

1. RESPONSIVE WEB DESIGN

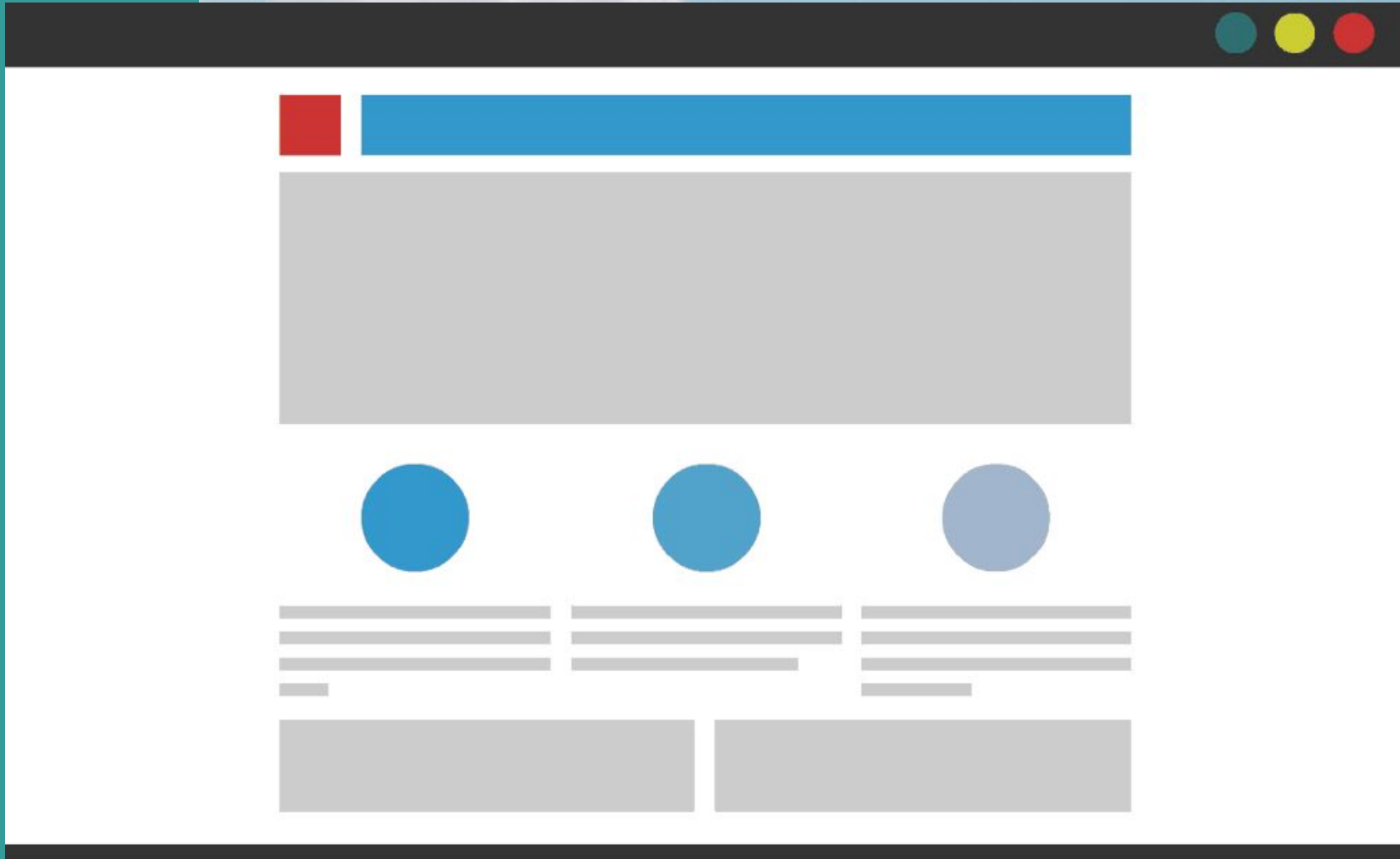
...czyli nasze strony na różnych urządzeniach i ekranach

