Aplikacje webowe dla urządzeń mobilnych

Płynne layouty

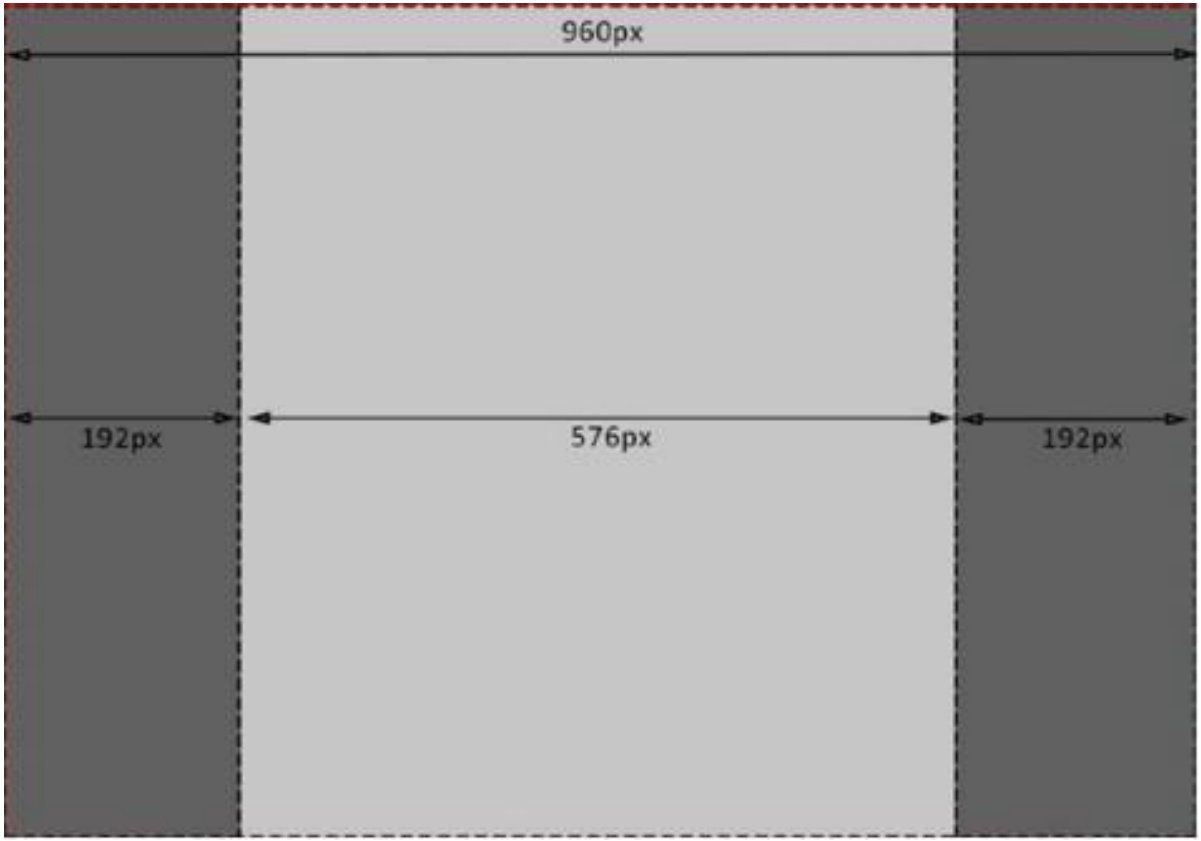
Michał Kuciapski
Wyższa Szkoła Bankowa

Tematyka

- Typy layoutów
- Zasady projektowania Fluid Layouts
- Płynne tła dla kolumn
- Flexbox

