

Dotazy na médium

V této kapitole:

- Zkoumáme vlastnosti média
- Dotazy na médium v praxi
- Využíváme příležitost

Dotazy na médium tvoří třetí pilíř responzivního webdesignu. Jedná se o rozšíření specifikace HTML5, jež umožňuje, aby schopnosti obrazovky určovaly, jaké prohlížeč aplikuje kaskádové styly (jak je definuje modul jazyka CSS3¹). Prohlížeč tedy kupříkladu rozpozná mobilní zařízení otočené svisle s průhledem o šířce 320 pixelů od monitoru desktopového počítače s průhledem o šířce 1 024 a může uplatnit jiné kaskádové styly. Standardně vývojáři přizpůsobují rozvržení, pozadí a obrázky, ale samozřejmě je možné použít zcela novou sadu stylů.

Existují tři způsoby, jak aplikovat kaskádové styly v závislosti na dotazech na médium. Zprv – jako šablony stylů v elementu `link` jazyka HTML nebo XHTML:

```
<link rel="stylesheet" type="text/css" media="all and (color)"  
href="/styly.css">
```

Zadruhé – v jazyce XML:

```
<?xml-stylesheet media="all and (color)" rel="stylesheet"  
href="/styly.css">
```

A nakonec – v šablonách stylů prostřednictvím pravidel `@import`:

```
@import url("/styly.css") all and (color);
```

nebo pomocí pravidel `@media`:

```
@media all and (color) { /* deklarace vlastností */ }
```

Responzivní webdesign poskytuje nástroje pro tvorbu rozvržení, na které se můžeme spoléhat. Díky tomu se můžeme soustředit více na obsah, kvůli němuž se k nám uživatelé budou rádi vracet.

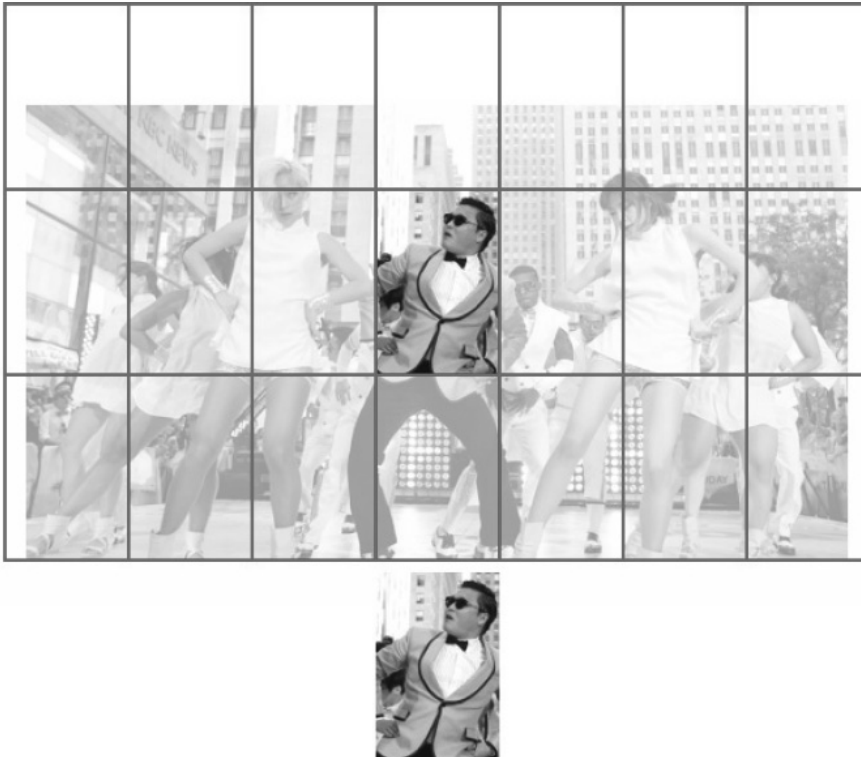
¹ <http://www.w3.org/TR/css3-mediaqueries/>

Jak ukazuje obrázek 4.1, moderní zařízení mají rozdílné schopnosti a požadavky, co se týká prostředků a stylování. Naštěstí je syntaxe dotazů na médium jednoduchá, a navíc je široce rozšířená i používána – ve skutečnosti jsme se s ní dosud setkali ve všech kapitolách této knihy.



