

IX. Další techniky s HTML a CSS

Responzivní web

- pod tímto pojmem je myšleno, že daný web se dokáže snadno přizpůsobit uživateli, tj. například jeho hardveru, který používá pro přístup na web
- dříve se vyvíjel web ve více verzích, jedna pro klasické prohlížeče na pc, druhá pro mobilní zařízení, případně další pro tablety
- s přibívajícím počtem zařízení pro práci s webem je tento přístup nešťastný

Komponenty flexibilního webu

- flexibilní obrázky a média. Aktiva jsou dimenzována pomocí procent, takže se zvětšují a zmenšují v prostoru, který mají k dispozici
- flexibilní ("tekuté") rozvržení založené na mřížce. Responzivní webová stránka má všechny šířky nastaveny v procentech, tak aby se komponenty rozvržení mohly zmenšovat nebo zvětšovat. Další horizontální vlastnosti obvykle také používají relativní jednotky (em, procenta nebo rem).
- *media queries*. Jejich přidání do stylů vám umožní upravit vaše design stránky na základě šířky zobrazitelné oblasti stránky prohlížeče a dalších charakteristik.

Flexibilita obrázků

- vynechání `width` a `height` v HTML kódu
- přidání CSS stylu s `max-width:100%;` (pro vše flexibilní)

Flexibilní mřížkové rozvržení

- pro prvky, u nichž je třeba zadat šířku, udej ji relativně vůči elementu, ve kterém vystupují
- nepoužívejte šířku `width: 100%;`
- uveď `max-width: value;` pro prvek obsahující celý obsah stránky (hodnota např. 60em odpovídá 960 px při fontu 16)

Media queries

- CSS3 modul umožňující adaptabilní vykreslování webových stránek podle různých činitelů jako rozlišení obrazovky, či velikost obrazovky
- přidáme pravidlo `@media` do css souboru

```
/* Styles for print only */  
@media print {  
  header[role="banner"] nav, .ad {  
    display: none;  
  }  
}
```

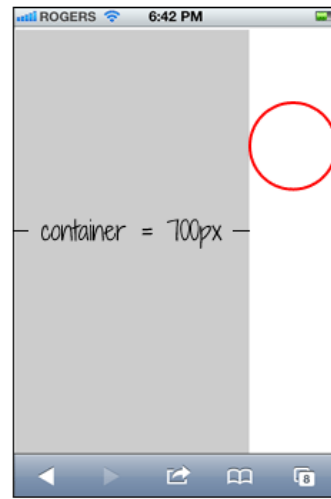
- do media queries lze zakomponovat vlastnosti:
- width, height, device-width, device-height, orientation, aspect-ratio, device-aspect-ratio, color, color-index, monochrome, resolution, scan, grid
- u většiny mimo orientation, scan, a grid lze přidat prefix **min-** (hodnoty musí být větší nebo rovny) **max-** (hodnoty menší nebo rovny)

```
@media logic type and (feature: value){  
  /* your targeted CSS rules go  
  → here */  
}
```

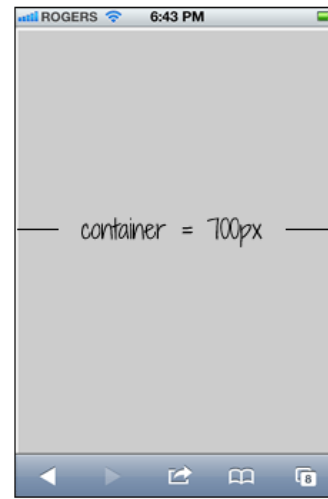
- komponenta `logic` :
 - nepovinná, nabývá hodnot `only` nebo `not`
- komponenta `type` :
 - typ media, např. `screen`, `print`
- komponenta `feature: value`
 - nepovinná, musí být v závorkách `()` před nimiž je slovo `and`
 - další vlastnosti přidáváme řetězením `and` a závorek `()`
- list media queries vytváříme přidáním čárek

Viewport

- oblast v mobilních prohlížečích, kde se zobrazuje obsah stránek
- media query `width` mapuje na šířku viewportu, jehož šířka je jiná než `device-width`
- přidáme do `<head> ... </head>`
`<meta name="viewport" content="width=device-width, initial-scale=1" />`



without viewport tag



viewport width=720

Efekty s CSS

- vylepšení grafické podoby webu se dříve dělalo přidáním či úpravou obrázků. To často mělo za následek, že web byl více komplikovaný a trvalo déle ho načíst.
- v CSS se tomu můžeme "snadno" vyhnout a udělat web přehlednější a rychlejší k načtení
- starší prohlížeče jako IE8 nemusí všechny tyto funkce podporovat

