

CSS (CASCADING STYLE SHEETS)

Box Model

Positioning and visualization

Responsive web

CSS Bootstrap

Methodologies

Bruno Oliveira: bmo@estg.ipp.pt

Marco Gomes: mfg@estg.ipp.pt

Miguel Andrade: mja@estg.ipp.pt

Overview

2

- HTML -> Content | CSS -> Presentation
- Styling HTML with CSS
 - ▣ **CSS** stands for **C**ascading **S**tyle **S**heets
 - ▣ CSS describes **how HTML elements are to be displayed on screen, paper, or in other media.**
 - ▣ **CSS saves a lot of work.** It can control the layout of multiple web pages all at once.
- CSS can be added to HTML elements in 3 ways:
 - ▣ **Inline** - by using the style attribute in HTML elements
 - ▣ **Internal** - by using a <style> element in the <head> section
 - ▣ **External** - by using an external CSS file

Overview

3

- Set up a class in CSS:
 - ▣ If the style can be applied to more than one element on a page
 - ▣ Use the notation. (dot) in the style sheet
 - ▣ Use the class attribute in HTML

- Set an ID in CSS:
 - ▣ If the template is specific to a single element of a page
 - ▣ Use # notation in style sheet
 - ▣ Use the id attribute in HTML

Overview

4

I.D Selector



#menu-item

Declarations = Property + Value



{ color: blue; }

Class Selector



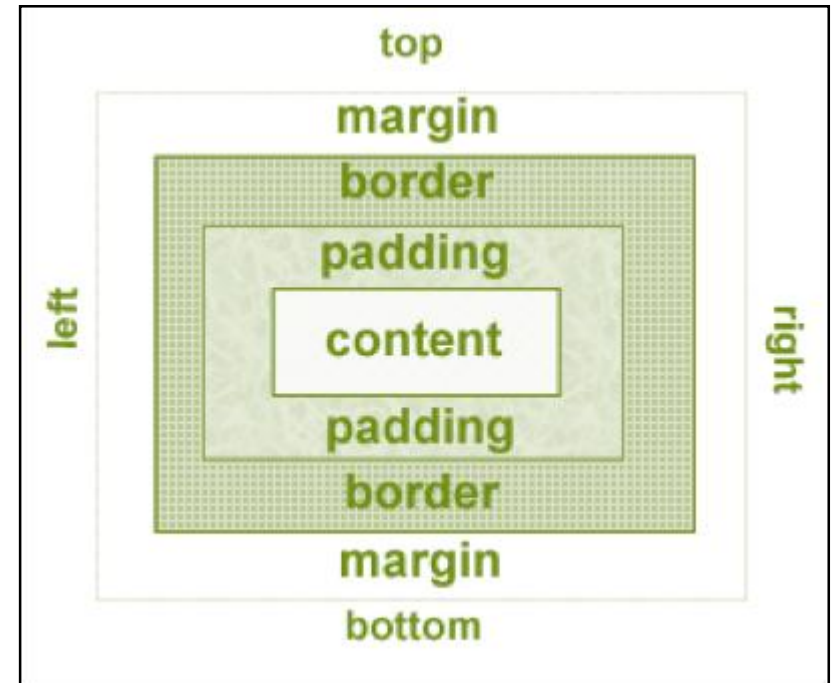
.menu-class

{ color: blue; }

Box Model

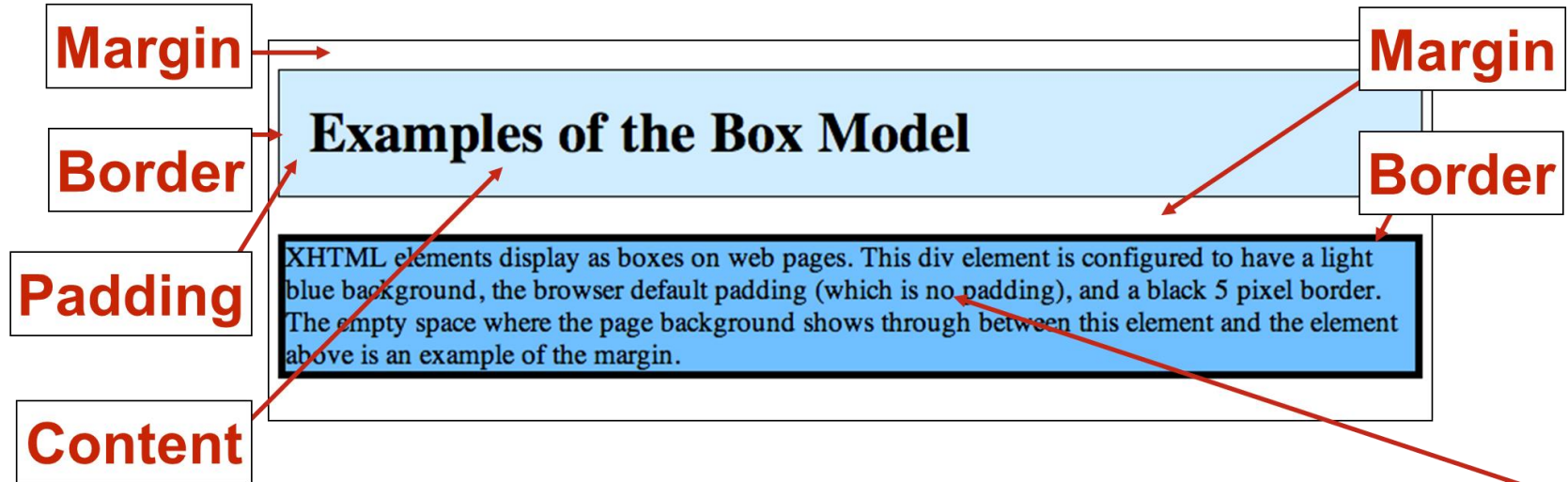
5

- Content
 - ▣ Text and elements
- *Padding*
 - ▣ Space between content and border
- *Border*
 - ▣ Between padding and margin
- *Margin*
 - ▣ Creates space around elements, outside of any defined borders



