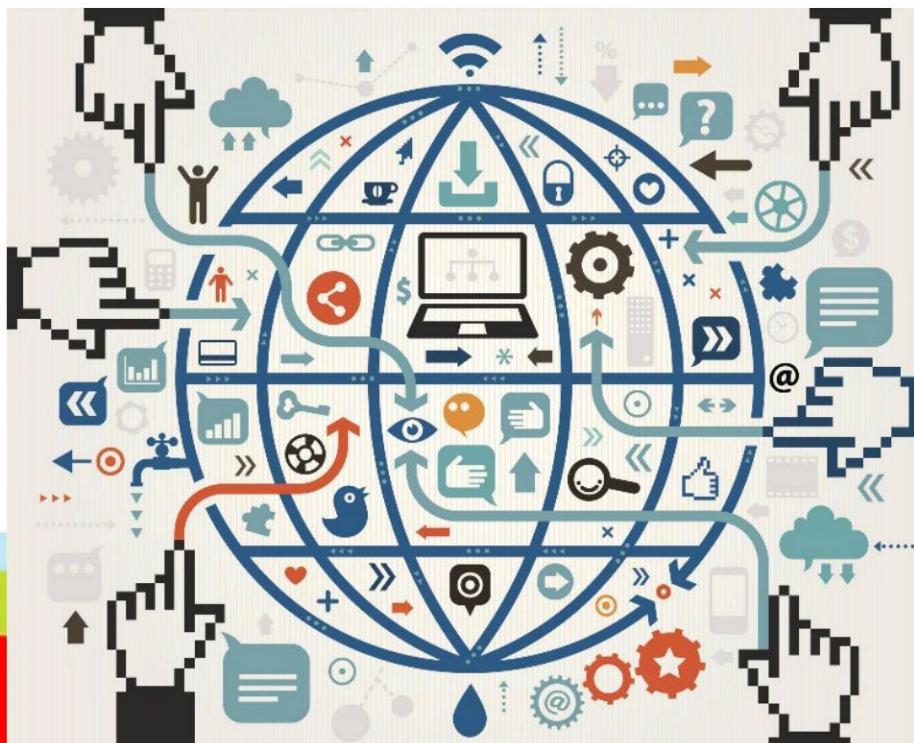


CEU | *Universidad  
San Pablo*



CSS

Web Systems I

Degree in Information Systems Engineering Álvaro Sánchez  
Picot

alvaro.sanchezpicot@ceu.es

v20231002

Based on the work of: •  
David González Márquez

# Introduction

- CSS: Cascading Style Sheets
- Describes the presentation of the document
- Defines layout, colors, fonts, variations according to the device...
- Allows multiple documents to use the same style
- Cascading: priority to see which rule is applied



# Introduction

- It is managed by the World Wide Web Consortium (W3C)
- CSS1: First version in 1996
- CSS3: Most current version, modular
- CSS versions are actually known as Levels.
  - Each level is a superset of the previous one – Each module has a specific level ([info.](#))
- It is updated periodically
  - Annual snapshots are taken ([2021](#), [2022](#), [2023](#))
  - [CSS](#)

## [Specifications](#)



# Introduction

- How to insert CSS – External file

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link rel="stylesheet" href="mystyle.css">  
  </head>  
  <body>  
    mystyle.css file: body  
    { background-color: lightblue;  
    } h1  
      { color: navy;  
      margin-left: 20px;  
    }  
</body>  
</html>
```



# Introduction

- How to insert CSS – Internal

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background -color: linen;

      } h1
      { color: maroon;
        margin -left: 40px;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```



# Introduction

- How to insert CSS – Online

```
<!DOCTYPE html>  
<html>  
  <body>  
    <h1 style="color:blue;text-align:center;">This is a heading</h1> <p  
      style="color:red;">This is a paragraph.</p> </body>  
  
</html>
```



# When to use each shape?



# Introduction

- HTML was never intended to include tags for formatting
- HTML is used to describe content
- HTML 3.2 introduced tags like `<font>` and the nightmare began
  - As a result of this, the W3C created CSS
- CSS should never be inserted inline
- An external file should be used to reuse the styles



## Introduction

- Comments

- Syntax: /\*...\*/
  - They can be multiline
  - They can be used in the HTML within the <style></style> tag p { color: red; /\*

Set

text color to red \*/

}

/\* This is  
to multi-line  
comment \*/



# Introduction

- CSS works based on rules, that is, declarations about the style of one or more elements.

