

RESPONSIVE DESIGN



- Design that Adapts to Different Devices

- Nikolay Chochev
- Technical Trainers
- Chochev.EU Team
- <http://chochev.eu/html/>



TABLE OF CONTENTS

- Responsive Design
- Creating Responsive Design
 - Fluid Layout
 - Flexible Images and Fonts
 - Flexible Tables and Menus
 - Media Queries
- Media Queries Everywhere
- Constructing Responsive Design





RESPONSIVE DESIGN

- Overview

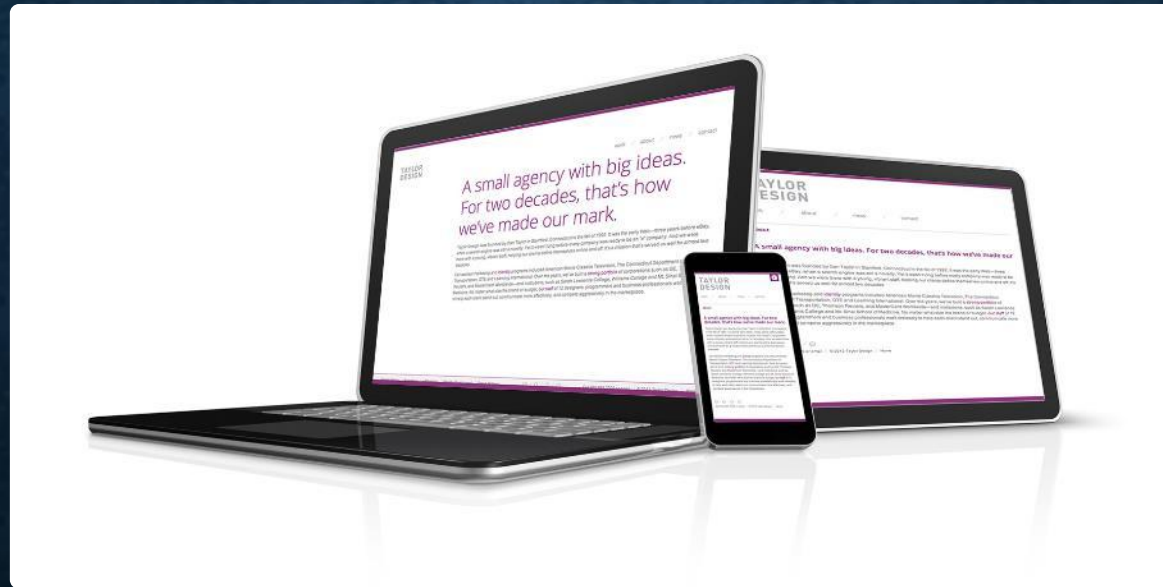
RESPONSIVE DESIGN

- **Responsive design** (adaptive design) is an approach to optimize the viewing experience on range of devices
 - Better user experience on mobile, desktop, TV
 - The UI depends on the device and device specifics
 - Screen size
 - Screen resolution
 - DPI
 - Color range
 - Etc...



RESPONSIVE DESIGN

- Live Demo



CREATING RESPONSIVE DESIGN

- Ways to Implement a Nice UI



CREATING RESPONSIVE DESIGN

- Responsive design involves one or more of the following:
 - Fluid layout
 - Use proportional values for widths, margins, etc...
 - Flexible fonts
 - Fonts based on the root
 - Flexible images
 - Images cannot go beyond their container
 - Media queries
 - Apply styles based on the client screen size