Box model (in action)

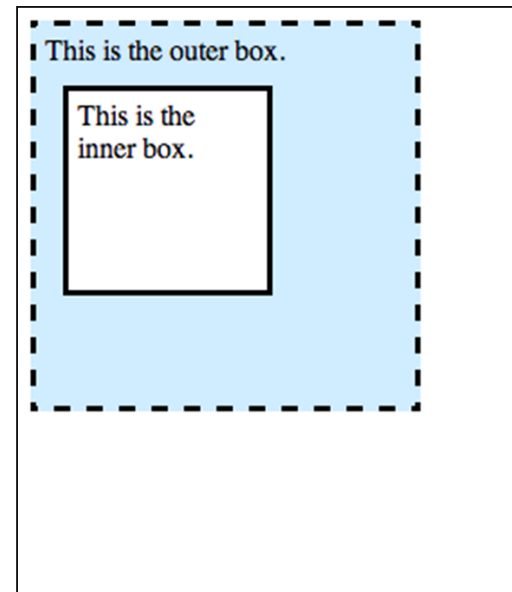
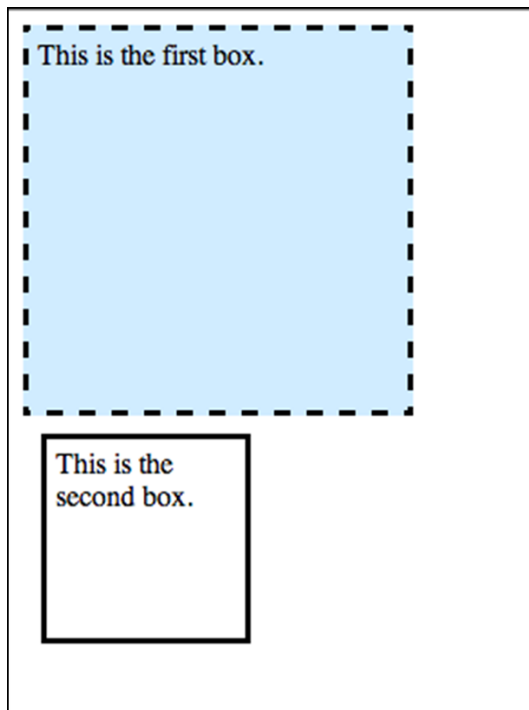
6



Box model (normal flow)

7

- Display elements in the browser in the order they are encoded in the Web page document



Relative positioning

8

- ❑ Changes the location of an element relative to the point where it would originally appear:

Relative Positioning

This paragraph uses CSS relative positioning to be placed 30 pixels in from the left side.

```
h1 { background-color:#cccccc;  
      padding: 5px;  
      color: #000000; }  
#myContent { position: relative;  
              left: 30px;  
              font-family: Arial,sans-serif; }
```


Absolute positioning

9

- Specifies precisely the location of an element in the browser window

Absolute Positioning

This paragraph is contained in a div which is 300 pixels wide and uses CSS absolute positioning to be placed 200 pixels in from the left and 100 pixels down from the top of the browser window.

```
h1 { background-color: #cccccc;
      padding: 5px;
      color: #000000; }
#content {position: absolute;
          left: 200;
          top: 100;
          font-family: Arial,sans-serif;
          width: 300; }
```

Positioning and display

10

□ HTML without CSS

Trillium Media Design

Our professional staff takes pride in its working relationship with our clients by offering personalized services that listen to their needs, develop their target areas, and incorporate these items into a well-presented web site that works.