Podstawowe pojęcia

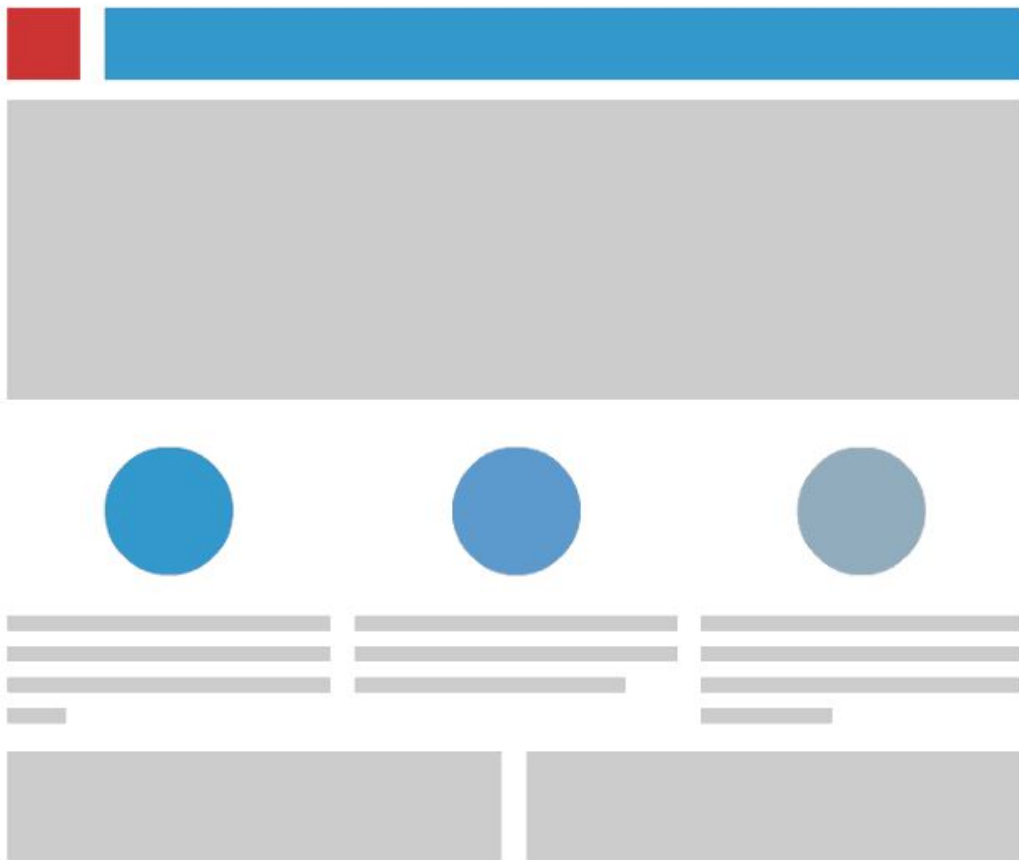
1. Jednostki statyczne - px, cm, point
2. Jednostki relatywne - %, vw, vh (wymiary), em, rem (font)
3. Media queries - specjalne warunki w CSS, dzięki którym wybrane fragmenty CSS są interpretowane tylko dla określonych wymiarów i rodzajów ekranów

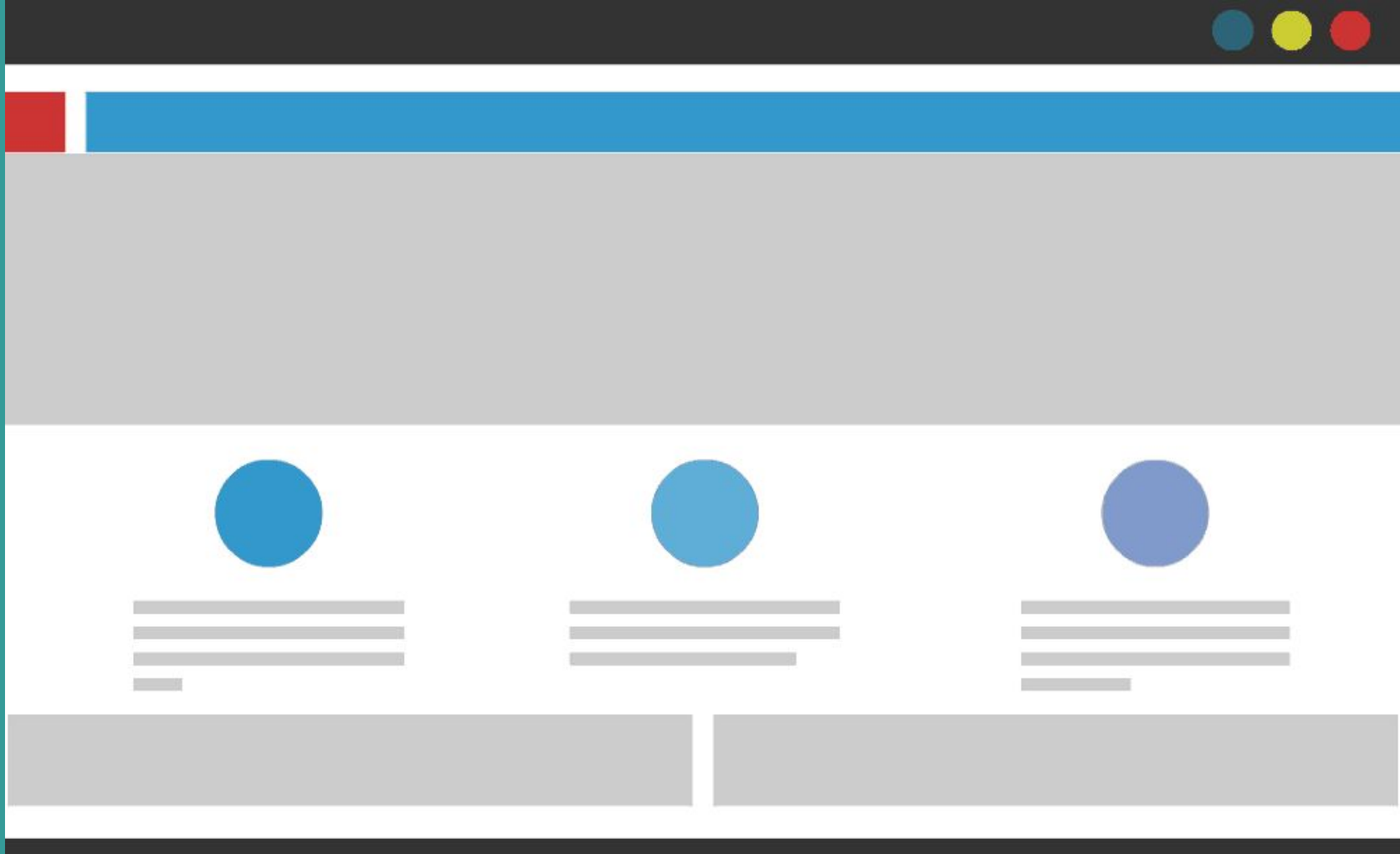
Różne podejścia do “design”

