RESPONSIVE DESIGN

Bálint Pál Tóth

History

- Until 2008 traditional desktop + mobile / WAP version of interfaces.
- Increasing problem: wide variety of resolutions from smart watches to 4K TVs.
- In 2010 Ethan Marcotte, independent designer introduced the fundamentals of responsive user interfaces.

What responsive UI is?

- Comes from architect: **responsive spaces** (eg. „smart glass", „fluid walls")
- Smart way to resize the UI on devices with different sizes.
- Previously pixel based design patterns, like newspapers.
  From now on: percentage based design patterns.

Fluid layout 1.

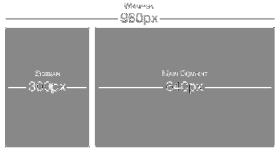
- Going from pixel to percentage based design pattern:

$$\frac{original\ width}{available\ space} = scaling\ factor$$

- Eg. 960 px / 1920 px = 50%

Fluid layout 2.

Scaling the child elements is done in a similar way.



Sidebar: 300px / 960px = 31.25%
Main Content: 640px / 960px = 66.66667%
Margin: 20px / 960px = 2.08334%

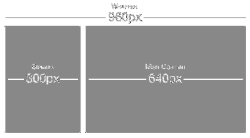
Fluid layout 3.

HTML

```
<div class="wrapper">
 <div class="sidebar"><h1>Sidebar</h1></div>
 <div class="content"><h1>Content</h1></div>
</div>
```

CSS

```
wrapper {
  width: 100%;
  margin: 0 auto;
}
.sidebar {
  width: 31.25%;
  margin-right: 2.08334%;
}
.content {
  width: 66.66667%;
}
```



Fluid images

- Images must be scaled as well.
- Simple solution: `img { max-width: 100%; }`
- Ideal solution: usage of SVG

```
<object data="your.svg" type="image/svg+xml">
 <img src="yourfallback.jpg" />
</object>
or
<img src="your.svg" onerror="this.src='your.png'">
or (CSS)
div {
   background-image: url(fallback.png);
   background-image: url(your.svg), none;
}
```

Media queries 1.

- Zooming is not an option for scaling down desktop versions to mobile devices.
- CSS 2.1:

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

- CSS 3:

```
<link rel="stylesheet" type="text/css"
 media="screen and (max-device-width: 480px)"
 href="shetland.css" />
```

Media queries 2.

- More conditions can be applied:

```
<link rel="stylesheet" type="text/css"
 media="screen and (max-device-width: 480px) and (resolution: 163dpi)"
 href="shetland.css" />
```

- CSS:

```
@media screen and (max-device-width: 480px) {
 .column {
  float: none;
 }
}
```

Media queries 3.

- Main media properties:
  - width, height, device-width, device-height,
  - orientation,
  - aspect-ratio, device-aspect-ratio,
  - color, min-color, color-index, monochrome,
  - resolution, min-resolution, max-resolution

Mobile first

- Increasing the resolutions the breakpoints can be defined. Eg. 768px, 992px and 1200px
- We create different CSS for the different resolutions:

```
@media screen and (min-width: 600px) { ... }
@media screen and (min-width: 900px) { ... }
```

- http://screensiz.es/

Testing

- Desktop browser
- http://beta.screenqueri.es/
- http://www.browserstack.com/
- Target device

Media queries 4.

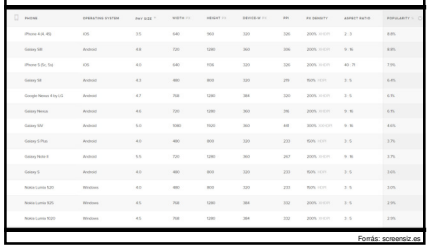
- Example: two column layout -> sidebar slides up
- HTML

```
<div class="wrapper">
 <div class="sidebar"><h1>Sidebar</h1></div>
 <div class="content"><h1>Content</h1></div>
</div>
```

- CSS

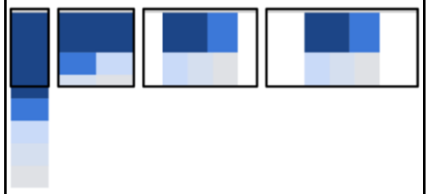
```
@media screen and (min-width: 600px) {
  .wrapper {
    width: 50%;
    margin: 0 auto;
  }
  .sidebar, .content {
    float: left;
  }
  ....
}
```

Mobile first



Forrás: screensiz.es

Example 1.



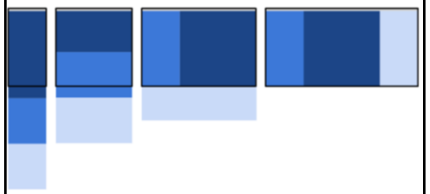
Mobile first

- Mobile user interface is designed first, than the interface is created for higher resolutions
  - designers are encouraged to create simple UI
  - mobile browsers are widely used
  - additional capabilities of mobile devices can be exploited (gyroscope, GPS, touch screen, etc.)

Responsive frameworks

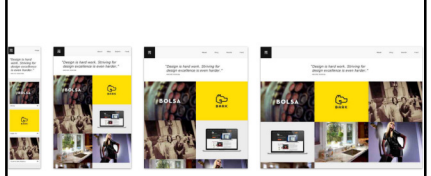
- Client-side frameworks
- Advantage
  - Faster development.
  - Reusable code.
- Disadvantage.
  - Slower.
  - Higher consumption.
  - *Bigger data traffic.*
- Bootstrap, Foundation
  - eg. http://startbootstrap.com/template-categories/all/
  - eg. http://foundation.zurb.com/templates.html

Example 2.



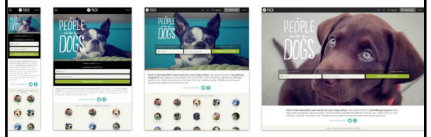
Example 3.

Example 6.



Example 4.

Example 7.



Example 5.