```
h1 {color: red;}
```

- h1 is the **selector**.
  - Indicates the HTML elements that will be affected.
- {color: red;} is the **declaration block**.
  - Each block contains one or more statements separated by ; –
  - color is the name of a property
  - red is the value



# Selectors

- Simple
- Combination
- Pseudo-classes
- Pseudo-elements
- Attributes



# Simple selectors

- Based on the name of an element (p, a, body, div...) • Syntax:  
element\_name {}

p

```
{ text-align: center; color:  
red;  
}
```



# Simple selectors

- Based on the id attribute of an element (which has to be unique)
- Syntax:

```
#id_name {}
```

```
#para1
```

```
{ text-align: center; color:
```

```
red;
```

```
}
```

-----

```
<p id="para1">Hello World!</p>
```



# Simple selectors

- Based on the class attribute of an element (can be repeated) •

Syntax: .class\_name {}

.center

```
{ text-align: center; color:  
red;  
}
```

---

```
<h1 class="center">Red and center-aligned heading</h1> <p  
class="center">Red and center-aligned paragraph.</p>
```



# Simple selectors

- Based on the class attribute of an element (it can be repeated)
- It can be specified that it only affects a certain type of elements
- Syntax:  
`element_name.class_name { }`

```
p.center { text-align: center; color: red; }
```

---

```
<h1 class="center">This heading will not be affected</h1> <p class="center">Red and center-aligned paragraph.</p>
```



# Simple selectors

- Universal selector
- Select all HTML elements
- Syntax: \* { }

\*

```
{ text-align: center;  
color: blue;  
}
```

---

```
<p>Every element on the page will be affected by the style.</p> <p id="para1">Me too!  
</p> <p>And me!</p>
```



# Simple selectors

- Grouping •

Syntax: selector\_name1, selector\_name2... { }

```
h1
  { text-align: center; color:
    red;

} h2 {
  text-align: center;
  color: red;

} p
  { text-align: center; color:
    red;
}
```



```
h1, h2, p
  { text-align: center; color:
    red;
}
```



## Combination selectors

- CSS Combinators
- Relate selectors
- Guys:
  - Descendant
  - Son
  - Adjacent brother
  - Brother General



# Combination selectors – Descending

- Selects all descendant elements of the specified element

- Syntax:  
simple\_selector1 simple\_selector2 {}

```
div p  
{ background-color: yellow;  
}
```

```
<div>  
  <p>Paragraph 1 in the div.</p>  
  <section>  
    <p>Paragraph 2 in the div.</p> </  
    section> </  
  div>  
  <p>Paragraph 3. Not in a div .</p>
```



## Combination selectors – Son

- Selects all elements that are children of the specified element • Syntax: simple\_selector1 > simple\_selector2 {}

div > p

```
{ background-color: yellow;  
}
```

```
<div>  
    <p>Paragraph 1 in the div.</p>  
    <section>  
        <p>Paragraph 2 in the div.</p> </  
        section> </  
        div>  
    <p>Paragraph 3. Not in a div .</p>
```



# Combination Selectors – Adjacent Brother

- Select all elements that are siblings (have the same parent) and they are immediately below
- Syntax: simple\_selector1 + simple\_selector2 {}

div + p

```
{ background-color: yellow;  
}
```

<div>

```
<p>Paragraph 1 in the div.</p>  
<p>Paragraph 2 in the div.</p> </div>
```

```
<p>Paragraph 3. Not in a div.</p> <p  
>Paragraph 4. Not in a div.</p>
```



# Combination Selectors – General Brother

- General sibling
- Select all elements that are siblings and appear after

simple\_selector1 ~ simple\_selector2 { }

div ~ p

```
{ background-color: yellow;  
}
```

```
<p>Paragraph 1.</p>  
<div>  
    <p>Paragraph 2.</p>  
</div>  
<p>Paragraph 3.</p>  
<code>Some code.</code>  
<p> Paragraph 4.</p>
```



# Pseudoclass selectors

- Define a special state of an element

- List with pseudoclasses

- Syntax:

```
selector:pseudo-class { }
```

:hover — Select the element that has the mouse over it  
:focus — Select the element that has the focus  
:first-child — Select the first child  
:last-child — Select the last child



# Pseudo-element selectors

- They identify a special part of an element •

Syntax: selector::pseudo-element { }

::after ↪ To insert something before. Use with content ::before property ↪ To insert something after.

Use with content ::first-letter ↪ Select the first letter ::first-line ↪ Select the first line

::selection ↪ The user's selection



# Attribute selectors

- Select an element with a specific attribute
- Can be associated with a selector
- Syntax: selector[attribute] { }

a[target]

```
{ background-color: yellow;  
}
```

---

```
<a href="http://...">w3schools.com</a> <a  
href="http://..." target="_blank">disney.com</a> <a href ="http://..."  
target="_top">wikipedia.org</a>
```



## Attribute selectors with value

- Select an element with a specific attribute and value
- Can be associated with a selector
- Syntax: selector[attribute="value"] { }

