



Responsive **Web** Design

Zmieniamy Internet na plus

Adrian Gadzina

Kilka słów o mnie

- Autor bloga – 7pl.pl „Web is Brutal”
- Miłośnik Androida
- Pasjonat kolarstwa (MTB + szosa)
- Programista aplikacji mobilnych i stron internetowych w desi9n.pl



desi9n.pl

Adrian Gadzina



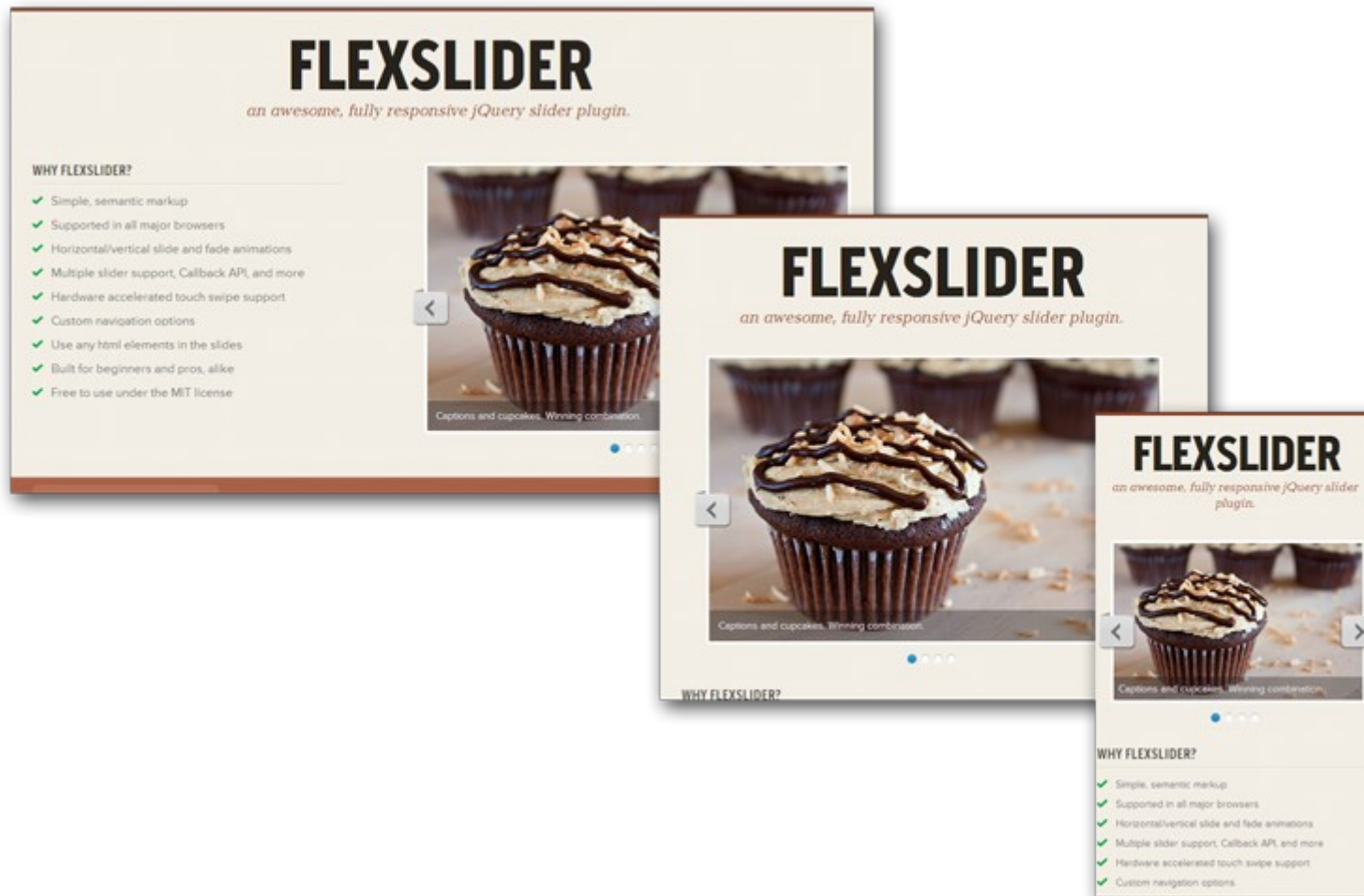
Czym jest Responsive **Web** Design?



