



# Responsive Web Design

Mateusz Marmołowski

[www.cta.ai](http://www.cta.ai)

[www.edu-sense.pl](http://www.edu-sense.pl)

[www.professor-why.pl](http://www.professor-why.pl)

# Fluid Design

- Strona wypełnia całą dostępną przestrzeń ekranu
- brak ustalonej na sztywno szerokości elementów (relative units)
- Treść dopasowuje się elastycznie
- opcjonalnie użycie maksymalnej szerokości (max-width)

# Fluid Design

Relative Units



Static Units



# Zadanie 1

Zmodyfikuj szablon `rwd_basic`, tak aby wypełniał całą stronę.

# Responsive image

- Obrazek wypełnia całą dostępną przestrzeń swojego kontenera
- Kontener jest elastyczny na wysokość
- Obrazek dostosowuje swoją szerokość do szerokości kontenera
- Obrazek zachowuje swoje proporcje
- Obrazek nie przekracza swoich naturalnych rozmiarów

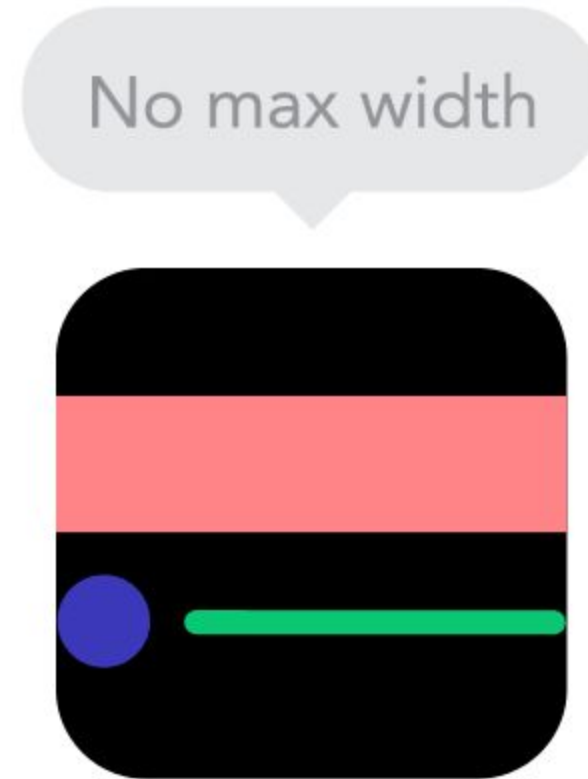
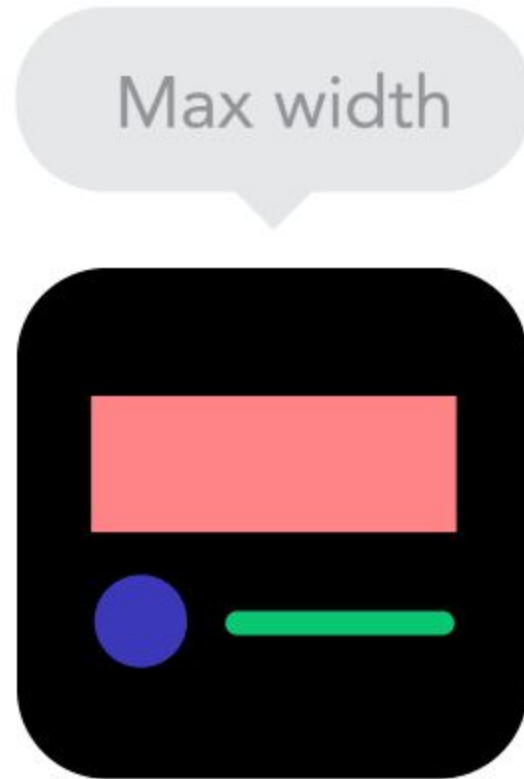
# Responsive image

```
img {  
    max-width: 100%;  
}
```

## Zadanie 2

Zmodyfikuj szablon `rwd_basic`, tak aby banner dostosowywał swój rozmiar względem dostępnej szerokości kontenera, ale nie przekraczał swoich naturalnych rozmiarów. Zadbaj o to aby był wycentrowany.

# Fluid Design - max-width





## Zadanie 3

Zmodyfikuj szablon `rwd_basic`, tak aby wypełniał całą stronę tylko jeśli szerokość przeglądarki jest  $< 1280\text{px}$ .

# Responsive video

- Film wypełnia całą dostępną przestrzeń swojego kontenera
- Film dostosowuje swoją szerokość do szerokości kontenera
- Film zachowuje swoje proporcje

# Flexible YouTube Video

```
.flexible-video-16-9 {  
    position: relative;  
    width: 100%;  
    padding-bottom: 56.25%; /* wynika z proporcji obrazu 16:9 */  
}  
  
.flexible-video-16-9 > iframe {  
    border: 0;  
    position: absolute;  
    left: 0; top: 0;  
    width: 100%; height: 100%;  
}
```

## Zadanie 4

Zmodyfikuj szablon `rwd_flexible-media`, tak aby film dostosowywał swój rozmiar względem dostępnej szerokości kontenera.