Zaoblování rohů

- zaoblit nejdou jenom obrázky, ale i kusy textu jako je odstavec
- tato vlastnost se nedědí

Do kruhu

- vlastnost `border-radius:r;`, kde r je poloměr pro zaoblední rohů
- pro zaoblení jen určitých rohů použijeme `border-top-left-radius: r;`
(zaoblí se jen horní levý roh)
 - další rohy `top-right`, `bottom-left`, `bottom-right`

```
.about img {  
  border: 5px solid #bebebe;  
  border-radius: 15px;  
}
```

Do elipsy

- vlastnost `border-radius: x / y;`, kde x a y jsou horizontální a vertikální poloměry

Kruhový obrázek

- vlastnost `border-radius: r;`, kde r musí být meší nebo rovno minimu z výšky a délky obrázku

```
.example-1 {  
  /* Makes the radius of the top-left and bottom-right corners 10px and the  
  top-right and bottom-left corners 20px */  
  border-radius: 10px 20px;  
}  
.example-3 {  
  /* Makes the radius of the top-left corner 10px, the top-right corner  
  20px, the bottom-right corner 0, and the bottom-left corner 30px */  
  border-radius: 10px 20px 0 30px;  
}
```

Stín textu

- vlastností `text-shadow` přidáme stíny k textu v nadpisech, případně odstavcích
- hodnoty `x-offset y-offset blur-radius color` (**neoddělit** čárkou), hodnota `blur-radius` je volitelná
- pro přidání více stínů, oddělíme je čárkami
- tato vlastnost se dědí

```
.multiple {  
  text-shadow:  
    2px 2px white,  
    6px 6px rgba(50, 50, 50, .25);  
}
```

Stínování prvků

- vlastnost `box-shadow`
- oproti `text-shadow` máme možnost přidat hodnotu `inset` (stíny dovnitř), `spread` hodnotu zúžení
- pokud chceme definovat různé stíny pro daný prvek, v definici je oddělíme čárkou
- tato vlastnost se nedědí

```
.inset-shadow {  
  box-shadow: 2px 2px 10px #666 inset;  
}  
.multiple {  
  box-shadow: 2px 2px 10px rgba(255, 0, 0, .75),  
             5px 5px 20px blue;  
}
```

Vícero pozadí

- CSS3 zavádí možnost aplikovat více pozadí, čímž se zjednoduší samotný HTML dokument
- začneme s `background-color: b` pro základní barvu. To ocení uživatelé starých prohlížečů bez podpory CSS3
- následuje `background-image: u`, kde u je čárkami oddělený seznam url s adresami obrázků
- následuje `background-position: p`, kde p je čárkami oddělená množina offsetů (`x-offset y-offset`) pro každý obrázek na pozadí
- následuje `background-repeat: r`, kde r je čárkami oddělený seznam s položkami `repeat-x`, `repeat-y` nebo `no-repeat`

- alternativně

```
.night-sky {  
/* fallback color and img */  
background: navy url(ufo.png) no-repeat center bottom;  
background:  
    url(ufo.png) no-repeat 50% 102%,  
    url(stars.png) no-repeat 100% -150px,  
    url(stars.png) no-repeat 0 -150px,  
    url(sky.png) repeat-x 50% 100%;  
}
```


Barevný gradient na pozadí

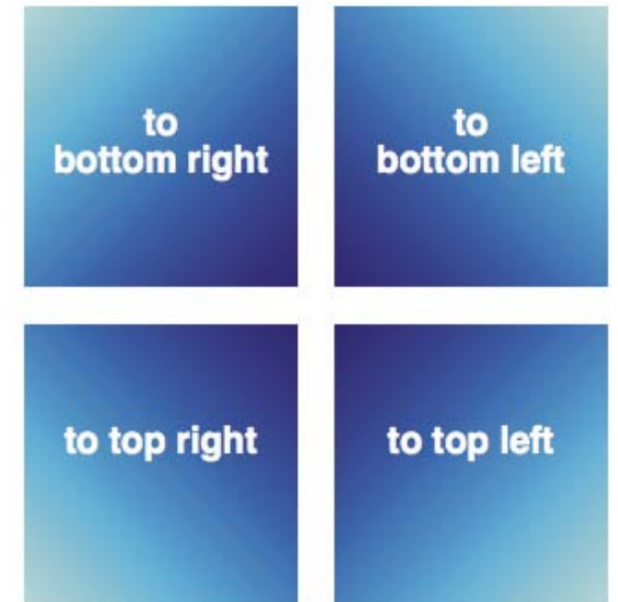
- opět vlastnost zavedená CSS3
- dva základní styly: lineární, gradientní
- nejednoznačná definice stylu mezi prohlížeči (třeba přidat vendor-prefix), proto je dobré přidat nějaký výchozí pozadí bez těchto efektů