Czym jest Responsive **Web** Design?

- Strony dostosowane do każdego urządzenia
- Brak osobnych wersji witryny pod urządzenia mobilne
- Wyjątkowa elastyczność - nie tylko sztywne 960px
- Wygoda użytkowania na pierwszym miejscu

Czym jest Responsive **Web** Design?





Dla kogo ten temat jest istotny?

- Twórcy stron i aplikacji internetowych
- Twórcy aplikacji mobilnych – PhoneGap
- Osoby zajmujące się webusability
- Graficy - ktoś to musi wszystko zaprojektować
- Projektanci i menedżerowie zajmujący się projektami opartymi na WWW
- Wszyscy pozostali zainteresowani tematem :)



Jaki jest **mobilny** internet?

- Coraz więcej użytkowników w mobilnej Sieci
– najwyższy czas na start, naukę i eksperymenty
- Większa dostępność
- Mobilny internet to nie tylko wolny internet
- Pełen szereg urządzeń z unikalnymi parametrami

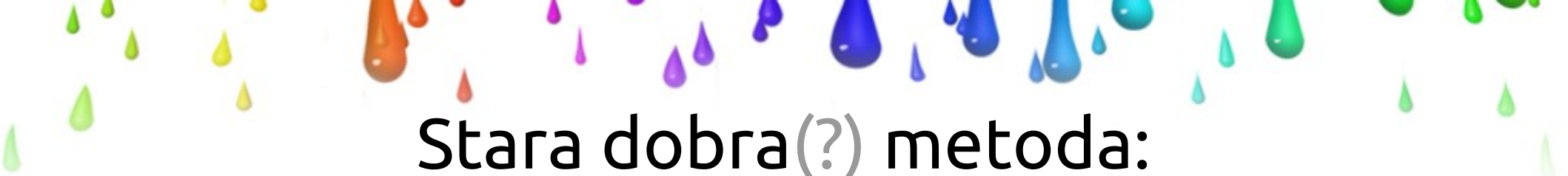


Strony „niemobilne”

- Flash, Silverlight
- Duże ilości niezoptymalizowanych zdjęć
- Rozbudowana zawartość (np.: portale)
- Duże ilości skryptów (często nawet powielane)
- Widżety społecznościowe
- Nawigacja niedostosowana do „dotyku”
- Itd...



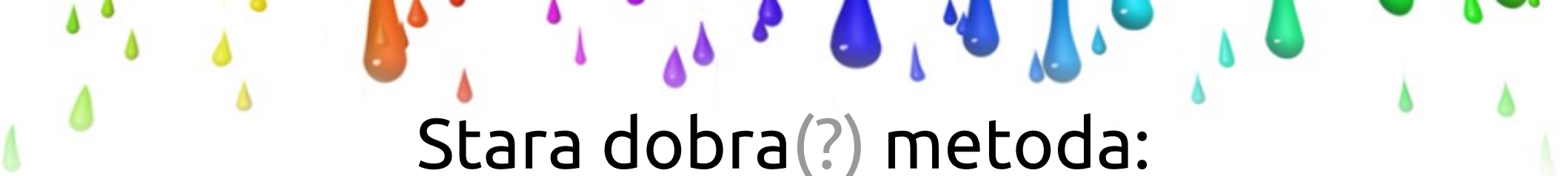
Dwie **osobne** wersje witryny?



Stara dobra(?) metoda: dwie wersje witryny

Plusy:

- Stare i ciężkie strony których nie chcemy na siłę udoskonalać
- Dopasowanie do konkretnych mobilnych wymagań
 - Dedykowany interfejs
 - Sposób zachowania się aplikacji webowej
 - Jedynie wymagana funkcjonalność
 - Zmniejszenie wagi strony
 - Odciążenie backendu



Stara dobra(?) metoda: dwie wersje witryny

Minusy:

- Rozmiar ekranu ma znaczenie – całkowity brak dostosowania
 - **Dawniej:** mały mobilny ekranik
 - **Dzisiaj:** pełna skala przeróżnych rozdzielczości ekranu
- Nadmierne uproszczenie witryny względem możliwości wybranych urządzeń – dostępne duże rozdzielczości, zwiększona wydajność, itd
- Koszty wdrożenia i rozwoju dwóch wersji (?)
- Nowe urządzenia..?
- Trzeba dostosować backend (?)