- Home
- [About](#)
- [Services](#)

□ HTML with CSS

Trillium Media Design

- Home
- [About](#)
- [Services](#)

Our professional staff takes pride in its working relationship with our clients by offering personalized services that listen to their needs, develop their target areas, and incorporate these items into a well-presented web site that works.

Display (property)

11

- Set how and if an element is displayed:

- ▣ Element will not be displayed

- ```
display: none;
```

- ▣ Element is processed as a block element - even though it is actually an inline element, such as a hyperlink

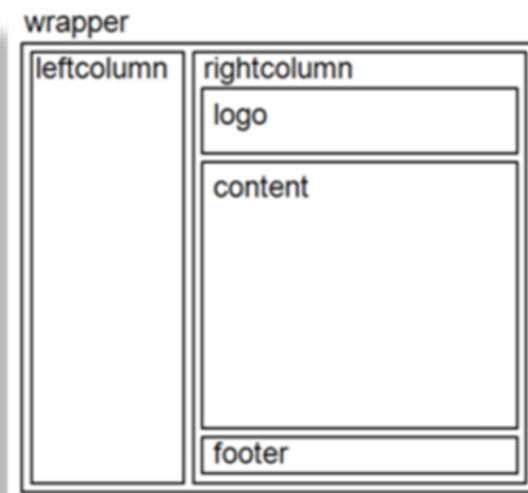
- ```
display: block;
```

- ▣ The element will be rendered as an inline element - even though it is actually a block element - such as a ``

- ```
display: inline;
```

# CSS: Page layout - two columns (LEFT and NAV)

12



[Home](#)  
[Menu](#)  
[Directions](#)  
[Contact](#)

## Lighthouse Island Bistro

### Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

### Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and cinammon rolls.

### Lunchtime is Anytime

Savor delicious wraps and sandwiches on hearty, whole-grain breads with locally-grown salad, fruit, and vegetables.

### Panoramic View

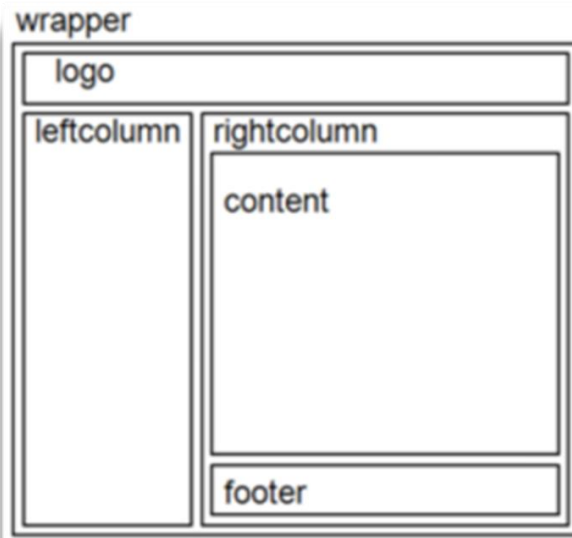
Take in some scenery! The top of our lighthouse offers a panoramic view of the countryside. Challenge your friends to climb our 100-stair tower.



Copyright © 2012

# CSS: Page layout - two columns (TOP, LOGO, LEFT NAV)

13



## Lighthouse Island Bistro

- [Home](#)
- [Menu](#)
- [Directions](#)
- [Contact](#)

### Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

### Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and cinnamon rolls.

### Lunchtime is Anytime

Savor delicious wraps and sandwiches on hearty, whole-grain breads with locally-grown salad, fruit, and vegetables.

### Panoramic View

Take in some scenery! The top of our lighthouse offers a panoramic view of the countryside. Challenge your friends to climb our 100-stair tower.