Lineární gradient

- `linear-gradient(direction, color specification)`
 - `direction`
 - není povinný, výchozí hodnota je `to bottom`, jiné hodnoty `to top`, `to bottom left`, `to top right`, ...
 - případně lze definovat pomocí úhlu gradientu (`45deg`, `107deg`, ...)

Př: pomocí směru

```
.diagonal-bot-rt {  
  background: aqua;  
  background: linear-gradient( to bottom right, aqua, navy);  
}  
.diagonal-bot-lt {  
  background: aqua;  
  background: linear-gradient( to bottom left , aqua, navy);  
}  
.diagonal-top-rt {  
  background: aqua;  
  background: linear-gradient( to top right, aqua, navy);  
}  
.diagonal-top-lt {  
  background: aqua;  
  background: linear-gradient( to top left , aqua, navy);  
}
```



Př: pomocí úhlu

```
.angle-120deg {  
  background: aqua;  
  background: linear-gradient( 120deg ,aqua, navy);  
}  
.angle-290deg {  
  background: aqua;  
  background: linear-gradient( 290deg ,aqua, navy);  
}
```



Radiální gradient

- `radial-gradient(shape size position, color);`
 - `shape`
 - není povinný údaj, hodnoty `circle`, `ellipse`
 - `size`
 - není povinný údaj, jedna hodnota nebo dvě hodnoty v jednotkách délky určující délku gradientu v x-ovém a y-ovém směru (výchozí `farthest-corner`)
 - případně hodnoty `closest-side`, `farthest-side`, `closest-corner`, `farthest-corner`
 - `position` s hodnotami `at top`, `at right`, `at bottom left`, ..., případně dvojice souřadnic `at 200px 43px`

```
.radial-various-1 {  
  background: red;  
  background: radial-gradient( closest-side at 70px 60px, yellow, lime, red );  
}  
.radial-various-2 {  
  background: red;  
  background: radial-gradient( 30px 30px at 65% 70%, yellow, lime, red );  
}
```



```
.color-stops-1 {  
  background: green;  
  background: linear-gradient(yellow 10%, green);  
}  
.color-stops-2 {  
  background: green;  
  background: linear-gradient(to top right, yellow, green 70%, blue);  
}
```



Specifikace barev

- je třeba uvést alespoň dvě barvy oddělené čárkou. První barva je startovní (vnitřní), druhá koncová (vnější).
- lze definovat více barev s procentem určující, kde v rámci gradientu bude daná barva vystupovat

Nastavení průhlednosti prvků

- vlastnost `opacity: 0;`, kde 0 je míra průhlednosti
 - výchozí hodnota je 1 (100%)
 - jinak hodnoty mezi `0 - 1.00`
- ovlivňuje i text, ne jenom pozadí

```
.box {  
  background: #000; /* black */  
  opacity: .5;  
  padding: 20px;  
  width: 420px;  
}
```


Vygenerování speciálního obsahu

- pseudoprvky `:before` a `:after` lze použít s vlastností `content` pro přidání generovaného obsahu
- negenerujeme text, nýbrž speciální prvky a symboly

```
<p>This area is one of the most tranquil spaces at the Villa.  
I couldn't help but think &hellip;  
<a href="victoria.html" class="more"> Read More</a></p>
```

```
.more:after {  
  content: " »";  
}
```

help but think ... [Read More »](#)

Seznamy s HTML a CSS

- seznamy obvykle mají různé typy odrážek nebo způsob řazení (číselné, abecední, ...)
- typy seznamů:
 - seřazený seznam (`ol` pro rodiče, `li` pro prvky)
 - neseřazený seznam (`ul` pro rodiče, `li` pro prvky)
 - seznam s popisem jednotlivých prvků (`dl` pro rodiče, `dt` označuje popisovaný pojem, `dd` popis)
- neseřazený seznam se hodně používá pro navigační list
- typický problém v čistém HTML je například neestetická mezeta mezi naspisem a seznamem -> CSS

```
<ol>
  <li>Make sure you h ... wall socket.</li>
  <li>Unscrew the old bulb.</li>
  <li>Get the new ... package.</li>
  <li>Check the ... it's correct.</li>
  <li>Screw in the new bulb.</li>
  <li>Plug in the ... it on!</li>
</ol>
```

1. Make sure you have unplugged the lamp from the wall socket.
2. Unscrew the old bulb.
3. Get the new bulb out of the package.
4. Check the wattage to make sure it's correct.
5. Screw in the new bulb.
6. Plug in the lamp and turn it on!