Responsive **Web** Design



Responsive **Web** Design

Co mamy do dyspozycji?

- CSS (w tym potężne Media Query z CSS3)
- JavaScript (JS to nie tylko jQuery)
- Inne...



Responsive **Web** Design

Projektowanie – zasada „**Mobile First**”

- Priorytet dla urządzeń mobilnych
- Lekkość użytych rozwiązań
- Kompatybilność z „klasycznymi” urządzeniami

Responsive **Web** Design:
CSS - **C**ascading **S**tyle **S**heets





CSS – na sztywno?

Nie tędy droga - musimy działać elastycznie...

CSS – MIN, MAX – 'rozmiary graniczne'

- min-width, max-width
- min-height, max-height

```
1 #element1 {  
2     min-width: 240px;  
3     max-width: 1200px;  
4 }  
5  
6 #element2 {  
7     min-height: 50px;  
8     max-height: 200px;  
9 }
```

CSS – procenty

```
1 #container{
2     margin: 0 auto;
3     width: 100%;
4     max-width: 900px;
5 }
6
7 article {
8     width: 80%;
9 }
10
11 nav {
12     width: 20%;
13 }
```



CSS – procenty

Nieco matematyki: na sztywno (wersja robocza)

```
1 #container{
2     margin: 0 auto;
3     width: 100%;
4     max-width: 966px;
5 }
6
7 article {
8     width: 750px;
9     margin-right: 20px;
10 }
11
12 nav {
13     width: 196px;
14 }
```

CSS – procenty

Nieco matematyki: elastycznie (przeliczamy na %)

```
1 #container{
2   margin: 0 auto;
3   width: 100%;
4   max-width: 966px;
5 }
6
7 article {
8   width: 77.6397516%;          /* 750px -> 750/966 = 0.776397516 */
9   margin-right: 2.07039337%; /* 20px -> 20/966 = 0.0207039337 */
10 }
11
12 nav {
13   width: 20.2898551%;         /* 196px -> 196/966 = 0.202898551 */
14 }
```

CSS – EMy

```
1 body {  
2   font-size: 1em;  
3 }  
4  
5 li {  
6   font-size: 1.5em;  
7 }
```

- Element 1
- Element 2
- Element 3
 - Element 3.1
 - Element 3.2
 - Element 3.3
 - Element 3.3.1
 - Element 3.3.2
 - Element 3.4
- Element 4

CSS – EMy

Narzędzie online: riddle.pl/emcalc/

 Em Calculator

ver 2.2

Toggle

```
body { 16 px = 1.000em } "1"
```

```
node2 { 32 px = 2.000em } "1" *
```

```
node3 { 20 px = 0.625em } "1" *
```

```
node2 { 8 px = 0.500em } "1" *
```

```
node3 { 20 px = 2.500em } "1" *
```

```
node2 { 20 px = 1.250em } "1" *
```

```
node3 { 20 px = 1.000em } "1" *
```

```
node4 { 16 px = 0.800em } "1" *
```

CSS – REMy (Root EM)

```
1 body {  
2   font-size: 1em;  
3 }  
4  
5 li {  
6   font-size: 1.5rem;  
7 }
```

- Element 1
- Element 2
- Element 3
 - Element 3.1
 - Element 3.2
 - Element 3.3
 - Element 3.3.1
 - Element 3.3.2
 - Element 3.4
- Element 4

CSS – Media Query

CSS2:

```
1 <link rel="stylesheet" type="text/css" href="style.css" media="screen">
2 <link rel="stylesheet" type="text/css" href="print.css" media="print">
```

Do dyspozycji: screen, print, tv, projection, handheld, tty, braille, aural, all

CSS3:

```
1 <link rel="stylesheet" type="text/css" href="1.css"
2 media="screen and (min-width: 480px)">
3
4 <link rel="stylesheet" type="text/css" href="2.css"
5 media="screen and (min-width: 800px)">
6
7 <link rel="stylesheet" type="text/css" href="3.css"
8 media="screen and (min-width: 1200px)">
```



CSS – Media Query

Osadzanie w arkuszu CSS:

```
1  @media screen and (min-width: 480px) {  
2      /* kod css */  
3  }  
4  
5  @media screen and (min-width: 800px) {  
6      /* kod css */  
7  }  
8  
9  @media screen and (min-width: 1200px) {  
10     /* kod css */  
11 }
```



CSS – Media Query

- Gdzie to zadziała?
 - IE9 (dla starszych mamy odpowiedni skrypt)
 - Opera 9.5+
 - Safari 3+
 - Firefox 3.5+
 - Chrome
 - Opera Mobile
 - iOS, Android
 - ...