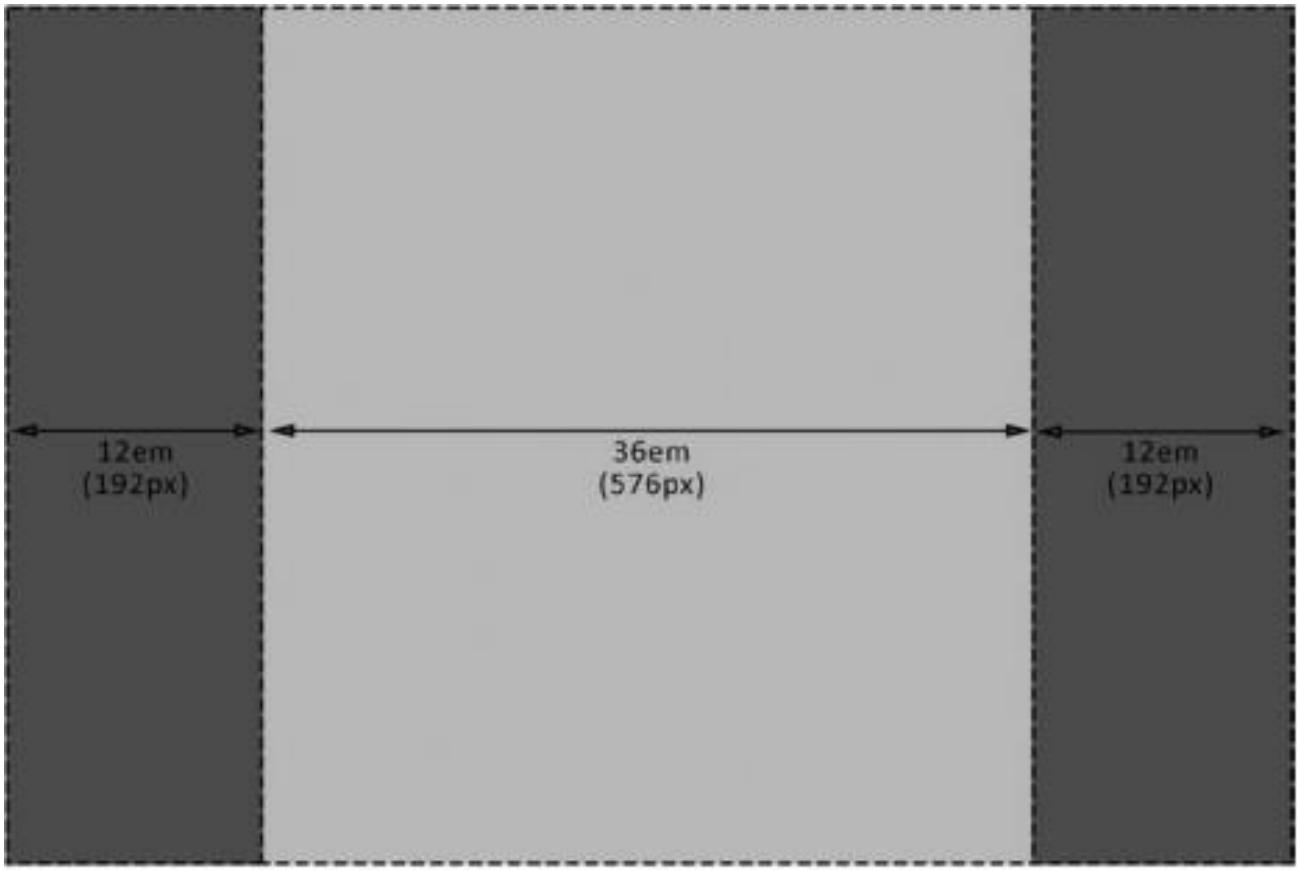
TYPY LAYOUTÓW

- **Fixed layout**



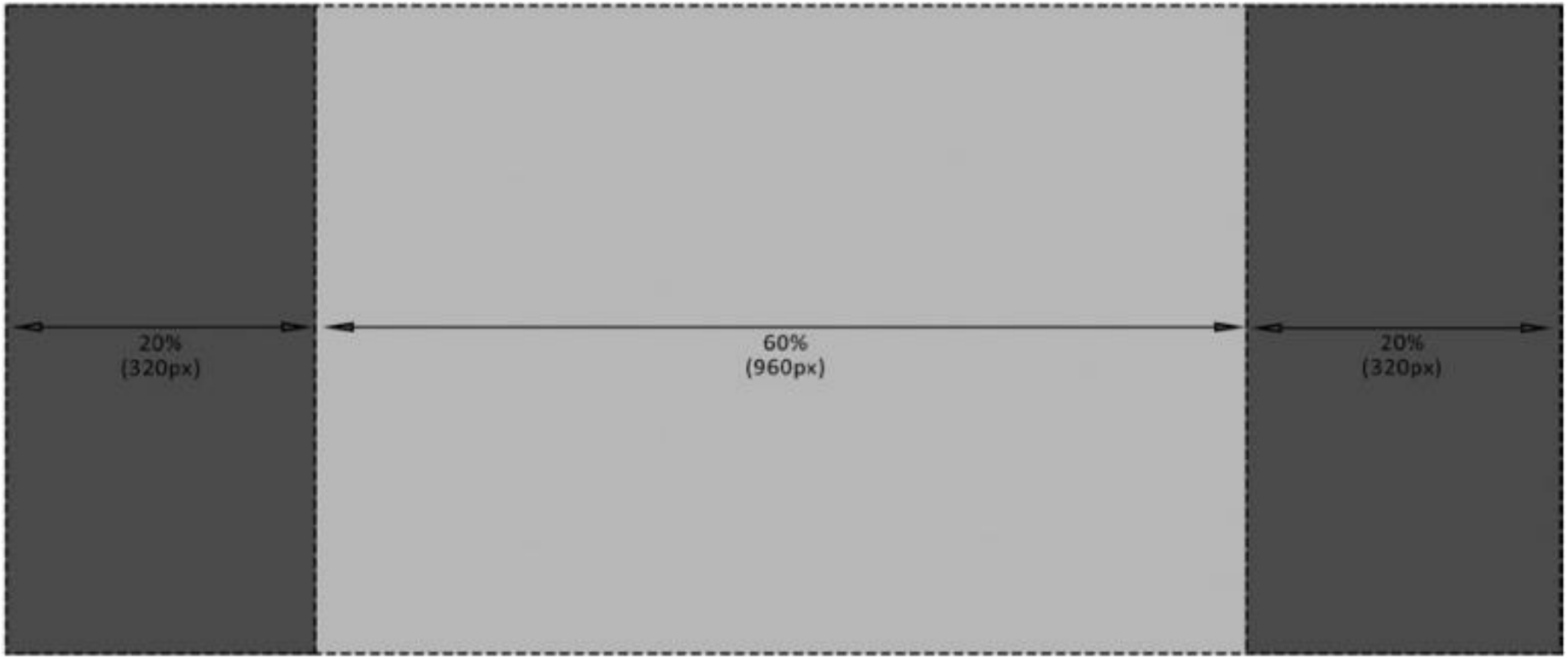
TYPY LAYOUTÓW

- **Elastic layout**



TYPY LAYOUTÓW

- **Fluid layout (liquid layouts or relative layouts)**



Zasady projektowania Fluid Layouts

- Nie stosowanie sztywnych rozmiarów.