Výběr značek (markerů)

- pro daný list si můžeme vybrat jednu z následujících značek definováním `list-style-type: marker`, kde marker je
 - `disc`, `circle`, `square`, `decimal`, `upper-alpha`, `lower-alpha`, `upper-roman`, `lower-roman`, `none`
 - další značky např. na <http://www.quirksmode.org/css/lists.html>
- typ značky lze definovat i přímo v HTML souboru pomocí `<ol type="I">`
- vlastní typ značek:
 - v daném listu nastavíme `list-style: none;` a dále nastavíme `margin-left` anebo `padding-left` (nastavíme správné odsazení)
 - dále v pravidle pro `li` nastavíme `background: url(image.ext) repeat-type horizontal vertical;`, dále nastavíme `padding-left: value;` (nejméně na šířku daného obrázku)

```
ul{
  list-style: none;
  margin-left: 0;
  padding-left: 0;
}
li{
  /* show image slightly down from top
    of item. */
  background: url(../img/checkmark.png)
              no-repeat 0 .1em;
  /* make line tall enough to show full
    checkmark */
  line-height: 1.8;
  /* bump the text over to make room
    for the checkmark */
  padding-left: 1.75em;
}
```

Product Features

- ✓ One-click page layout.
- ✓ Spell-checker for 327 languages.
- ✓ Image retouching tool.
- ✓ Unlimited undos and redos.

- alternativa `list-style-image: url(marker.png);`
 - ale problém s konzistencí mezi prohlížeči

Definice začátku číslování

- přidáme `start="n"` k počátečnímu tagu daného seznamu, kde `n` označuje počáteční hodnotu seznamu
 - `n` je vždy číselné, i když číslujeme podle abecedy
- pokud chceme změnit hodnotu daného prvku seznamu, přidáme `value="n"` k počátečnímu tagu prvku seznamu
 - následující označení prvků seznamu jsou pak náležitě změněny

Změna pozice značek seznamu

- defaultně jsou značky umístěné vlevo mimo text seznamu zarovnané podle okolního textu
- tuto vlastnost lze změnit použitím `list-style-position` s hodnotou
 - `inside` (značky jsou umístěné uvnitř textu seznamu), `outside` (zarovnání značek s okolním textem, výchozí nastavení)
- tuto vlastnost lze definovat i konkrétním prvkům daného seznamu

Nastavení vnořených seznamů

- vnější seznam definujeme pomocí `toplevel {style_rules}`, pro druhou úroveň použijeme `toplevel 2ndlevel {style_rules}`, pro třetí úroveň použijeme `toplevel 2ndlevel 3rdlevel {style_rules}`, atd.

```
ol{  
  list-style-type:upper-roman;  
}  
ol ol{  
  list-style-type: upper-alpha;  
}  
ol ol ol{  
  list-style-type: decimal;  
}  
ol li{  
  font-size: .875em;  
}
```

- typickým použitím je strukturování navigačního panelu

Popisné seznamy

- příklad Věta/Definice
- prvek pro seznam `<dl>`, prvky daného seznamu `<dt>` pro title a `<dd>` pro popis
- v `<dt>` je dobré přidat kolem jmen `<dfn>` pro zvýraznění, že daný seznam něco definuje
- pro jednotlivé elementy můžeme vyrvořit vlastní css styl

```
<dl>
<dt>Boris Karloff</dt>
  <dd>Best known for his role in <cite>Frankenstein</cite>
    and related horror films, this scaremaster's real name
    was William Henry Pratt.</dd>
<dt>Christopher Lee</dt>
<dd>Lee took a bite out of audiences as Dracula in multiple
  Hammer horror classics.</dd>
... [more scary legends] ...
</dl>
```

Formuláře s HTML a CSS

- představuje první interakci s uživatelem
- budeme uvažovat podporu vlastností z HTML5
 - pro starší prohlížeče JavaScript (INTA)
 - informace od podpoře na <https://caniuse.com/>
- jednotlivé prvky formuláře stylizujeme pomocí CSS