Obrázek 4.1. Obrazové prostředky na cílových zařízeních

Běžně se snažíme omezovat kaskádové styly pro jednotlivá cílová zařízení. Začínáme od obecných stylů a pokračujeme styly pro konkrétní zařízení. Nezaměřujeme se přitom ale jen na styly, důležité jsou také prostředky zobrazované těmito styly. Na obrázku 4.2 lze vidět, že kdybychom zobrazili pozadí o šířce 1 920 pixelů na obrazovce široké 320 pixelů z obrázku 4.1, odesílali bychom 14krát větší obrázek, než je nutné. To by mělo negativní dopad na datový přenos a výkon, a to zcela zbytečně.



Obrázek 4.2. Náš malý obrázek se vleze do velkého obrázku přibližně 14krát

Ukažme si dotaz na médium určený pro telefon iPhone 5:

```
@media only screen and (min-device-width: 640px) and
(max-device-width: 1136px) and
(-webkit-min-device-pixel-ratio: 2) {
/* Cíleno na telefon iPhone5. */
}
```

Tato syntaxe je jistě povědomá. Setkali jsme se již s vlastností `device-width` u elementu `meta` typu `viewport` a s vlastností `device-pixel-ratio` u elementu `picture`. Kromě toho jsou dotazy na médium příbuzné s typy média, které spatřily světlo světa v roce 1998 jako součást specifikace jazyka CSS 2.0.²

² <http://www.w3.org/TR/2008/REC-CSS2-20080411/>

Zkoumáme vlastnosti média

Dotazy na médium jsou mocným nástrojem, i když se spoléháte jen na vlastnost `device-width`. Nezapomínejte však, že se nemusíte omezovat jen na šířku zařízení. Specifikace³ počítá s 13 vlastnostmi média – `width`, `height`, `device-width`, `device-height`, `orientation`, `aspect-ratio`, `device-aspect-ratio`, `color`, `color-index`, `monochrome`, `resolution`, `scan` a `grid`. Téměř všechny vlastnosti (s výjimkou vlastností `scan` a `grid`) můžou mít předpony `min-` a `max-`. Míra kontroly je ohromná a budí úžas.

Jestliže jste už teď nadšení z množství možností, nezapomínejte, že s vlastnostmi dotazů na médium pracujete v jazyce CSS, a proto můžou výrobci prohlížečů přidávat také proprietární předpony – například jako u vlastnosti `-webkit-device-min-pixel-ratio`, kterou společnost Apple umožňuje detekovat zařízení s Retina displeji. Navíc existuje spousta jiných předpon než jen `-webkit-`:

```
@media only screen and (-webkit-min-device-pixel-ratio: 2) and
  (min-width: 320px),
  only screen and (min--moz-device-pixel-ratio: 2) and
  (min-width: 320px),
  only screen and (-o-min-device-pixel-ratio: 2/1) and
  (min-width: 320px),
  only screen and (min-device-pixel-ratio: 2) and
  (min-width: 320px) {
  /* Styly pro Retina/AMOLED displeje. */
}
```

Poslední vlastnost nemá žádnou předponu. Všichni vývojáři doufají, že jednoho dne to bude stačit pro všechny webové prohlížeče.

Pryč jsou dny, kdy nás mohla příruční zařízení dostat do potíží. Dnes musíme myslet na všechny možnosti a rozmyslet si rádně design a způsob vývoje. Právě správný úsudek je nejtěžší, abychom nepodlehli svodům vytvářet stále specifitější dotazy na médium. Telefon iPhone 5 s operačním systémem iOS6 získal navíc oproti svému předchůdci 176 pixelů na výšku. Je však nepravděpodobné, že bychom potřebovali cílit na telefony iPhone pomocí podmínky `max-device-height: 960px`. Kdybychom to ale udělali, přehlédli bychom všechny telefony iPhone 5 (viz obrázek 4.3).

