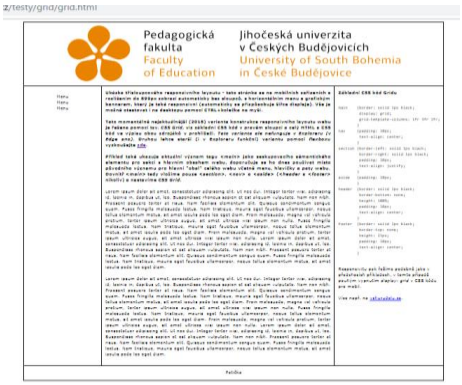
Responzivní web

Ukázka třísloupcového responzivního layoutu - tato stránka se na mobilních zařízeních s rozlišením do 899px zobrazí automaticky bez sloupců, s horizontálním menu a grafickým bannerem, který je také responzivní (automaticky se přizpůsobuje šířce displeje). Vše je možné otestovat i na desktopu pomocí CTRL+kolečka na myši.



Tato varianta konstrukce responzivního layoutu webu je řešena pomocí tzv. *CSS Grid*, viz základní CSS kód na konci textu a celý HTML a CSS kód ve výpisu obou zdrojáků v prohlížeči. *Tato varianta ale nefunguje v Exploreru (v Edge ano)*.

Příklad také ukazuje aktuální význam tagu <*main*> jako seskupovacího sémantického elementu pro sekci s hlavním obsahem webu, doporučuje se ho dnes používat místo původního významu pro hlavní "obal" celého webu včetně menu, hlavičky a paty webu. Dovnitř <*main*> tedy vložíme pouze <*section*>, <*nav*> a <*aside*> (<*header* a <*footer*>nikoli) nastavíme *CSS Grid*.

Responzivní webdesign stojí na třech základních pilířích

- Flexibilní struktura/layout
- Flexibilní obrázky
- Media Queries

Viewport

Výřez-Viewport je prostor pro vykreslení webové stránky, tedy šířka okna prohlížeče. Viewport se liší podle zařízení a bude menší na mobilním telefonu než na obrazovce počítače. Před tabletem a mobilním telefonem byly webové stránky určeny pouze pro počítačové obrazovky a pro webové stránky bylo běžné, že měly statický design a pevnou velikost.

Nastavení Viewportu

HTML5 zavedlo metodu, která umožňuje webovým designérům převzít kontrolu nad Viewportem prostřednictvím značky <meta>. Prvek <meta> poskytuje prohlížeči pokyny, jak řídit rozměry a měřítko stránky.

Část width = device-width nastavuje šířku stránky tak, aby následovala šířku obrazovky zařízení (která se bude lišit v závislosti na zařízení).

Velikost obsahu do Viewportu

Uživatelé používají rolování webových stránek raději vertikálně – ne vodorovně! Pokud je tedy uživatel nucen vodorovně posouvat nebo oddálit, aby viděl celou webovou stránku, jde o stránky se špatným uživatelským komfortem.

Dodatečná pravidla:
1. NEPOUŽÍVEJTE velké prvky s pevnou šířkou - Pokud je například obrázek zobrazen na šířku širší než Viewport, může dojít k vodorovnému posouvání. Nezapomeňte tento obsah přizpůsobit šířce Viewportu.
2. Nenechávejte obsah spoléhat na určitou šířku Viewportu, aby se dobře vykreslil - protože rozměry a šířka obrazovky v pixelech se mezi zařízeními značně liší
3. Nastavení velkých absolutních šířek CSS pro elementy stránky se nedoporučuje. Místo toho zvažte použití relativních hodnot šířky, např.: 100%.

Po vložení meta značky pro viewport se z desktopového rozlišení stane mobilní. Bez použití meta značky se web vykreslí do výchozího layoutového viewportu, který má většinou šířku 980 pixelů. Web bude vypadat „jako na počítači, jen zmenšený". S použitím meta značky pro viewport se šířka layoutového viewportu nastaví na velikost rozlišení v CSS pixelech. Teoreticky dělá `initial-scale=1` na všech zařízeních totéž co `width=device-width`, ale bez toho druhého chybuje Internet Explorer na mobilních Windows 8 stejným způsobem jako osmá a starší verze mobilního operačního systému od Applu.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Do atributu **content** je možné dávat různé vlastnosti a jejich hodnoty:

width

Nastaví šířku layoutového viewportu v pixelech. Nejčastěji využívaná hodnota device-width sjednotí šířku layoutového viewportu se šířkou ideálního viewportu. Takže uživatel nebude muset zoomovat a vaši responzivní stránku uvidí jedna ku jedné. Pokud použijete hodnotu, např. width=400, nastavíte šířku layoutového viewportu na 400 pixelů.

initial-scale

Nastaví výchozí zoom, ale také šířku layoutového viewportu. Ve výsledku dělá zápis initial-scale=1 totéž jako width=device-width. Pokud chcete maximální kompatibilitu, uvádějte oba dva.

user-scalable

Hodnota no zakazuje uživateli jakkoliv zoomovat. Prosím, nepoužívejte ji. Zoomování je na mobilních zařízení fakt potřeba. Ať už jde o zvětšení textu v horších světelných podmínkách, nebo jen touhu vidět detaily z nějakého obrázku, přibližování obsahu prostě potřebují všichni uživatelé. Safari na iOS 10 a novějších navíc zákaz zoomování úplně ignoruje.

minimum-scale/maximum-scale

Minimální a maximální možný zoom. maximum-scale=1 ruší možnost přiblížení stejně jako user-scalable=no. Opět - nepoužívejte to.

shrink-to-fit

Pokud nějaké prvky pozicujete částečně mimo viewport (například pomocí position: absolute), na zařízeních s iOS se vizuální viewport přepočítá tak, aby se zobrazil i onen pozicovaný element.