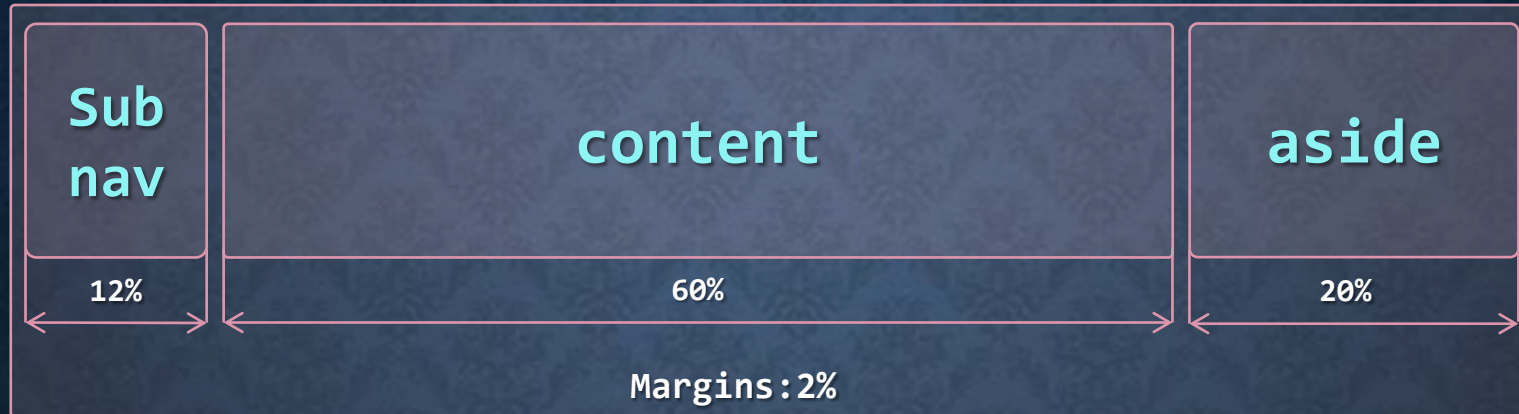
FLUID LAYOUT

- Make the Elements Flow on the Screen



FLUID LAYOUT

- Fluid layout uses proportional sizes



- Pros:
 - UI responds better to the client resolution
 - Spares code on media queries
- Cons:
 - More whitespace on large screens (TV)

FLUID LAYOUT

- Live Demo





**FLEXIBLE
IMAGES**

FLEXIBLE IMAGES

- Fluid design gets broken when using elements with fixed size
 - By concept images are always with fixed size
- Example:
 - Resolution: **1024px**, container with width: **60%** ($= 60\% * 1024 = 614.4\text{px}$) and an image with width: **500px**
 - Seems OK
 - When the resolution becomes **780px**, the container's width is still **60%** ($= 468\text{px}$), and the images width is still **500px**
 - The image overflows its container

MAKING IMAGES FLEXIBLE

- The fix to the image overflow is simple

- Just a reset in the top of the CSS

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

- `max-width` overrides the `width` property
 - If the image size is larger than the container's size → the image gets the entire container's width

FLEXIBLE IMAGES

- Live Demo



FLEXIBLE FONTS



FLEXIBLE FONTS

- Flexible fonts means proportional font size
 - Based on the context (parent)
 - Instead of pixels use proportional values (**em**)
- Make all font sizes based on the context
 - To change the **font-size** of all elements just change the context's font-size

FLEXIBLE FONTS (2)

- Making fonts "responsive" needs a little math
 - $\text{ems} = \text{target} / \text{root}$
 - $1.4375\text{em} = 23\text{px} / 16\text{px}$

```
body { font-size: 16px; }  
  body header { font-size: 23px; } // 23 / 16 = 1.4375
```



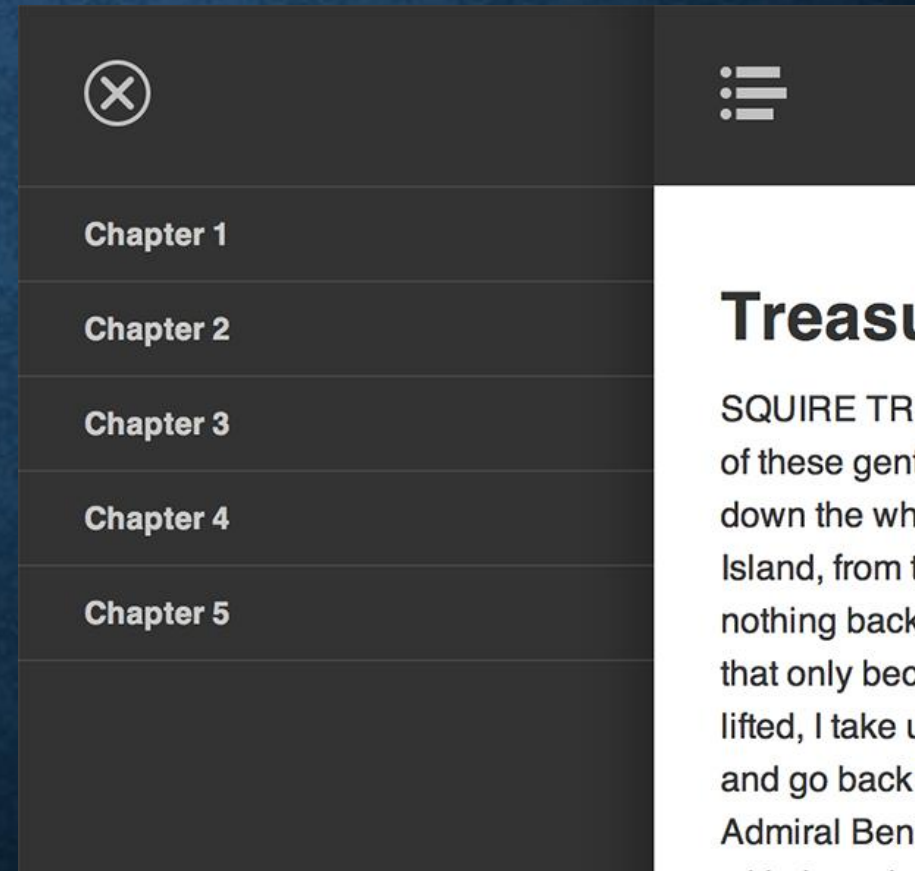
```
body { font-size: 16px; }  
  body header { font-size: 1.4375em; }
```