```
a[target="_blank"] {  
    background-color: yellow;  
}
```

---

```
<a href="http://...">w3schools.com</a> <a  
    href="http://..." target="_blank">disney.com</a> <a href ="http://..."  
    target="_top">wikipedia.org</a>
```



# Attribute selectors – Others

[attribute~="value"]

- Selects an element with an attribute whose value contains a specific word

[attribute|= "value"]

- Selects an element with an attribute whose value begins with a specific word

[attribute^= "value"]

- Select an element with an attribute whose value starts in a specific way

[attribute\$= "value"]

- Selects an element with an attribute whose value ends in a specific way

[attribute\*= "value"]

- Selects an element with an attribute whose value contains a specific value



# Statements

## Box model •

**Content:** The content of the box as text or images •

**Padding:** Empty area around the content. It is transparent

• **Border:** Border around the padding and content

• **Margin:** Empty area outside the border. It is transparent



# Statements – margin

- Space around the element on the outside •  
margin-top, margin-right, margin-bottom, margin-left •

Values (can be negative):

- car
- Length (px, pt, cm...) –
- % with respect to the width of the element
- inherit: inherited from the

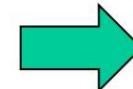
parent • [More information](#)



## Statements – margin

- Can collapse
- Syntax:
  - margin: top right bottom left; – margin: top right bottom;
  - margin: top+bottom right+left;
  - margin: top+bottom+right+left;

```
p  
{ margin-top: 100px;  
margin-right: 150px;  
margin-bottom: 100px;  
margin-left: 0;  
}
```



```
p  
{ margin: 100px 150px 100px 0;  
}
```

## Declarations – padding

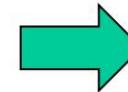
- Space around the content of the element and inside the borders • padding-top
  - padding-right
  - padding-bottom
  - padding-left •
- Values (cannot be negative): – Length (px, pt, cm...) – %  
with regarding the width  
of the element – inherit: inherited from  
the parent
- More information [\\_\\_\\_\\_\\_](#)



## Declarations – padding

- Can collapse
- Syntax:
  - padding: top right bottom left; – padding: top right bottom;
  - padding: top+bottom right+left;
  - padding: top+bottom+right+left;

```
p  
{ padding-top: 100px;  
padding-right: 150px;  
padding-bottom: 100px;  
padding-left: 0;  
}
```



```
p  
{ padding: 100px 150px 100px 0;  
}
```

## Statements

width and height • Width and height  
of the content • Then you will also  
have to add the padding, the border and the margin  
• Values:  
- car  
– Length (px, pt, cm...) –  
% with respect to the width of the containing block  
–initial  
– inherit: inherited from the father



# Statements

max-width, max-height, min-width, min-height •

Maximum/minimum width and height of the content

- Values:

- Length (px, pt, cm...) –

- % with respect to the width of the containing block –

- none: there is no maximum/minimum



## Declarations – text

- color
- background-color •  
text-align: left/right/center/justify;
- direction
- vertical-align
- text-decoration
- text-transform: uppercase/lowercase/capitalize; • text-indent:  
Indentation of the first line



## Declarations – text

- letter-spacing • line-height • word-spacing • white-space
- text-shadow

More information



## Statements – Font

- font-family: "Times New Roman", Times, serif; • font-style: normal/italic/oblique;
- font-weight: normal/bold;
- font-variant: normal/small-caps;
- font-size
- font: font-style font-variant font-weight font-size/line-height font-family;

[More information](#)



# Statements – position

- Positioning type used for the element
- Values:
  - static
    - Placed according to the normal flow of the page. Default.
  - relative
    - Relative to its default position.
  - fixed
    - Relative to the viewport, it does not move when scrolling.
  - absolute
    - Relative to the closest ancestor

# Statements

- Values (cont.):
  - sticky
    - It gets “stuck” when scrolling
- To position also use:
  - top
  - right
  - bottom
  - left
  - z-index

[More information](#)



## Values – Units

- Absolute
  - cm
  - mm
  - in: inches (1in = 2.54cm) –
  - px: pixels, relative to the device –
  - pt: points (1pt = 1/72 in)



## Values – Units

- Relative
  - em: relative to the font size. 2em is twice the current font size
  - rem: relative to the font size of the root element. It is usually equivalent to 16px.
  - ch: relative to the size of zero
  - %: relative to the parent element

[More information](#)



## Values – Colors

- Names: List of color names
- RGB(Red, Green, Blue): `rgb(255, 99, 71)`
- HEX: `#ff6347`
- HSL (Hue, Saturation, Lightness): `hsl(9, 100%, 54%)` • RGBA  
(RGB + Alpha): `rgba(255, 99, 71, 0.5)` • HSLA (HSL  
+ Alpha): `hsla(9, 100%, 64%, 0.5)`

More information



## Several

- Rounded edges
- Gradients
- Transitions
- Tooltips •

Multiple columns • Media  
queries (responsive)



## Specificity

- Conflict resolution
- The more specific an element, the higher priority
- id > class > element
- In case of equality, the one that appears later in the file

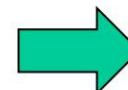
More information



# !important

- Add more importance to a property or value •
- Override all rules for that property

```
#myid {  
    background-color: blue;  
  
} .myclass  
{ background-color: gray;  
  