CSS – Media Query

Co mamy do dyspozycji w MQ?

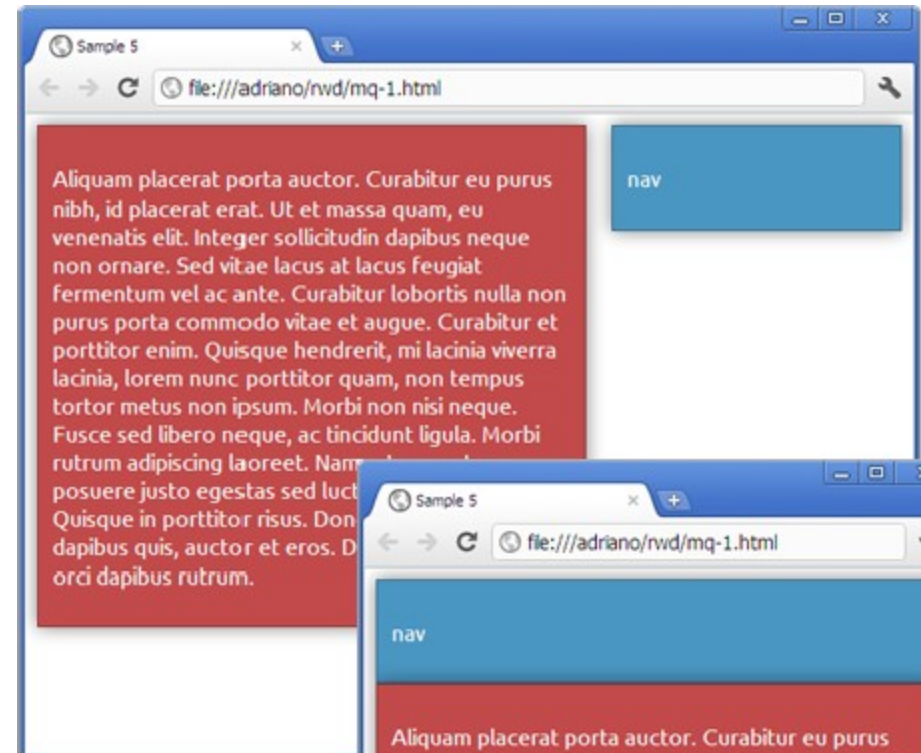
- kilka wybranych przykładów:

- min-width
- max-width
- min-height
- max-height

CSS – Media Query

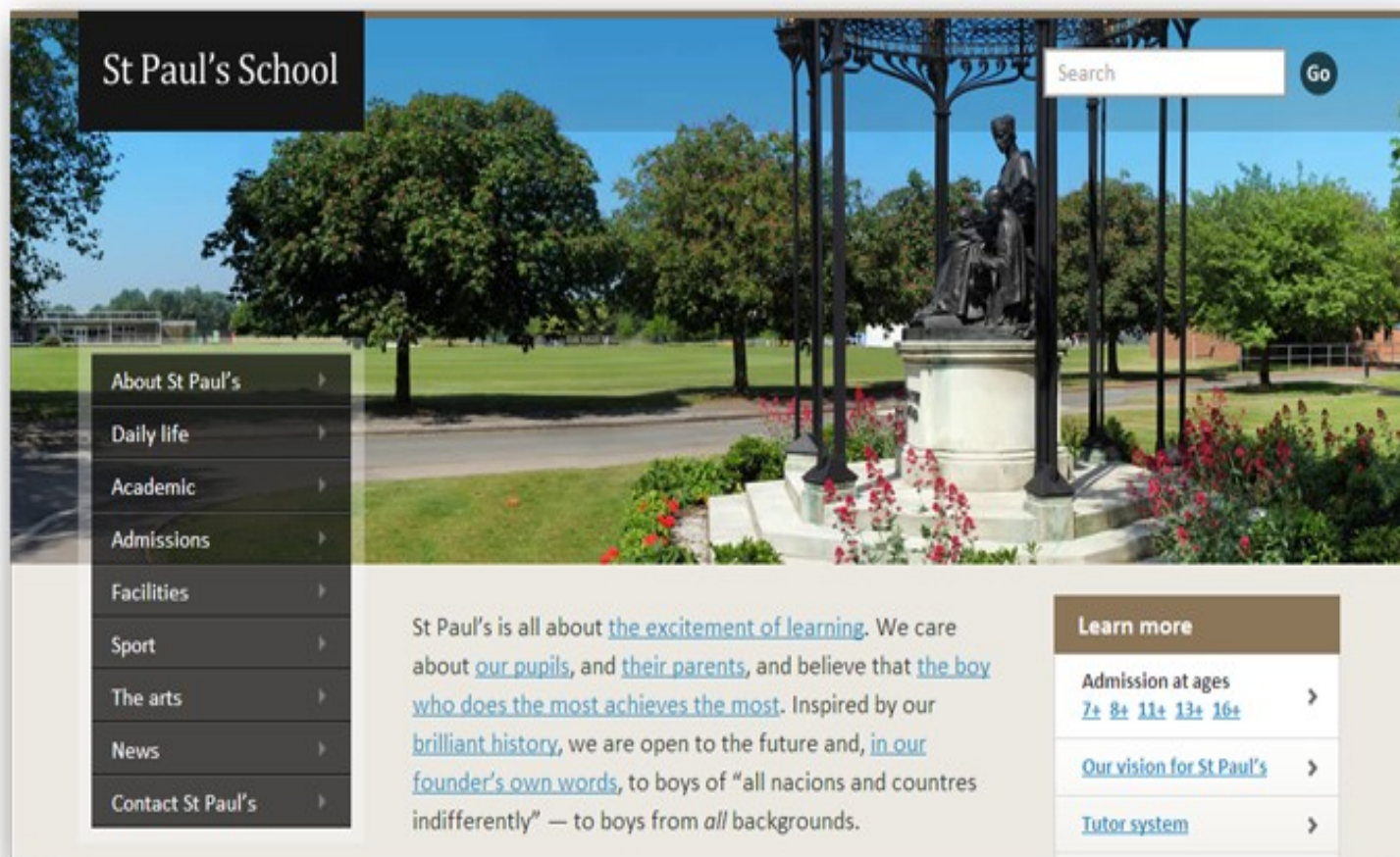
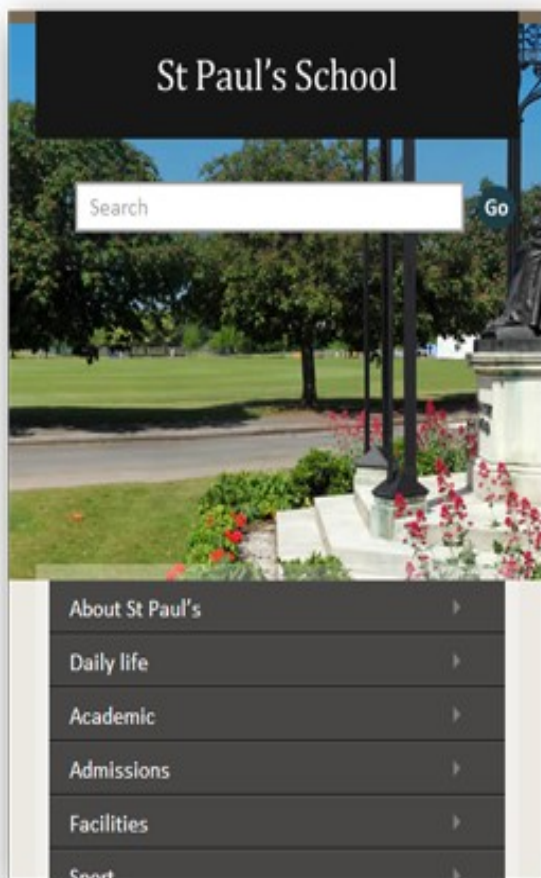
Przykładowy kod:

```
1  nav { /* style */ }
2  article { /* style */ }
3
4  @media screen and (min-width: 600px) {
5      nav {
6          float: right;
7          width: 30%;
8      }
9
10     article {
11         float: left;
12         width: 60%;
13     }
14 }
```