CSS

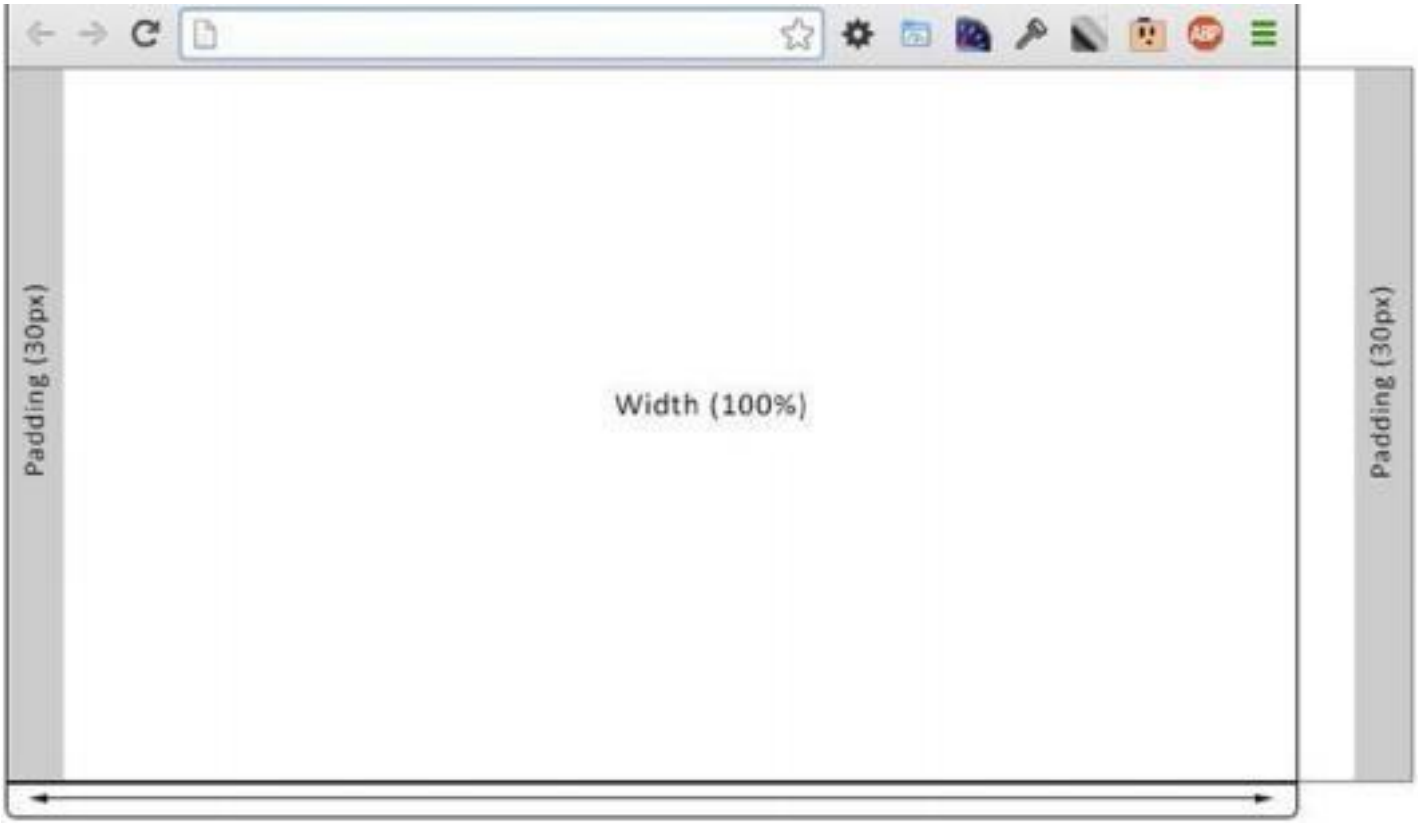
```
width: 50%;
height: 50%;
font-size: 1em;
```

Zdarzenia JS

```
window.addEventListener("resize", equalColumns, true);
```

Zasady projektowania Fluid Layouts

- Nie stosowanie poziomych suwaków.



Zasady projektowania Fluid Layouts

- Weryfikacja jak obrazki wyglądają w różnych rozmiarach.

```
img {
max-width:100%;
background-size: 100% 100%;
background: url(scalableimage.jpg) 0 0 no-repeat;
padding-bottom: 125%;
}
```