} p  
{ background-color: yellow !important;  
}
```



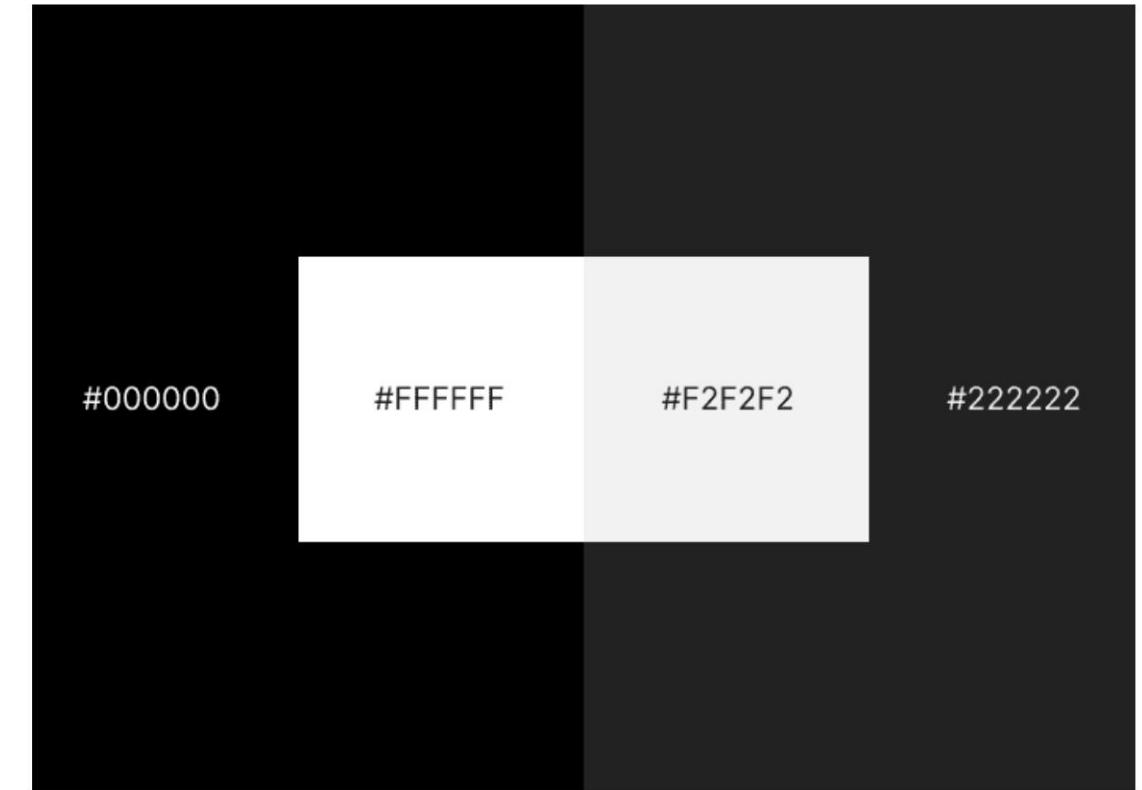
```
<p id="myid" class="myclass">  
    test text  
</p>
```



# Design recommendations

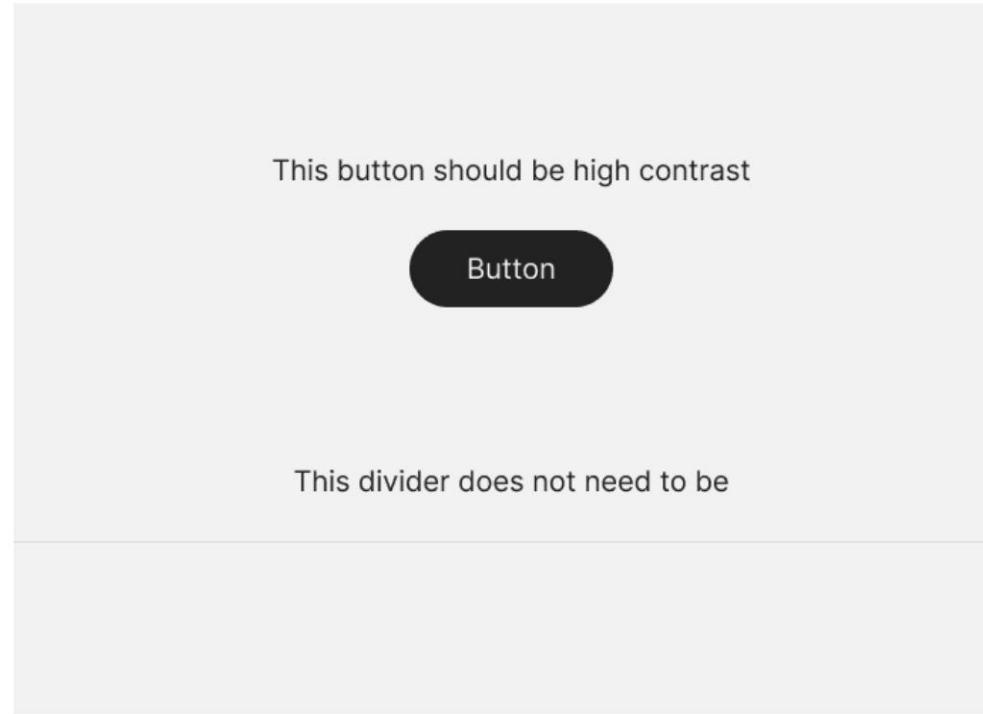
- Do not use black or white •

Use almost black and almost white



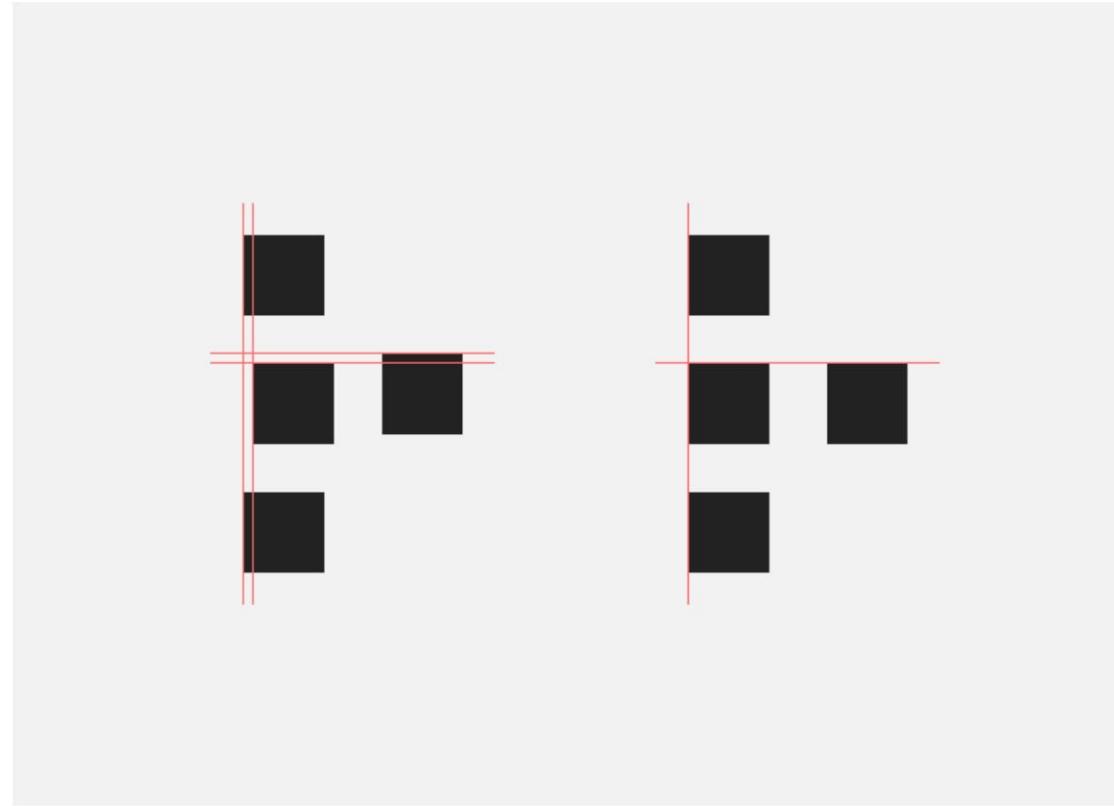
# Design recommendations

- High contrast for important elements such as buttons



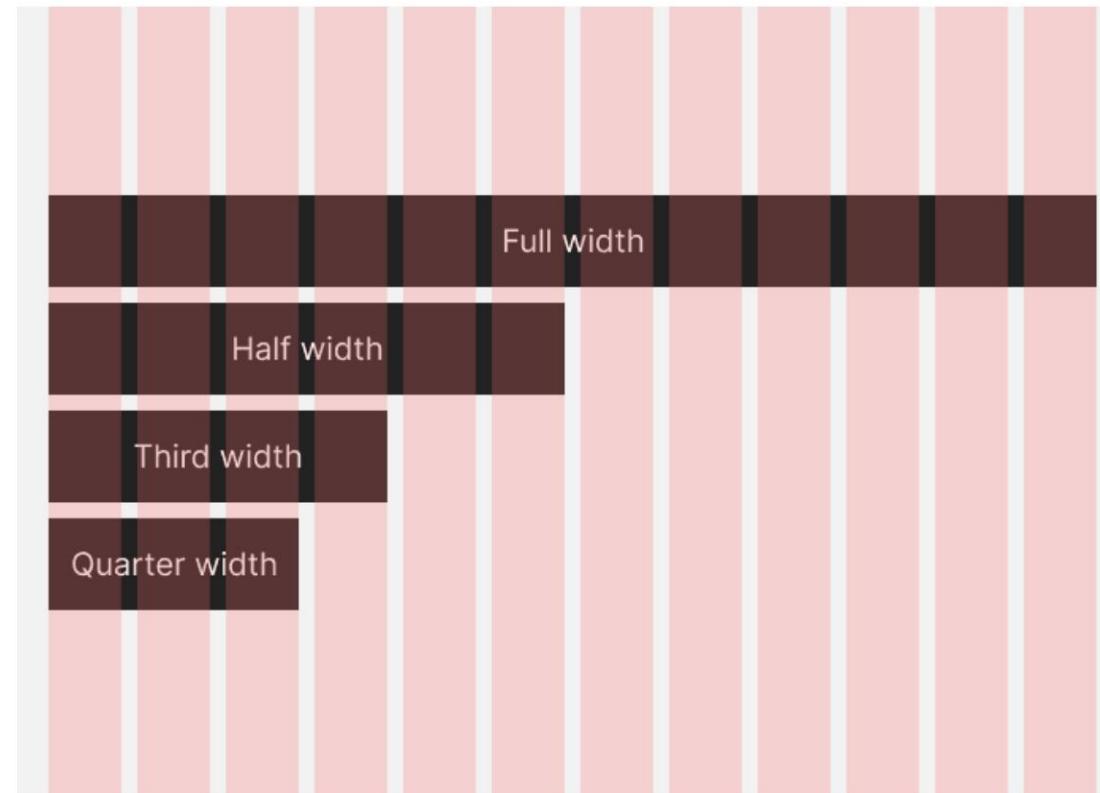