Copyright © 2012

# CSS layout techniques

14

- ❑ **Tables:** Don't use them to website layout (past)
- ❑ **Float:** Avoid! ([https://www.w3schools.com/cssref/pr\\_class\\_float.asp](https://www.w3schools.com/cssref/pr_class_float.asp)) (past/present)
- ❑ **CSS Frameworks:** Like Bootstrap, are a good choice! (<https://www.w3schools.com/bootstrap4/default.asp>)
- ❑ **Flexbox:** Preferred approach for most web developers! ([https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)) (present)
- ❑ **Grid Layout:** Powerful technique that is relatively recent! (future?) ([https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp))
- ❑ Use them Together!
- ❑ Additional (interesting) resources:
  - ❑ <https://tutorialzine.com/2017/03/css-grid-vs-flexbox>
  - ❑ <https://www.codementor.io/codementorteam/4-different-html-css-layout-techniques-to-create-a-site-85i9t1x34>

# Responsive web

15

- A collection of techniques that allow your website to respond to your users environment based on screen-size, platform and orientation.
- Responsive Web Design makes your web page look good on all devices (desktops, tablets, and phones).
- Responsive Web Design is about using HTML and CSS to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

# Responsive web

16

- Combination of
  - ▣ Fluid grids
  - ▣ Flexible images
  - ▣ Media queries
- Golden rule: **never make users scroll horizontally!**



# Responsive web

17

- Combination of
  - ▣ Fluid grids
  - ▣ Flexible images
  - ▣ Media queries
- Golden rule: **never make users scroll horizontally!**
- Sizing elements matters

# Responsive web

18

- Viewport meta tag

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

- Default action for most mobile devices is to zoom out and scale the webpage
- Sets the browser viewport's initial width and scale

# Responsive web

19

## □ Max-width

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

```
img.foo {
 width: 300px;
}
```

- Relative sizing is not the (full) solution
  - ▣ Too big on desktop, too small on mobile
  - ▣ One relative-sized layout for all

# Responsive web

20

- Can use media queries to select layouts
- Media queries examples

```
/* elements in common for phones and tablets */
```

```
@media (max-width: 480px) {
 /* layout for phones */
}
```

```
@media (max-width: 720px) {
 /* layout for tablets */
}
```

# Responsive web

21

- Media queries:
  - ▣ Select layout by device (Firefox and Chrome have options to emulate small screens)
    - 320, 480px: phone
    - 768px: tablet
    - 992px: laptop
    - 1200px: widescreen
- One stream of content
  - Find the best layout for devices and viewport sizes
  - Manipulate blocks of content for the needs of the layout

# Responsive web

22

## □ Media queries:

```
/* small */
```

```
@media (min-width: 480px) {
 /* medium */
}
```

```
@media (min-width: 720px) {
 /* large */
}
```

# Bootstrap

23

- ❑ Bootstrap is a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.
  - ▣ Provides a clean and uniform solution for building an interface for developers.
  - ▣ It contains beautiful (?) and functional built-in components which are easy to customize.
  - ▣ It also provides web based customization.
  - ▣ And best of all it is an open source.

# Bootstrap

24

- You can download the latest version of Bootstrap from <http://getbootstrap.com/>
- What Bootstrap Package includes?
  - ▣ **Scaffolding:** Bootstrap provides a basic structure with Grid System, link styles, background. This is covered in detail in the section Bootstrap Basic Structure
  - ▣ **CSS:** Bootstrap comes with feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section Bootstrap with CSS.



# Bootstrap

25

- **Components:** Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, popovers, and much more. This is covered in detail in the section Layout Components.
- **JavaScript Plugins:** Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section Bootstrap Plugins.
- **Customize:** You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

# Bootstrap

26

- HTML Template: a basic HTML template using Bootstrap would look like as this:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Bootstrap 101 Template</title>
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <!-- Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet">

 <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
 queries -->
 <!-- WARNING: Respond.js doesn't work if you view the page
```

# Bootstrap

27

- Here you can see the jquery.js and bootstrap.min.js and bootstrap.min.css files are included to make a normal HTML file to Bootstrapped Template.