Může se ale stát, že si to takhle nepřejete. Třeba jen chcete, aby byl element částečně oříznutý a mimo viewport. Od iOS 9 můžete použít deklaraci shrink-to-fit=no, kterou to zařídíte.Váš meta tag pro viewport by pak měl vypadat takto:

```
<meta name="viewport"
  content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
```

viewport-fit

Tohle je nová vlastnost, která řeší způsob zobrazování na zařízeních s jinou než hranatou obrazovkou. Jako příklad vezměme chytré hodinky nebo iPhone X a novější.

Vlastnost může mít následující hodnoty (už znáte z background-size):

auto – výchozí stav, který vše nechává na prohlížeči. U iPhone X a novějších to například odpovídá hodnotě contain.

contain – zmenší viewport pro stránku tak, aby byla vidět celá. Jakou barvu vykreslí po stranách, záleží na prohlížeči. U nových iPhonů je to background-color z body.

cover – roztáhne viewport pro stránku tak, aby nikde „nevyčuhovaly" neobarvené části rozhraní prohlížeče. S tím rizikem, že kulaté rohy nebo výčnělky na displeji zařízení některé části stránky překryjí.

Flexibilní obrázky

Technika flexibilních obrázků má zajistit, aby se obrázky přizpůsobovaly podobně jako samotná struktura/layout stránky. Není tedy vhodné, aby byla např. šířka a výška obrázku fixně definována v tagu <img>. K zamezení „přetékání" obrázků při přizpůsobování prvků stránky displeji zařízení je možné použít pro obrázky např. styl max-width nastavený na 100%. Tento styl je pak přiřazen obalovému elementu, třeba divu, ve kterém je obrázek umístěn. Šířka obrázku je pak dána velikostí tohoto divu. Vlastnost max-width navíc znamená automatickou reakci výšky obrázku na změnu šířky, kdy je zachován jejich původní poměr.

Zobrazení pomocí grid- mřížky

Mnoho webových stránek je založeno na zobrazení mřížky, což znamená, že stránka je rozdělena do sloupců. Používání mřížkového pohledu je velmi užitečné při navrhování webových stránek. Usnadňuje umístění prvků na stránku.

Media Query
Media Query je technika CSS zavedená v CSS3. Pravidlo @media slouží k nastavení použití bloku vlastností CSS pouze v případě, že je splněna určitá podmínka.
Příklad:
Pokud je okno prohlížeče 600px nebo menší, barva pozadí bude modrá:

```
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

Mobile First
Vždy nejprve vytváříme design pro mobilní zařízení před návrhem pro stolní počítač nebo jiné zařízení (stránka zobrazí rychleji na menších zařízeních). Tzn., že v CSS namísto změny stylů, když šířka bude menší než 768px, bychom měli mít design pro menší šířku a měnit ho, když šířka bude větší než 768px.

Typické šířky zařízení
Existuje mnoho zařízení s různou výškou a šířkou, takže je těžké vytvořit pro každý přístroj přesnou šířku stránky. Pro zjednodušení určíme pět skupin (důležité je začít od nejmenšího k největšímu-viz. Mobile First):

```
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {...}

/* Small devices (portrait tablets and large phones, 600px and up) */
@media only screen and (min-width: 600px) {...}

/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {...}

/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {...}

/* Extra large devices (large laptops and desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {...}
```

Dalším běžným použitím Media Query je skrýt prvky podle různých velikostí obrazovky:

```
/* If the screen size is 600px wide or less, hide the element */
@media only screen and (max-width: 600px) {
  div.example {
    display: none;
  }
}
```

Změnit velikost písma s Media Query na různých velikostech obrazovky:

```
/* If the screen size is 601px or more, set the font-size of <div> to
80px */
```

```
@media only screen and (min-width: 601px) {
  div.example {
    font-size: 80px;
  }
}
/* If the screen size is 600px or less, set the font-size of <div> to
30px */
@media only screen and (max-width: 600px) {
  div.example {
    font-size: 30px;
  }
}
```

Vytvoření layoutu rozděleného na pět sloupečků mřížky. Kde první a poslední části pro postranní obsah zabírají každá jednu pětinu, na prostřední část (content) zbývají tři pětiny.

```
@media screen and (min-width: 600px) {
  .container {
      grid-template-columns: 1fr 3fr 1fr;
  }
}
```

Základní CSS kód Gridu

```
main    {border: solid 1px black;
         display: grid;
         grid-template-columns: 1fr 3fr 1fr;
        }
nav     {padding: 10px;
         text-align: center;
        }
section {border-left: solid 1px black;
         border-right: solid 1px black;
         padding: 10px;
         text-align: justify;
        }
aside   {padding: 10px;
        }
header  {border: solid 1px black;
         border-bottom: none;
         height: 100%;
         padding: 10px;
         text-align: center;
        }
footer  {border: solid 1px black;
         border-top: none;
         height: 40px;
         padding: 10px;
         text-align: center;
        }
```

Responzivitu pak řešíme podobně jako v předchozích příkladech, v tomto případě pouhým vypnutím display: grid v CSS kódu pro mobil.

Zdroje:

https://gridbyexample.com/

http://www.petrpexa.cz/testy/grid/grid.html

https://www.vzhurudolu.cz/prirucka/css-grid