³ <http://www.w3.org/TR/css3-mediaqueries/#contents>



Obrázek 4.3. Srovnání výšky obrazovky telefonu iPhone 4 a iPhone 5

Pravdou je, že můžete odlišit telefon iPhone 5 od jeho předchůdců pomocí vlastnosti `device-height`, ale musíte rovněž počítat s tím, že například zápatí, které jste tak pracně rozvrhli pro telefon iPhone 4S, bude umístěné 176 pixelů nad spodkem obrazovky. Když iPhone 5 získal 176 pixelů, jak na tom budou další generace? Nezapomínejte, že byste měli psát kaskádové styly tak, aby to nemělo negativní dopad na vzhled vaší aplikace na žádném zařízení – ať už současném nebo budoucím – abyste nemuseli přepisovat kód pro všechny nové rozměry.

Proto bychom si měli ve stručnosti popsat syntaxi pro kontrolu vlastností. Seznam dotazů na médium je textový řetězec s dotazy na médium oddělenými čárkami:

```
<link rel="stylesheet" media="screen and (color), projection and (color)"
      href="priklad.css">
```

V případě, že je některý dotaz na médium v seznamu pravdivý, je pravdivý také celý seznam. Kdyby byl kupříkladu pravdivý dotaz na médium `screen and (color)` z výše uvedeného seznamu `screen and (color), projection and (color)`, tento seznam by byl pravdivý.

Oddělení dotazů na médium čárkami reprezentuje logickou operaci OR, přičemž v jednotlivých dotazech můžeme používat klíčové slovo `and`, abychom vyjádřili logickou operaci AND. Při logické operaci OR postačí, když danou podmínku splní libovolná vlastnost, kdežto s logickou operací AND ji musí splnit všechny vlastnosti. Předchozím seznamem dotazů na médium tudíž ověřujeme, zda je zařízení schopné zobrazovat naše stránky barevně na obrazovce nebo projektoru. Pokud toto zařízení splní naše podmínky, prohlížeč načte danou šablonu stylů.

S klíčovým slovem `not` můžeme vyloučit vlastnost, takže v níže uvedeném dotazu na médium cílíme na telefony iPhone, ale bez Retina displeje. Vyhledáme tak například telefon iPhone 3, ale už ne telefon iPhone 5:

```
@media only screen (min-device-width: 640px) and
  not (-webkit-min-device-pixel-ratio: 2) {
  /* styly pro telefony bez vyšší hustoty pixelů */
}
```

Koneckonců syntaxe dotazů na médium je přehledná a lze ji snadno číst a pochopit. Pouze byste si měli dávat pozor na implicitní logické operace OR. Kdyby se skupiny W3C a WHATWG rozhodly pro element `picture` místo atributu `src-set`, syntaxi dotazů na médium byste zužitkovali ještě více.

Podpora vlastností média

U dotazů na médium se můžeme soustředit převážně na rozměrové vlastnosti, jimiž jsou vlastnosti `width`, `height`, `device-width` a `device-height`. Ty jsou jednak srozumitelné, ale také mají dobrou podporu napříč prohlížeči. Musíme si však pamatovat, že vlastnost `width` odpovídá šířce průhledu prohlížeče, zatímco vlastnost `device-width` odpovídá šířce obrazovky zařízení, jak lze vidět na obrázku 4.4.