Základní syntax

- prvek `<form>...</form>` , v rámci něhož definujeme prvky daného formuláře
- každý formulář by měl mít prvek pro odeslání daného formuláře

```

<form method="post" action="show-data.php">
<!-- Various form elements -->
<fieldset>
    <h2 class="hdr-account">Account</h2>
    <div class="fields">
        <p class="row">
            <label for="first-name">First Name:</label>
            <input type="text" id="first-name" name="first_name" class="field-large" />
        </p>
        <p class="row">
            <label for="last-name">Last Name:</label>
            <input type="text" id="last-name" name="last_name" class="field-large" />
        </p>
        ...
    </div>
</fieldset>
... more form elements ...
<!-- Submit Button -->
    <input type="submit" value="Create Account" class="btn" />
</form>

```

Atributy prvku `form`

- základní `action`, `method`
 - `action="save-info.php"` url skriptu, který zpracuje daný formulář
 - hodnoty `method`:
 - `get` - údaje formuláře se zobrazí v adresním řádku prohlížeče po odeslání formuláře
 - `post` - informace ve vašem formuláři se nezobrazují v adrese URL po odeslání formuláře
 - v případě pochybností použijte funkci `post`, aby data nebyla vystavena v adrese URL

Zpracování formuláře

- třeba vytvořit skript v "nějakém" jazyce pro zpracování získaných údajů
 - většinou v PHP díky velké škále příkladů, ...
 - alternativa Django (python), Ruby on Rails, ASP.NET, JSP
- v rámci zpracování provádíme kontrolu vyplnění (vše a jak má)
 - některé prvky mají již vlastní validaci
- třeba dbát na bezpečnost a zabezpečení dat
 - viz. např. <https://css-tricks.com/serious-form-security/>

Organizace prvků formuláře

- prvek `fieldset` pro seskupení prvků formuláře
- prvek `legend` pro přidání nadpisu k dané skupině (alternativně přes nadpisy `h1` - `h6`)
 - musí hned následovat za `<fieldset>`

```
<fieldset>
  <h2 class="hdr-public-profile">Public Profile</h2>
  ... Public Profile fields ...
    <div class="row">
      <fieldset class="radios">
        <legend>Gender:</legend>
        <input type="radio" id="gender-male" name="gender" value="male" />
        <label for="gender-male">Male</label>
        <input type="radio" id="gender-female" name="gender" value="female" />
        <label for="gender-female">Female</label>
      </fieldset>
    </div>
  </fieldset>
  <fieldset>
    <h2 class="hdr-emails">Emails</h2>
    ... Emails fields ...
  </fieldset>
```


Výtvor textových boxů

- typicky používané pro získání informací typu jméno, adresa, ...
- prvek se značkou `input` a atributou `type="text"`
 - další atributy `name` (používané skripty na straně serveru pro načtení daných hodnot), `id`, `required`, `aria-required`, `placeholder="hinttext"` (text uvedený na počátku), `autofocus` (při načtení stránky se kurzor zaměří do daného pole), `size="n"` (šířka boxu v písmenech), `maxlength="n"`, `autocomplete="off"` (pro zamezení automatického vyplňování), `value="default"` (výchozí hodnota)
 - před `<input>` můžeme přidat label s atributou `for` jejíž hodnota se pak shoduje s hodnotou atributy `id` u prvku `input`

```
<h2 class="account">Account</h2>
<div class="fields">
  <p class="row">
    <label for="first-name">First Name:</label>
    <input type="text" id="first-name" name="first_name"
      class="field-large" required aria-required="true" />
  </p>
  <p class="row">
    <label for="last-name">Last Name:</label>
    <input type="text" id="last-name" name="last_name"
      class="field-large" />
  </p>
  <p class="row">
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"
      placeholder="yourname@example.com" class="field-large" />
  </p>
  ... more fields ...
</div>
```

Boxy pro heslo

- rozdíl oproti boxu pro text je, že zadávaný text je defaultně schován

```
<p class="row">
  <label for="password">Password:</label>
  <input type="password" id="password" name="password" />
</p>
<p class="row">
  <label for="password2">Re-enter Password:</label>
  <input type="password" id="password2" name="password2" />
</p>
```