# Design recommendations

- Elements should be aligned



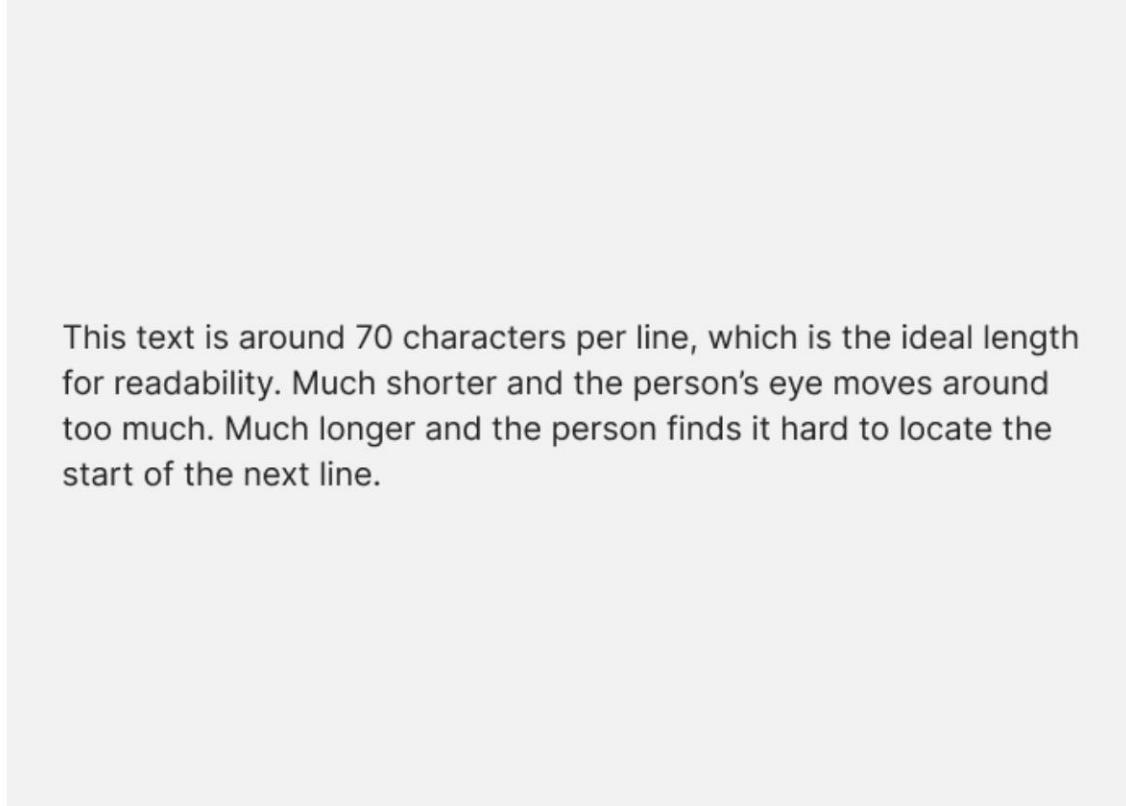
## Design recommendations

- Use 12 columns: It can be divided into columns of 1, 2, 3, 4 and 6.



## Design recommendations

- Use lines of about 70 characters to improve readability

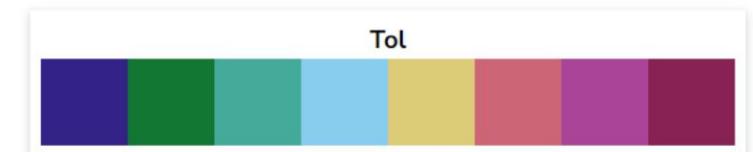
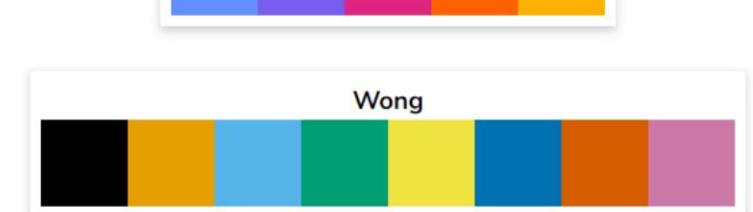


This text is around 70 characters per line, which is the ideal length for readability. Much shorter and the person's eye moves around