CSS – Media Query

Przykładowe wdrożenie:



CSS – Media Query

Orientacja ekranu:

```
1 @media screen and (orientation: landscape) {  
2     #menu {  
3         float: right;  
4     }  
5 }  
6  
7 @media screen and (orientation: portrait) {  
8     #menu {  
9         float: none;  
10    }  
11 }
```

CSS – Media Query

Łączenie reguł:

```
1 @media screen and (min-width: 320px) and (min-height: 320px) {
2     #element1 {
3         background-color: red;
4     }
5 }
6
7 @media screen and (min-width: 320px) and (max-width: 480px) {
8     #element2{
9         background-color: blue;
10    }
11 }
12
13 @media screen and (min-width: 200px), screen and (min-height: 200px) {
14     #element3 {
15         background-color: green;
16     }
17 }
```

CSS – Media Query

Not & Only:

```
1  @media not print and (color) {  
2      #element4 {  
3          background-color: black;  
4      }  
5  }  
6  
7  @media only screen and (min-width: 320px) {  
8      #element5 {  
9          background-color: pink;  
10     }  
11 }
```


CSS – Media Query

Wyraźniejsza grafika

```
1 @media
2 only screen and (-webkit-min-device-pixel-ratio: 2),
3 only screen and (min-device-pixel-ratio: 2) {
4
5     .icon {
6         background-image: url(img/icon-2x.png);
7         background-size: 20px 20px;
8     }
9
10 }
```





CSS – Media Query

Co jeszcze mamy do dyspozycji w MQ?

Obszar ekranu != obszar przeglądarki dostępny dla strony

- **device**-width
- **device**-height
- min-**device**-width
- max-**device**-width
- min-**device**-height
- Max-**device**-height




CSS – Media Query

MQ oferuje dużo więcej...

- min-resolution
- device-aspect-ratio
- min-aspect-ratio
- max-aspect-ratio
- max-device-aspect-ratio
- max-color
- max-color-index
- color, monochrome
- ...



Responsive **Web** Design:
JS – **J**ava **S**cript



Java Script – Media Query

Respond.js

- Obsługa min-width & max-width
- Przeznaczenie: IE7, IE8 i inne przeglądarki nie wspierające Media Query
- Nie musimy modyfikować kodu CSS
- Komentarze warunkowe dla IE: nie obciążamy pozostałych przeglądarek
- Wydajność zadowalająca, ale nie najlepsza w porównaniu do natywnego MQ



Responsive **Web** Design:
Obrazy



Obrazy i RWD

Problemy i oczekiwania:

- Obraz ma wyglądać dobrze zarówno na małym jak i dużym ekranie
- Nie można serwować dużego obrazu na mały ekran (internet nie zawsze jest szybki)
- Czasami detale na obrazie są istotne – zmniejszenie zdjęcia to utrata informacji



Obrazy i RWD

Adaptive Images – automatyczne dopasowanie:

- *adaptive-images.com* (Strona twórcy)
- Instalacja:
 - Wstawiamy odpowiedni `.htaccess` oraz skrypt `adaptive-images.php`
 - Tworzymy katalog „`ai-cache`”
 - Wstawiamy odpowiedni skrypt `JS` w sekcji `<head>`
- Dodatkowo: linki do obrazów w nominalnej rozdzielczości



RWD – kilka istotnych uwag



RWD to nie tylko plusy

- Konieczność ukrywania wybranych elementów
- Zmiana układu strony podczas zmiany orientacji ekranu czy zoomowania
- Dłuższy czas wdrożenia strony / aplikacji
 - termin powstania części desktopowej jest opóźniany tworzeniem części mobilnej
- Kolejna przeszkoda we współpracy grafików i koderów




RWD – na co uważać?

- Definiowanie różnych wariantów graficznego tła
- „Nietypowe” szerokości ekranu
- Fixed – lepiej unikać



RWD – testowanie

- Przeglądarka desktopowa - testujemy pełną skalę wysokości / szerokości
- Testy na rzeczywistych urządzeniach i różnych przeglądarkach



RWD – **warto?**
Ja jestem na **TAK**

A **Ty?**



Dziękuję za uwagę ;)