiaci kopírovať stránku na iný počítač, musia skopírovať všetky s ňou súvisiace súbory, teda nielen html, ale aj css.

**Úloha 11.16** je opakovacia. Pravidlá, čo ako zobrazovať pri jednotlivých šírkach, si môžu žiaci vymyslieť aj sami.

**Zdroje k tejto kapitole:**

- W3 schools - časť HTML responsive - to je pre plávajúce objekty
- W3 schools - časť CSS Flexbox (na konci) - pre flexový dizajn
- <https://www.alejtech.sk/sk/blog-o-webdizajne/responzivny-dizajn-standard-pri-tvorbe-novych-webov.html>
- <https://www.websupport.sk/blog/2013/11/aky-je-to-responzivny-web/>
- <http://www.run.sk/responzivny-dizajn>