FLEXIBLE FONTS

- Live Demo

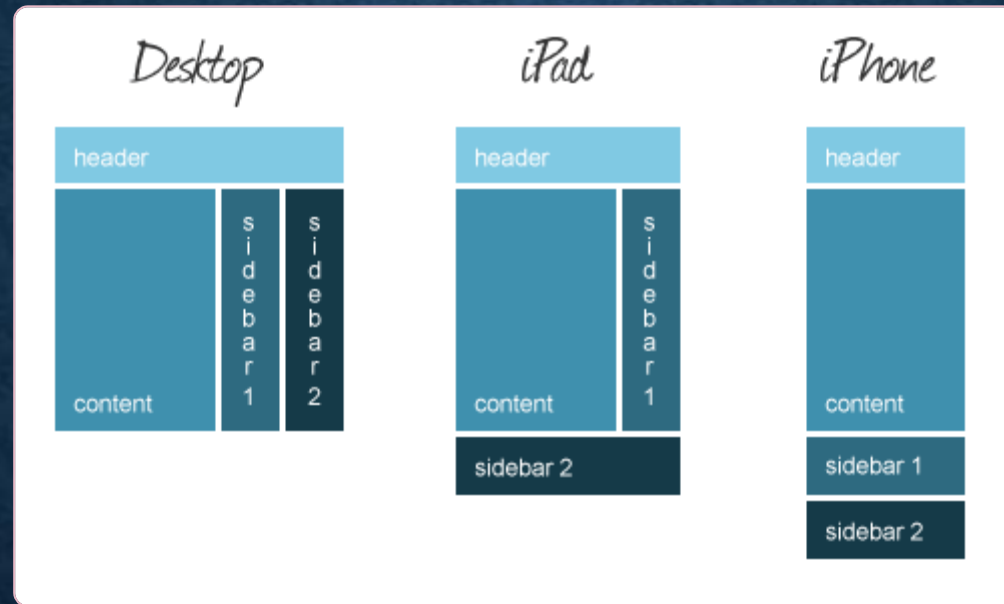


FLEXIBLE TABLES AND MENUS



RESPONSIVE TABLES AND MENUS

- Responsive tables
 - Tables are ugly and not scalable
 - <http://css-tricks.com/responsive-data-table-roundup/>
- Responsive menus
 - Menus take to much space
 - <http://css-tricks.com/responsive-menu-concepts/>



MEDIA QUERIES

MEDIA QUERIES

- Media queries are part of CSS 3
 - Supported in all major browsers
- A media query consists of a media type and at least one expression
 - By using media features like width, height and color
- MQ change the presentation of content
 - Not the content itself

MEDIA QUERIES (2)