# Media queries

- Pozwalają na alternatywne stylowanie
- względem szerokości, wysokości, orientacji i proporcji ekranu
- względem typu urządzenia

@media <device-type> and (media-property: value) and ...

# Media queries - max-width

Zastosuj style jeśli szerokość ekranu nie przekracza max-width.

```
@media screen and (max-width: 800px) {  
    /* apply rules on mobile devices only */  
}
```

# Media queries - min-width

Zastosuj style jeśli szerokość ekranu przekracza min-width.

```
@media screen and (min-width: 800px) {  
    /* apply rules on tablets and desktops only */  
}
```

# Media queries - width

Zastosuj style jeśli szerokość ekranu wynosi dokładnie width.

```
@media screen and (width: 800px) {  
    /* apply rules on screens exactly 800px wide */  
}
```



# Media queries - width range

Zastosuj style jeśli szerokość ekranu mieści się pomiędzy min-width oraz max-width.

```
@media screen and (min-width: 800px) and (max-width: 1200px)
{
    /* apply rules on tablets only */
}
```

# Media queries - orientation

Zastosuj style jeśli ekran jest ułożony poziomo.

```
@media screen and (orientation: landscape) {  
    /* apply rules when screen is aligned horizontally */  
}
```

# Media queries - aspect-ratio

Zastosuj style jeśli ekran jest ułożony poziomo.

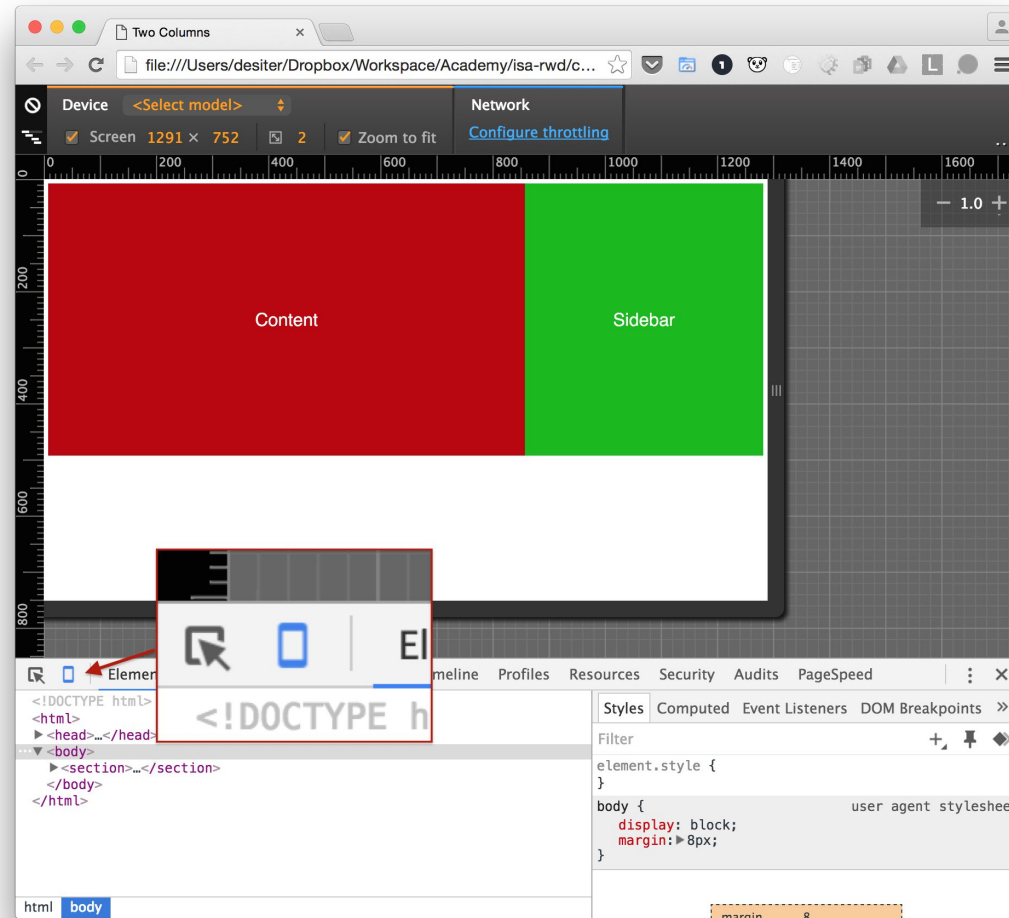
```
@media screen and (aspect-ratio: 16/9) {  
    /* apply rules for wide screen 16/9 only */  
}
```