Zasady projektowania Fluid Layouts

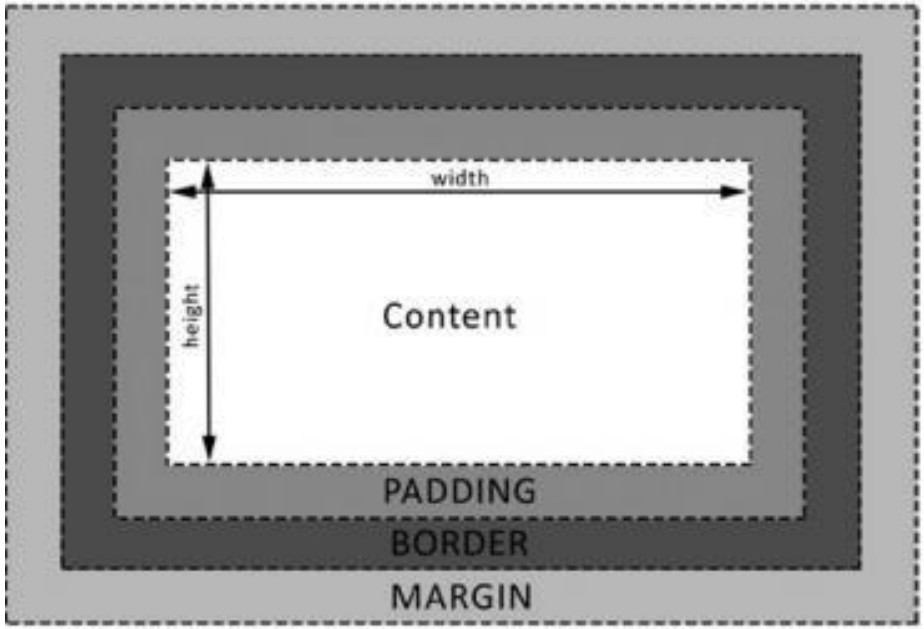
- Stosowanie zawijania zawartości.

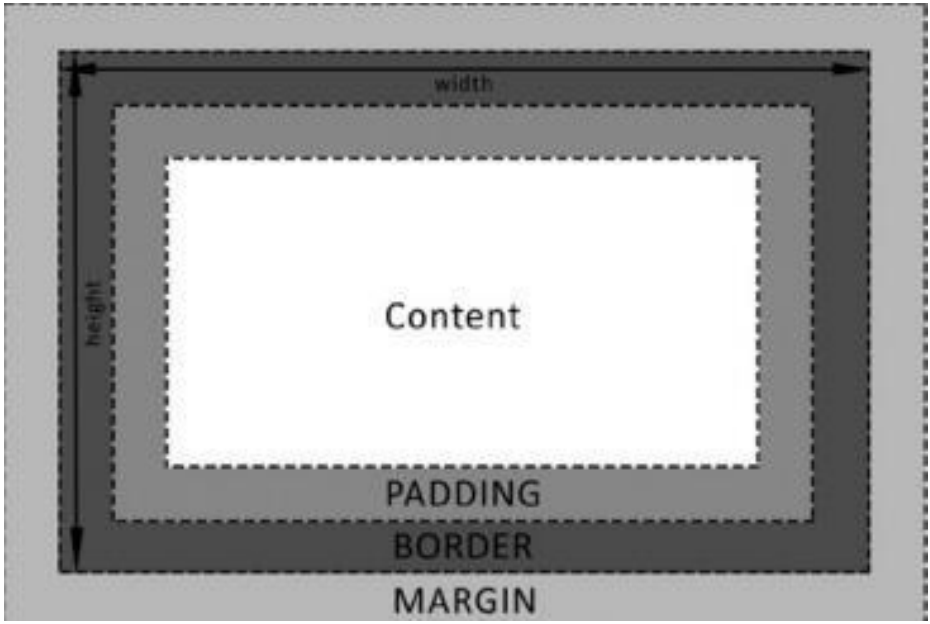
```
.col-container{
background: #000;
color: #fff;
}

.col{
float: left;
width: 50%;
}

.col-container:after{
content: ' ';
clear: both;
display: block;
}
```

Zasady projektowania Fluid Layouts

- Stosowanie zawijania zawartości.

```
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box; /*Opera/IE 8+*/
padding-…: …
```