```
via file:// -->
<!--[if lt IE 9]>
 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/
 html5shiv.js"></script>
 <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/
 respond.min.js"></script>
<![endif]-->
</head>
<body>
 <h1>Hello, world!</h1>

 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script src="https://code.jquery.com/jquery.js"></script>
 <!-- Include all compiled plugins (below), or include individual files
 as needed -->
 <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

# Bootstrap

28

- HTML Template: a basic HTML template using Bootstrap would look like as this:

```
via file:// -->
<!--[if lt IE 9]>
 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/
 html5shiv.js"></script>
 <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/
 respond.min.js"></script>
<![endif]-->
</head>
<body>
 <h1>Hello, world!</h1>

 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
 <script src="https://code.jquery.com/jquery.js"></script>
 <!-- Include all compiled plugins (below), or include individual files
 as needed -->
 <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

# Bootstrap

29

## □ Bootstrap Grid System

- ▣ Rows must be placed within a **.container** class for proper alignment and padding.
- ▣ Use rows to create horizontal groups of columns.
- ▣ Content should be placed within columns, and only columns may be immediate children of rows.
- ▣ Predefined grid classes like **.row** and **.col-xs-4** are available for quickly making grid layouts.
- ▣ Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on **.rows**.
- ▣ Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three **.col-xs-4**.

# Bootstrap (Grid System)

30

## □ Table summarizing grid system

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	750px	970px	1170px
Class prefix	<b>.col-xs-</b>	<b>.col-sm-</b>	<b>.col-md-</b>	<b>.col-lg-</b>
# of columns	12	12	12	12
Max column width	Auto	60px	78px	95px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

# Bootstrap (Grid System)

31

## ■ Basic grid structure:

```
<div class="container">
 <div class="row">
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
 </div>
 <div class="row">...</div>
</div>
<div class="container">....
```

## ■ Details

- `<div class="container">...</div>` element is added to ensure proper centering and maximum width for layout.
- Once container is added, next you need think in terms of rows. Add `<div class="row">...</div>` and columns `<div class="col-md-6"></div>` inside rows.
- Every row in the grid is made up of 12 units and you can define the desired size of your columns using those units. In our example we have two columns each made of 6 units wide i.e  $6+6=12$ .

# Bootstrap (Media queries)

32

- Recall: a media query is a really fancy term for "conditional CSS rule". It simply applies some CSS based on certain conditions set forth. If those conditions are met, the style is applied.
- Media Queries in Bootstrap allow you to move, show and hide content based on viewport size. Following media queries are used in LESS files to create the key breakpoints in the Bootstrap grid system.



# Bootstrap (Media queries)

33

## □ Example:

```
/* Extra small devices (phones, less than 768px) */
/* No media query since this is the default in Bootstrap */

/* Small devices (tablets, 768px and up) */
@media (min-width: @screen-sm-min) { ... }

/* Medium devices (desktops, 992px and up) */
@media (min-width: @screen-md-min) { ... }

/* Large devices (large desktops, 1200px and up) */
@media (min-width: @screen-lg-min) { ... }
```

## □ Occasionally these are expanded to include a max-width to limit CSS to a narrower set of devices.

```
@media (max-width: @screen-xs-max) { ... }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }
@media (min-width: @screen-lg-min) { ... }
```

# Methodologies

34

- BEM (Block Element Modifier)
  - ▣ Provides a way to arrange your CSS classes into independent modules;
  - ▣ A **Block** represents an object in your website (a menu. A form etc.)
  - ▣ An **Element** is a component within the block that performs a particular function (a login button)
  - ▣ A **Modifier** represents a variation of a block (e.g. a search input field with a particular button style)
  - ▣ <https://en.bem.info/methodology/>

# Methodologies

35

## □ SMACSS

- A framework for CSS which is more *styling guide*;
- A **Base** is used for defaults like html, body, a etc.
- **Layout** divides a page into sections with elements like header, footer, etc.
- **Module** is a reusable modular element in the design
- **State** is used for the variations for each element using the prefixed **is-** (e.g. is-active)
- **Theme** is similar to **state** but defines how modules and layouts will look.
- <https://smacss.com/>

# References

36

- ❑ <https://validator.w3.org> – HTML validator
- ❑ <https://jigsaw.w3.org/css-validator/> - CSS Validator
- ❑ <https://en.bem.info/methodology/> - BEM
- ❑ <https://smacss.com/> - SMACSS
- ❑ <https://www.w3schools.com/css/>

# Next week..

37

- JavaScript
- Study suggestions:
  - ▣ [https://www.w3schools.com/js/js\\_intro.asp](https://www.w3schools.com/js/js_intro.asp)
  - ▣ <https://www.codecademy.com/learn/introduction-to-javascript>

# CSS (CASCADING STYLE SHEETS)

Box Model

Positioning and visualization

Responsive web

CSS Bootstrap

Methodologies

Bruno Oliveira: [bmo@estg.ipp.pt](mailto:bmo@estg.ipp.pt)

Marco Gomes: [mfg@estg.ipp.pt](mailto:mfg@estg.ipp.pt)

Miguel Andrade: [mja@estg.ipp.pt](mailto:mja@estg.ipp.pt)