Obrázek 4.4. Srovnání elementů `width` a `device-width`

Projdeme si v rychlosti zbylé vlastnosti média pro lepší představu o jejich užitečnosti:

- `orientation` – určuje, jestli je zařízení otočené v režimu portrétu (svisle) nebo krajiny (vodorovně),
- `aspect-ratio` – poměr stran obrazovky (například 16:9 nebo 4:3),

- `resolution` – bodů na palec (DPI) nebo bodů na centimetr (DPCM),
- `scan` – typ obrazovky – cílí na televizory s progresivním skenováním,
- `grid` – hledá obrazovky Teletype nebo zařízení s pevným písmem,
- `monochrome` – ověřuje počet bitů na pixel v monochromatické vyrovnávací paměti, přičemž hodnota 0 značí, že daná obrazovka není monochromatická,
- `color` – ověřuje počet bitů na barvu, přičemž hodnota 0 značí, že daná obrazovka není barevná,
- `color-index` – ověřuje počet položek v indexové tabulce barev daného zařízení, přičemž hodnota 0 značí, že daná obrazovka není barevná.

Pokud chcete poznat vlastnosti média důkladně, měli byste znát poměr stran, rozlišení, otočení a barevnost obrazovek. S těmito termíny jste se jistě setkali, když jste vytvářeli grafiku pro web. Jsou rovněž v souladu s terminologií specifikace HTML5.⁴ Křivka učení je tedy velmi strmá.

Třebaže podpora těchto vlastností v prohlížečích není jednotná, i nerozměrové vlastnosti by vám měly být povědomé, jelikož použitá terminologie je poměrně standardní pro webové odvětví. Vlastnosti `scan` a `grid` nemusí být jasné všem moderním vývojářům, protože jsou určené pro specifické typy zařízení, které však budou vývojáři pro tato zařízení jistě znát. Kdybyste například vyvíjeli aplikace pro televizory, jistě byste poznali vlastnost `scan` a princip progresivního skenování.⁵



Poznámka: Co se stalo s typy média?

Měli bychom si uvědomit, že ačkoliv se zabýváme responzivním webdesignem, neměli bychom zapomínat také na další vlivy na naše aplikace. Třebaže se učíme RWD, neměli bychom přehlížet přístupnost, lokalizaci nebo to, jak budou naše aplikace vypadat na nestandardních displejích.

Naše dotazy na médium budou obvykle obsahovat typ média `screen`, ale existují rovněž jiné možnosti.⁶

Protože používat dotazy na médium je nejlepší způsob, jak se s nimi seznámit, přesuneme se k praktickému cvičení.

Dotazy na médium v praxi

Naše stránka WDS už obsahuje několik dotazů na médium. Obrázek 4.5 ukazuje, jak vypadá záhlaví této stránky v prohlížečích širších než 800 pixelů a užších než 800 pixelů.

⁴ https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

⁵ http://en.wikipedia.org/wiki/Progressive_scan

⁶ <http://drafts.csswg.org/mediaqueries3/#background>



Obrázek 4.5. Obrázky o šířce 1 366 a 700 pixelů definované pomocí dotazů na médium

Vzhled stránky zůstává konzistentní, a co je důležitější – prohlížeč stahuje nejmenší možné množství dat. Co je cílem? Prohlédněte si obrázek 4.6.



Obrázek 4.6. Náš cíl: zápatí stránky WDS při šířce 1 440, 960, 800 a 480 pixelů

Stránka WDS aplikuje dotazy na médium prostřednictvím elementu `link` a syntaxe jazyka CSS. Prvním místem, kde prohlížeč hledá šablony stylů, je element `head`. Nyní tyto elementy deaktivujeme komentáři:

```

kapitola4/wds/index.html (úryvek)
<!-- <link rel="stylesheet" media="all and (max-width: 800px)"
      href="stylesheets/799max.min.css"> -->
<!-- <link rel="stylesheet" media="all and (min-width: 800px)"
      href="stylesheets/800plus.min.css"> -->
<!-- <link rel="stylesheet" media="all and (min-width: 800px)"
      href="stylesheets/panels.min.css"> -->

```

Nevýhodou tohoto postupu je, že prohlížeč stahuje i neodpovídající šablony stylů.⁷ Prohlížeč sice neaplikuje nevhovující styly ani nestahuje prostředky odkazované v neodpovídajících šablonách stylů, ale samotné šablony stylů stahuje vždy. To může negativně ovlivnit výkon aplikace a množství přenesených dat.

⁷ <http://scottjehl.github.io/CSS-Download-Tests/>