too much. Much longer and the person finds it hard to locate the start of the next line.



# Design recommendations

- Use accessible color palettes: – More than 4% of the population is color blind, mainly men ([info.](#))
- <https://davidmathlogic.com/colorblind>
- <https://www.ibm.com/design/language/color/>
- <https://www.nature.com/articles/nmeth.1618>
- <https://personal.sron.nl/~pault/>



## Design recommendations

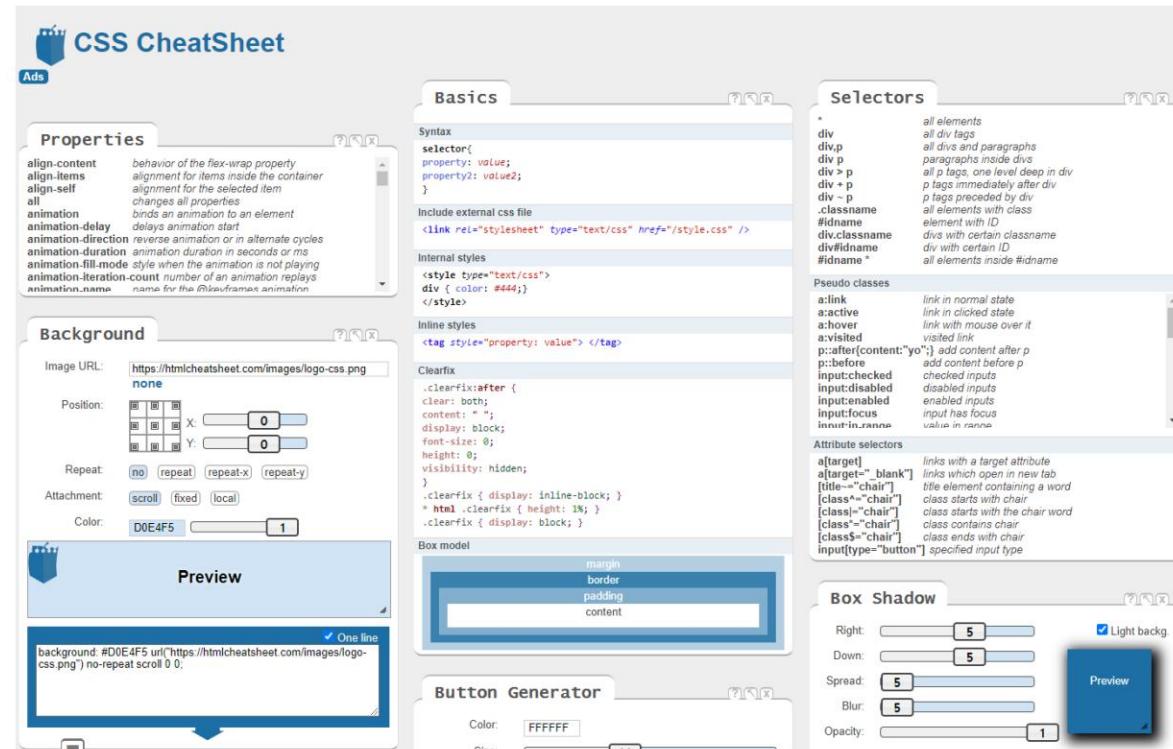