Boxy na email, url, telefon a vyhledávání

- podporováno do HTML5, starší prohlížeče zobrazí jako textový boxy
- `type="email"` , `type="url"` , `type="tel"` , `type="search"`
- `pattern="\d{3}-\d{3}-\d{4}"` atributa sloužící k validaci obsahu
 - <https://www.html5pattern.com/Phones>
- Chrome, Firefox, IE10, and Opera 10+ podporují automatickou validaci obsahu těchto boxů

```
<p class="row">
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" class="field-large" />
</p>
<div class="row">
  <label for="website">Website URL:</label>
  <input type="url" id="website" name="website" class="field-large"
    placeholder="http://www.example.com" />
    <p class="instructions">
      Have a site or a blog? Put the address
      here beginning with <kbd>http://</kbd> or <kbd>https://</kbd>.
    </p>
</div>
<p class="row">
  <label for="phone">Phone:</label>
  <input type="tel" id="phone" name="phone" class="field-large"
    placeholder="xxx-xxx-xxxx" pattern="\d{3}-\d{3}-\d{4}" />
</p>
```

Přepínače (radio buttons)

- atributa `type="radio"`
- přepínače v jené skupině musí mít stejnou atributu `name="someName"`
- atributa `checked` pro defaultní zaškrtnutí

```
<fieldset class="radios">
<legend>Gender:</legend>
<p class="row">
  <input type="radio" id="gender-male" name="gender" value="male" />
  <label for="gender-male">Male</label>
</p>
<p class="row">
  <input type="radio" id="gender-female" name="gender" value="female" />
  <label for="gender-female">Female</label>
</p>
</fieldset>
```

Checkboxes

- oproti přepínačům umožňují více voleb
- atributa `value="data"` , kde data jsou odeslány serveru při zaškrtnutí

```
<div class="fields checkboxes ">
<p class="row">
  <input type="checkbox" id="email-ok-msg-from-users"
    name="email_signup[]" value="user-emails" />
    <label for="email-ok-msg-from-users">
      It is okay to email me with messages from other users.
    </label>
</p>
<p class="row">
  <input type="checkbox" id="email-ok-occasional-updates"
    name="email_signup[]" value="occasional-updates" />
    <label for="email-ok-occasional-updates">
      It is okay to email me with occasional promotions about
      our other products.
    </label>
</p>
</div>
```

Rámy pro text

- typicky pro širší text jako komentáře, poznámky, ...
- element `<textarea>`
- dodatečné atributy `maxLength="n"`, `cols="n"`, `rows="n"`

```
<label for="bio">Bio:</label>  
<textarea id="bio" name="bio" cols="40" rows="5" class="field-large"></textarea>
```


Selektivní boxy

- perfektní pro definici voleb pro uživatele
- často se rendrují jako seznamy rozbalené dolů
- prvek `<select>`
- kromě základních atribut, atributa `size="n"` (velikost boxu v řádcích), `multiple` (vybrání více prvků)
- v rámci prvku `select` prvek `<option>` s atributami
 - `value="optiondata"` (data zaslaná serveru při výběru)
 - `selected` (specifikace defaultní volby)
- prvky výběru můžeme seskupit pomocí prvku `<optgroup label="submenu title">`

```
<label for="referral">Where did you find out about us?</label>
<select id="referral" name="referral">
  <optgroup label="Online">
    <option value="social_network">Social Network</option>
    <option value="search_engine">Search Engine</option>
  </optgroup>
  <optgroup label="Offline">
    <option value="postcard">Postcard </option>
    <option value="word_of_mouth">Word of Mouth</option>
  </optgroup>
</select>
```

Prvky pro nahrání souborů

- na začátku formuláře specifikujeme:

```
<form      method="post"      enctype="multipart/form-data"
action="upload.url">
```

, kde atributa `enctype` zaručuje upload souborů ve správném formátu a atributa `action` je url skriptu, který nahrané soubory zpracuje

- v rámci formuláře dále vytvoříme prvek

```
<input type="file" id="idlabel" name="dataname" size="n" multiple>
```

```
<form method="post" action="show-data.php" enctype="multipart/form-data" >
...
  <label for="picture">Picture:</label>
  <input type="file" id="picture" name="picture" />
  <p class="instructions">
    Maximum size of 700k. JPG, GIF or PNG.
  </p>
...
</form>
```

Skryté prvky

- často se používá ve zpracovatelských skriptech k uložení informací získaných z dřívějšího formuláře, aby je bylo možné zkombinovat s daty současného formuláře

```
<form method="post" action="your-script.php">  
  <input type="hidden" name="step" value="6" />  
  ... other form fields ...  
  <input type="submit" value="Submit Form" />  
</form>
```