- Media queries apply CSS styles on certain conditions (**media type** and **expression**)

```
.box {width: 250px; height: 250px; display: inline-block}

@media only screen and (max-width: 1024px) {
    .box { width: 300px; height: 300px; }
}

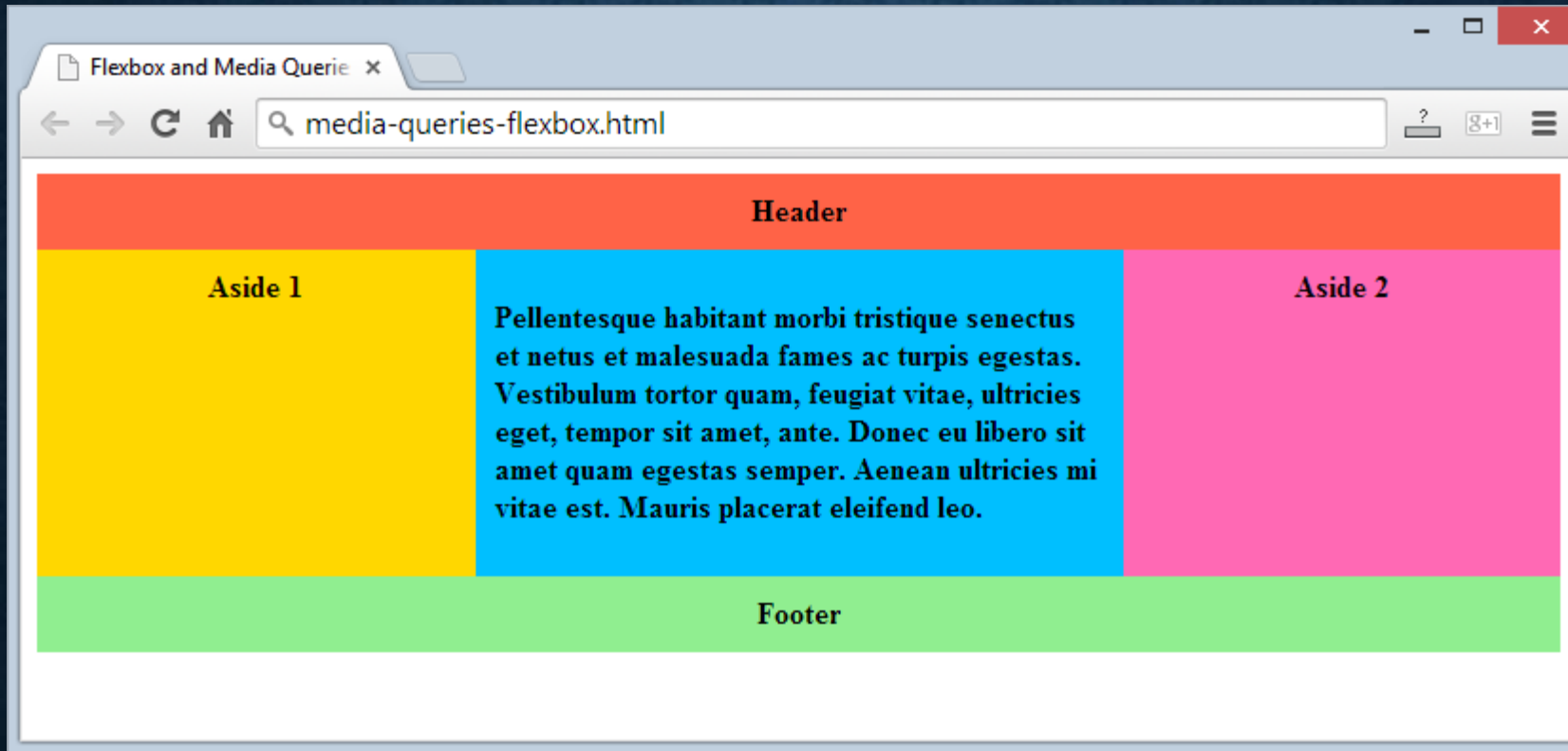
@media only screen and (max-width: 960px) {
    .box { width: 310px; height: 310px;}
}

@media only screen and (max-width: 480px) {
    .box { display: block; width: 95%; height: 95%; }
}
```



MEDIA QUERIES

- Live Demo



MEDIA QUERIES AND FLEXBOX

SUMMARY

- Responsive design involves
 - Fluid layouts
 - Flexible Images
 - Flexible fonts
 - Media queries
- Media queries syntax:



```
@media screen and (max-width: 480px) {  
    /* Apply styles for small devices (phones) */  
}
```