- And many more tips • <https://anthonyhobday.com/sideprojects/saferules/>



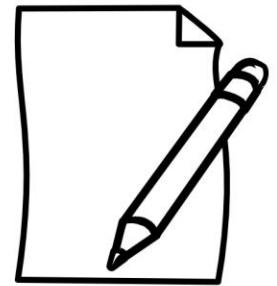
# Summary

- Reference: <https://www.w3schools.com/cssref/default.asp> •

Interactive CheatSheet: <https://htmlcheatsheet.com/css/>

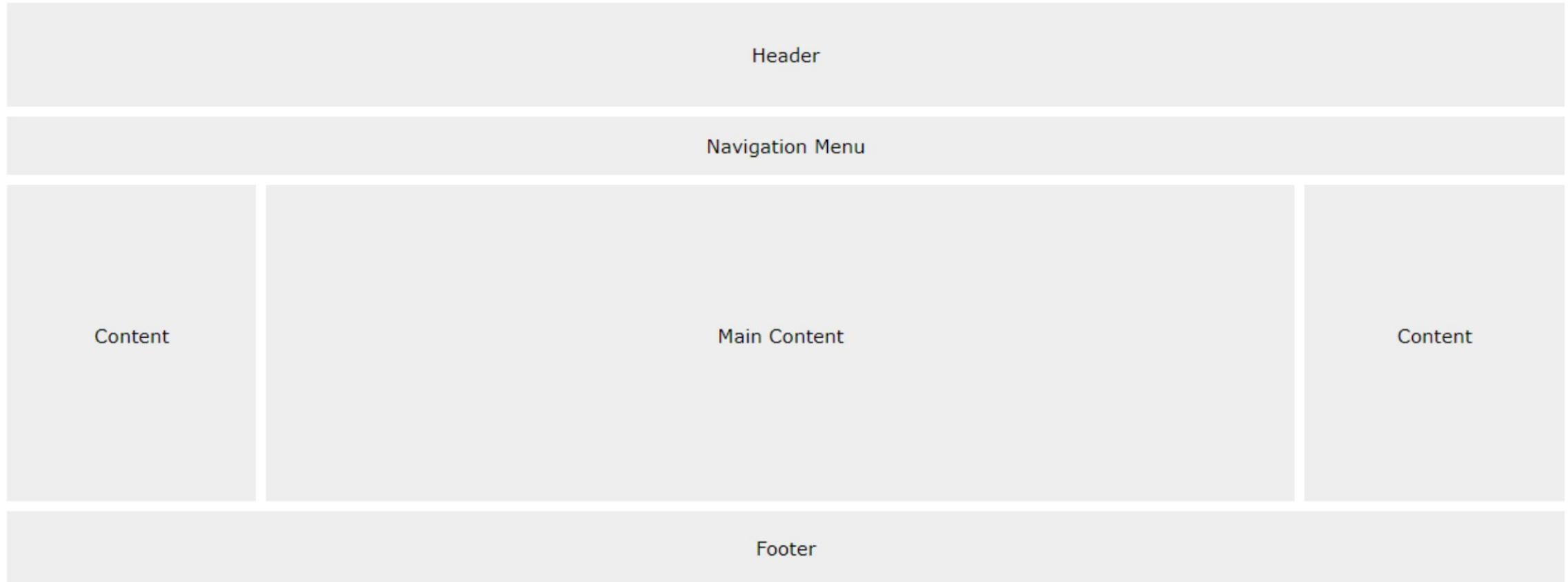


## Exercise 1



- Add CSS to the HTML project you started
- Add a header to the page
- Add a side menu
- Add varied styles, experiment with different selectors

# Example



## Example – Header