Tlačítko pro odeslání

- každý formulář by ho měl obsahovat, jinak data nezískáme
- může to být text, obrázek a nebo kombinace
- `<input type="submit" value="submitmessage">`

```
<input type="submit" value="Create Profile" class="btn" />
```

```
.btn {  
  background-color: #da820a;  
  border: none;  
  border-radius: 4px;  
  box-shadow: 2px 2px 2px #333;  
  color: #fff;  
  margin: 12px 0 0 26px;  
  padding: 8px;  
  text-shadow: 1px 1px 0px #777;  
}
```

Tlačítko odeslání s obrázkem

```
<input type="image" src="button-submit.png" width="188" height="95" alt="Create Profile"/>
```

Tlačítko odeslání s obrázkem a textem

```
<button type="submit" class="btn">  
    
  Create Profile  
</button>
```

Znepřístupnění částí formuláře

- na základě předchozích voleb můžeme chtít znepřístupnit nějakou část formuláře
- atributa `disabled="disabled"`
- pro interaktivní znepřístupnění se používá JavaScript => INTA

Stylizace obsahu formuláře

- prvky formuláře můžeme stylizovat pomocí pseudotříd v CSS3:
 - `:focus` (zaměření kurzorem)
 - `:checked` (při zaškrtnutí)
 - `:disabled` (s atributou `disabled`)
 - `:enabled` (bez atributy `disabled`)
 - `:required` (s atributou `required`)
 - `:optional` (bez atributy `required`)
 - `:invalid` (špatně vyplněn)
 - `:valid` (správně vyplněn)

```
input:checked + label{  
  color: green;  
}
```

Cvičení:

Vytvořte html stránku CSS styl dle zadání.

Video, audio a další multimédia

- budeme diskutovat vlastnosti zavedené od HTML5, starší prohlížeče tak mohou mít komplikace s kompatibilitou
- ve starších verzích bylo třeba mít plugin pro přehrání multimédií
 - často náročné pro hardware uživatele a krosprohlížečová nekompatibilita
- HTML5 multimedia neumožňují defaultně zabezpečení a tak zpřístupněný multimediální obsah mohou uživatelé stáhnout
- různé prohlížeče vyžadují různé formáty médií

Video formáty

- tři základní typy podporované HTML5:
 - *Ogg Theora*
 - přípona .ogg nebo .ogv
 - Firefox, Chrome, Opera, Firefox for Android
 - MP4
 - přípona .mp4 nebo .m4v
 - Safari, Chrome, Internet Explorer, iOS (Mobile Safari), Android, Chrome for Android, Firefox for Android, Opera Mobile 11+
 - WebM
 - přípona .webm
 - Firefox, Chrome, Opera, Android, Chrome for Android, Firefox for Android, Opera Mobile 14+
- je dobré mít připravené video alespoň ve dvou různých formátech

Přidání videa na web

- prvek `video` :

```
<body>  
<video src="paddle-steamer.webm" width="369" height="208"></video>  
</body>  
</html>
```

Další atributy prvku video

- `autoplay` = automatické přehrávání
- `controls` = přidání defaultních ovladačů
- `muted` = ztlumení audia
- `loop` = přehrávání ve smyčce
- `poster` = obrázek, který je zobrazen než se video načte
- `preload` = hint pro prohlázeč, kolik dat videa má načíst
 - hodnoty `none`, `metadata`, `auto`

Přidání více formátů najednou

- v rámci `video` přidáme prvek `source`
- prohlížeč nejprve hledá v rámci `video` atributu `source`, pak prochází atributy `source` a zkouší, který formát lze přehrát

```
<video width="369" height="208" controls>  
  <source src="paddle-steamer.mp4" type="video/mp4">  
    <source src="paddle-steamer.webm" type="video/webm">  
<p><a href="paddle-steamer.mp4"> Download the video</a></p>  
</video>
```

- prohlížeče, které nepodporují ani jeden formát místo toho ukáží link pro stažení
- atributy `source` jsou `src`, `type` (hodnoty `video/mp4`, `video/webm`, or `video/ogg`) a `media` (CSS media)

Audio formáty v HTML5

- podporované formáty:
 - Ogg Vorbis (.ogg)
 - MP3 (.mp3)
 - WAV (.wav)
 - AAC (.aac)
 - MP4 (.mp4)
 - Opus (.opus)
- stejně jako v případě videa je dobré přidat audio ve vícero formátech pro kompatibilitu u napříč prohlížeči

Přidání audia na web

- prvek `audio`

```
<body>  
<audio src="piano.ogg" controls ></audio>  
</body>
```