1. Fixed - stałe wymiary wyrażone w px
2. Fluid - płynnie dostosowujący się do obszaru layout (wymiary w %), brak media queries
3. Adaptive - kilka zakresów (wersji) dla różnych rozmiarów urządzeń przy użyciu media queries
4. Responsive - i płynny, i dostosowujący się layout (wymiary relatywne i media queries)









Źródło obrazków:

What is the difference between fixed, fluid, adaptive and responsive layouts and why should I care?

Zadanie 1

Zmodyfikuj szablon tak, aby wypełniał całą stronę (na szerokość), ale tylko jeśli szerokość strony jest mniejsza niż 960px

Obrazki w RWD

- Obrazek rozciąga się do pełnej szerokości swojego kontenera, ale jej nie przekracza
- Jednocześnie zachowuje proporcje (wysokość automatyczna)
- Obrazek nie przekracza swoich naturalnych rozmiarów (szerokości załączonego pliku)

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

Obrazki w RWD

Wyświetlanie różnych obrazków w zależności od rozmiaru ekranu:

- Przy użyciu właściwości display (block lub none)
- Przy użyciu srcset
 - Artykuł na MDN
 - DEMO

Zadanie 2

Zmodyfikuj szablon tak, aby baner:

- był wycentrowany
- dostosowywał rozmiar względem kontenera
- nie przekroczył swoich domyślnych rozmiarów

Osadzanie video w RWD

```
.video-container {  
  position: relative;  
  width: 100%;  
  padding-bottom: 56.25%;  
}
```

```
.video-container > iframe {  
  border: 0;  
  position: absolute;  
  left: 0; top: 0;  
  width: 100%; height: 100%;  
}
```

```
<iframe width="100%" height="auto" src="..." frameborder="0" ></iframe>
```

Niezbędne fragmenty kody

HTML:

```
<head>
  ...
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  ...
</head>
```

CSS:

```
* {
  box-sizing: border-box
}
```

Media queries - specyfikacja CSS2

```
<link rel="stylesheet" media="print" href="print.css">
```

Wybrane rodzaje “media”:

- screen
- print
- tv
- handheld
- all

Media queries - specyfikacja CSS3

CSS3 dorzuca kolejne warunki

- min-width, max-width
- width, device-width
- aspect-ratio
- orientation

Media queries - specyfikacja CSS3

CSS:

```
@media screen and (min-width: 800px) { ... }
```

CSS IMPORT:

```
@import url(full.css) screen and (min-width: 800px);
```

HTML:

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

Media queries

```
@media all and (min-width: 800px) {
```

```
}
```

```
@media (min-width: 800px) {
```

```
}
```