Zasady projektowania Fluid Layouts

- Stosowanie różnego rozmiaru długości linii tekstu.

```
min-width: 200px;
min-height: 200px;
max-width: 800px;
max-height: 800px;
```

Zasady projektowania Fluid Layouts

- Powiązanie z Media Query

```
<article class="sm-col-span-4 lg-col-span-2">

.sm-col-span-4 {
    width: 100%;
}
@media screen and (min-width: 768px) {
    .lg-col-span-2 {
        width: 50%;
}
```

PŁYNNE TŁA DLA KOLUMN

- 2 kolumny – opakowanie kontenerem

```
<div class="col-container">
  <aside class="col">
          Sidebar
  </aside>
  <div class="col main">
          Main Content Area
  </div>
</div>
```

```
.col-container{
        background: #000;
        color: #fff;
}
.col-container:after{
        content: ' ';
        clear: both;
        display: block;
}
.col{
        float: left;
        width: 50%;
}
.col.main{
        background: #999;
}
```

Płynne tła dla kolumn

- 3 kolumny – gradient

```
<div class="col-container">
  <aside class="col">
            Sidebar
  </aside>
  <div class="col main">
            Main content
  </div>
  <div class="col">
            Related content
  </aside>
</div>
```

```
.col-container{
  background: linear-gradient(to right, #000000
    0%,#000000 33%,#a0a0a0 33%,#a0a0a0 66%,#a0a0a0
    66%,#707070 66%);
  color: #fff;
}
.col-container:after{
  content: ' ';
  clear: both;
  display: block;
}
.col{
  float: left;
  width: 33.3%;
}
```

Płynne tła dla kolumn

- 3 kolumny – JavaScript

```
<div class="col-container">
  <aside class="col nav">
            Sidebar
  </aside>
  <div class="col main">
            Main content
  </div>
  <div class="col related">
            Related content
  </aside>
</div>
```

```
.col-container{
  color: #fff;
}
.col-container:after{
  content: ' ';
  clear: both;
  display: block;
}
.col{
  float: left;
  width: 33.3%;
}
.col.nav{
  background: #aaa;
}
.col.main{
  background: #000;
}
.col.related{
  background: #999;
}
```

PŁYNNE TŁA DLA KOLUMN

- 3 kolumny – JavaScript

```
var equalColumns = function () {
    var columns = document.getElementsByClassName('col'), height = 0;
  //Reset the height of all the columns so that we can recalculate max height
    for (var i = 0; i < columns.length; i++) {
        columns[i].style.height = "auto";
    }
    //Loop through columns and find max height
    for (var i = 0; i < columns.length; i++) {
        if (height < columns[i].clientHeight) {
            height = columns[i].clientHeight;
        }
    }
    //Apply the max height to all columns
    for (var i = 0; i < columns.length; i++) {
        columns[i].style.height = height + "px";
    }
}
//Initially set the column heights
equalColumns();
//Update column heights on browser resize
window.addEventListener("resize", equalColumns, true);
```

FLEXBOX

- Flexible boxes, or flexbox, is a new layout mode in CSS3.
- Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.
- There are flexbox responsive grids like: http://flexboxgrid.com

```
    <style>
        body {
            direction: rtl;
        }
        .flex-container {
            display: -webkit-flex;
            display: flex;
            -webkit-flex-direction: row-reverse;
            flex-direction: row-reverse;
            width: 100%;
            background-color: lightgrey;
        }
        .flex-item {
            background-color: cornflowerblue;
            width: 33%;
            margin: 1%;
        }
    </style>
</head>
<body>
    <div class="flex-container">
        <div class="flex-item">MK</div>
        <div class="flex-item">Michał Kuciapski</div>
        <div class="flex-item">W pracy dydaktycznej dr M. Kuciapski  specjalizuję
się w zagadnieniach z dziedziny wytwarzania aplikacji, takich jak: programowanie,
inżyniera programowania, jakość kodu, technologie webowe i mobilne, systemy
rozproszone, bazy danych w systemach informatycznych oraz akceptacja oprogramowania.
        </div>
    </div>
```