- atributy:
 - src, autoplay, controls, muted, loop, preload
- stejně jako v případě videa více formátů přidáme prvkem `source`

Tabulky v HTML

- prvek `table`
- každý řádek (`tr`) obsahuje buňku s nadpisem (`th`), daty (`td`), nebo obojím
- lze přidat prvek `caption` pro vysvětlení tabulky (musí být první)
- atribut `scope` dodává informace čtečkám obrazu o jaký nadpis se jedná (řádku, sloupce,...)

```

<table>
  <caption>Quarterly Financials for 1962-1964 (in Thousands)</caption>
  <tr>
    <th scope="col">1962</th>
    <th scope="col">1963</th>
    <th scope="col">1964</th>
  </tr>
  <tr>
    <td>$145</td>
    <td>$167</td>
    <td>$161</td>
  </tr>
  <tr>
    <td>$140</td>
    <td>$159</td>
    <td>$164</td>
  </tr>
  <tr>
    <td>$153</td>
    <td>$162</td>
    <td>$168</td>
  </tr>
  <tr>
    <td>$157</td>
    <td>$160</td>
    <td>$171</td>
  </tr>
</table>

```

Quarterly Financials
for 1962-1964 (in
Thousands)

1962	1963	1964
\$145	\$167	\$161
\$140	\$159	\$164
\$153	\$162	\$168
\$157	\$160	\$171

Rozšířené strukturování

```
<table>
  <caption>Quarterly Financials for 1962-1964 (in Thousands)</caption>
  <thead> <!-- table head -->
    <tr>
      <th scope="col">Quarter</th>
      <th scope="col">1962</th>
      <th scope="col">1963</th>
      <th scope="col">1964</th>
    </tr>
  </thead>
  <tbody> <!-- table body -->
    <tr>
      <th scope="row">Q1</th>
      <td>$145</td>
      <td>$167</td>
      <td>$161</td>
    </tr>
    <tr>
      <th scope="row">Q2</th>
      <td>$140</td>
      <td>$159</td>
      <td>$164</td>
    </tr>
    ... [Q3 and Q4 rows] ...
  </tbody>
  <tfoot> <!-- table foot -->
    <tr>
      <th scope="row">TOTAL</th>
      <td>$595</td>
      <td>$648</td>
      <td>$664</td>
    </tr>
  </tfoot>
</table>
```

Quarterly Financials for 1962-
1964 (in Thousands)

Quarter	1962	1963	1964
Q1	\$145	\$167	\$161
Q2	\$140	\$159	\$164
TOTAL	\$595	\$648	\$664

- tabulka se základním CSS stylem prohlížeče nevypadá pěkně => upravit CSS styl manuálně, např:

```
table {  
    border-collapse: collapse;  
}  
caption {  
    font-size: .8125em;  
    font-weight: bold;  
    margin-bottom: .5em;  
}  
th, td {  
    font-size: .875em;  
    padding: .5em .75em;  
}  
td {  
    border: 1px solid #000;  
}  
tfoot {  
    font-style: italic;  
    font-weight: bold;  
}
```

Rozšíření sloupců a řádků

- atributa `colspan`, resp. `rowspan` s hodnotami přes kolik řádků, resp. sloupců chcete daný řádek, sloupec rozšířit

```
<table>
<caption>TV Schedule</caption>
<thead> <!-- table head -->
  <tr>
    <th scope="rowgroup">Time</th>
    <th scope="col">Mon</th>
    <th scope="col">Tue</th>
    <th scope="col">Wed</th>
  </tr>
</thead>
<tbody> <!-- table body -->
  <tr>
    <th scope="row">8 pm</th>
    <td>Staring Contest</td>
    <td colspan="2">Celebrity Hoedown</td>
  </tr>
  <tr>
    <th scope="row">9 pm</th>
    <td>Hardy, Har, Har</td>
    <td>What's for Lunch?</td>
    <td rowspan="2">Screamfest Movie of the Weak</td>
  </tr>
  <tr>
    <th scope="row">10 pm</th>
    <td>Healers, Wheelers & Dealers</td>
    <td>It's a Crime</td>
  </tr>
</tbody>
</table>
```

TV Schedule

Time	Mon	Tue	Wed
8 pm	Staring Contest	Celebrity Hoedown	
9 pm	Hardy, Har, Har	What's for Lunch?	Screamfest Movie of the Weak
10 pm	Healers, Wheelers & Dealers	It's a Crime	

Zdroje:

Castro, E. (2006). HTML, XHTML, and CSS: Visual QuickStart Guide. Pearson Education.