Te zapisy są równoważne

Media queries - width

```
@media all and (width: 800px) {  
    /* szerokość dokładnie 800px */  
}
```

```
@media all and (min-width: 800px) {  
    /* szerokość większa niż 800px */  
}
```

```
@media all and (min-width: 600px) and (max-width: 1200px) {  
    /* szerokość pomiędzy 600px a 1200px */  
}
```

Media queries - width

Jaki kolor będzie miał tekst w p dla width = 800px?

```
@media (max-width: 800px) {  
  p {  
    color: red;  
  }  
}
```

```
@media (min-width: 800px) {  
  p {  
    font-weight: bold;  
    color: green;  
  }  
}
```

Media queries - orientation

```
@media (orientation: portrait) {  
    / * ekran w pionie */  
}
```

```
@media (orientation: landscape) {  
    / * ekran w poziomie */  
}
```

Media queries - orientation

```
@media (aspect-ratio: 16/9) {  
    /* proporcje ekranu 16/9 */  
}
```

```
@media (device-width: 800px) {  
    /* fizyczna szerokość ekranu urządzenia (nie szerokość w przeglądarce) = 800px */  
}
```

MOBILE-FIRST



Zaczynamy projektować (i pisać kod) od najmniejszych ekranów

Mobile first

- Najpierw piszemy CSS ogólny - dla wszystkich ekranów
- Przy pomocy “min-width” idziemy stopniowo w górę (nadpisujemy style dla większych ekranów)
- Użytkownicy różnych ekranów powinni mieć zapewnioną pełną funkcjonalność strony
- Zamiast ukrywać dodatkowe elementy, dodajemy je (pokazujemy) dopiero kiedy mamy za dużo miejsca na stronie
- Szybsze ładowanie i renderowanie strony, mniejszy ruch sieciowy

“ *Podejście “Mobile first” może być wielkim wyzwaniem...*

...kiedy musimy przerobić już istniejący serwis “desktopowy”

Pozycjonowanie elementów

- Static
- Relative
- Absolute
- Fixed
- Float i clear

Zmiana kolejności elementów CSSem

- Float
- Flex order
- Właściwość position
- Direction

Artykuł:

<https://webdesign.tutsplus.com/tutorials/a-few-different-css-methods-for-column-ordering--cms-27079>

Zadanie - menu RWD

Zmodyfikuj szablon rwd_menu, tak aby desktop wyglądał tak (> 800px)

My awesome blog

item1	item2	item3	item4	
-------	-------	-------	-------	--

Post title

Lorem ipsum dolor sit amet consectetur adipisicing elit. Nemo atque doloremque voluptas suscipit error. Quia, quod nemo exercitationem tempora dicta id nobis nesciunt laborum assumenda! Aspernatur qui quasi est laboriosam vero reiciendis aperiam atque eligendi ea id nisi blanditiis doloremque voluptatum, alias odio in expedita mollitia nemo? Labore, mollitia. Facere!

Zadanie - menu RWD

Zmodyfikuj szablon rwd_menu, tak aby “tablet” (500-800px), miał menu przyklejone do góry strony, elementy menu są wyśrodkowane

item1	item2	item3	item4
-------	-------	-------	-------

My awesome blog

Post title

Lorem ipsum dolor sit amet consectetur adipisicing elit. Nemo atque doloremque voluptas suscipit error. Quia, quod nemo exercitationem tempora dicta id nobis nesciunt laborum assumenda! Aspernatur qui

Zadanie - menu RWD

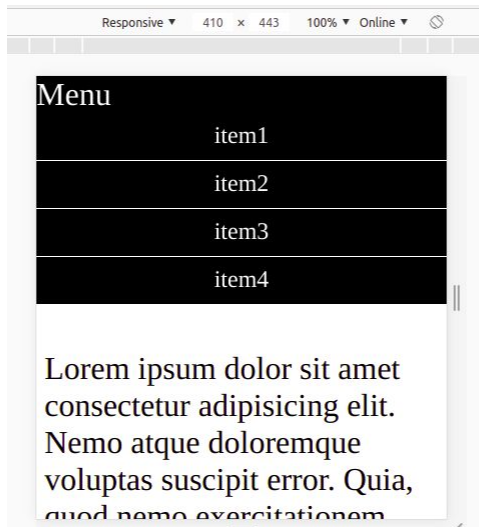
Zmodyfikuj szablon rwd_menu, tak aby mobile (<499px) miał menu domyślnie ukryte, przełączane na “Menu” (bez JS!):

Menu

My awesome blog

Post title

Lorem ipsum dolor sit amet
consectetur adipisicing elit.
Nemo atque doloremque
voluptas suscipit error. Quia,
quod nemo exercitationem



Do poczytania

- Pozycjonowanie elementów (static, relative, absolute)
- Centrowanie w CSS
- Jak działają floaty i clear