# Media queries - device-width

Zastosuj style jeśli fizyczna szerokość ekranu urządzenia (nie tylko dostępna przestrzeń w przeglądarce) wynosi dokładnie device width.

```
@media screen and (device-width: 800px) {  
    /* apply rules on when devices screen is exactly  
       800px wide */  
}
```

# Chrome DevTools



## Zadanie 5

Zmodyfikuj szablon `rwd_backgrounds`, tak aby tło było:

- zielone dla smartfonów
- niebieskie dla tabletów
- czerwone dla desktopów

## Zadanie 6

Zmodyfikuj szablon `rwd_columns`, tak aby sidebar mieścił się pod contentem na urządzeniach mobilnych.

Załącz maksymalną szerokość ekranu dla mobile na poziomie 800px.

# Viewport meta

Znacznik opisujący zachowanie strony względem rozmiaru ekranu urządzenia mobilnego

```
<meta name="viewport" content="width=device-width">
```

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



## Zadanie 7

Otwórz szablon `rwd_columns`. w przeglądarce Chrome.

Uruchom Dev Tools w trybie mobilnym i sprawdź różnicę dla:

- braku znacznika
- dodaj znacznik `<meta name="viewport" ...>`  
z wartościami `width=device-width, initial-scale=1`
- zaobserwuj różnice dla różnych wartości `initial-scale=X`

# Mobile first

- W pierwszej kolejności style zgodne dla wszystkich urządzeń
- stosując regułę `min-width` wprowadzaj stopniowo rozszerzenia dla większych urządzeń

# Mobile first - dlaczego warto?

- paradoksalnie mniej CSSów
- optymalny rendering
- możliwość optymalizacji łącza

## Zadanie 8

Zmodyfikuj swoje wcześniejsze zmiany w szablonie `rwd_backgrounds`, tak aby style dla urządzeń mobilnych były zastosowane jako pierwsze.

## Zadanie 9

Zmodyfikuj swoje wcześniejsze zmiany w szablonie `rwd_columns`, tak aby style dla urządzeń mobilnych były zastosowane jako pierwsze.

## Zadanie 10

Zmodyfikuj szablon `rwd_thumbnails`, tak aby siatka miniaturek załamywała się:

- po 4 dla tabletów ( $800\text{px} < \text{width} < 1200\text{px}$ )
- po 3 dla smartphonów ( $\text{width} < 800\text{px}$ )

Dodaj obrazki (`images/thumbnail.jpg`) w trybie flexible image

Pamiętaj o podejściu mobile-first!

# Zadanie 11

Zmodyfikuj szablon `rwd_simplework`, tak aby był kompatybilny z urządzeniami mobilnymi. Spróbuj zastosować podejście mobile-first.

Uwzględnij między innymi:

- degradację kolumn
- flexible image
- zwijanie menu :-)

## Zadanie 12

Zmodyfikuj szablon `rwd_retention`, tak aby był kompatybilny z urządzeniami mobilnymi.

Uwzględnij między innymi:

- degradację kolumn zakładając, że:
  - portfolio załamujemy 2-etapowo: 2 kol. - tablet, 1 kol. - smartphone
  - footer dwuetapowo: 2 + 1 (tylko "Follow us") - tablet, 1 kol - smartphone
- flexible image



## Zadanie 13

Zmodyfikuj szablon `rwd_privy`, tak aby był kompatybilny z urządzeniami mobilnymi.

Uwzględnij między innymi:

- flexible design z maks szerokością 1200px
- ukrywanie menu dla smartphone (< 800px) z efektem push left :-)

# Bandwidth optimization

- `<link ...>` + media queries
- `@import` + media queries
- dobór rozmiarów obrazów tła przez zastosowanie media queries

Bardziej zaawansowane:

- wspomaganie JavaScriptem (Skeleton, Modernizr)
- cookies + optymalizacja obrazków server-side

# Media queries - link tag

```
<link href="styles.css" rel="stylesheet"  
      media="screen and (min-width: 800px)">
```

# Media queries - @import

```
@import url(desktop.css) screen and (min-width: 1200px);
```

# Dziękuję za uwagę

## Mateusz Marmołowski

### Materiały uzupełniające

- <http://learn.shayhowe.com/advanced-html-css/responsive-web-design/>
- <http://blog.froont.com/9-basic-principles-of-responsive-web-design/>
- <https://css-tricks.com/forums/topic/media-queries-import-style-sheets/>
- <http://www.zell-weekeat.com/how-to-write-mobile-first-css/>
- <http://www.html5rocks.com/en/mobile/responsivedesign/>
- <https://www.usertesting.com/blog/2015/02/03/responsive-web-design-resources/>