```
<div class="header">  
    <h1>CSS Example</h1>  
</div>  
  
.header  
{ background-color: #F1F1F1;  
text-align: center;  
padding: 30px;  
  
} .header h1  
{ font-size: 50px;  
}  
Ejemplo CSS
```



# Example – Menu

```
<div class="topnav">  
  <a href="#">Link</a>  
  <a href="#">Link</a>  
  <a href="#">Link</a>  
  <div class="right">  
    <a href="#">Link</a>  
  </div>  
</div>
```

```
.topnav {  
  overflow: hidden;  
  background-color: #333;  
}  
.topnav a {  
  float: left;  
  display: block;  
  color: #f2f2f2; text-align: center; padding: 14px 16px; text-decoration: none;  
}  
.right{ float:right;  
}
```

Link Link Link

Link



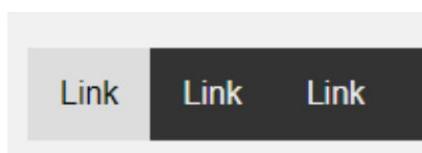
# Example – Menu

- Add hover effect .topnav a:hover {

```
background-color: #ddd;
```

```
color: black;
```

```
}
```



## Example – Columns

- Of equal size

```
<div class="column">  
  <h2>Title</h2>  
  <p>Lorem ipsum (...)</p>  
</div>  
  
<div class="column">  
  <h2>Title</h2>  
  <p>Lorem ipsum (...)</p>  
</div>
```

```
.column {  
  float: left;  
  width: 50%;  
}
```

Título	Título
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>	<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

# Example – Columns

- Different

```
<div class="column side">
  <h2>Title</h2>
  <p>Lorem ipsum (...)</p>
</div>
<div class="column middle">
  <h2>Title</h2>
  <p>Lorem ipsum (...)</p>
</div>
<div class="column side">
  <h2>Title</h2>
  <p>Lorem ipsum (...)</p>
</div>
```

```
.column { float: left; } .side{ width:
25%; } .middle{ width:
50%; }
```

Título	Título	Título
<p> Lorem ipsum dolor sit amet.</p> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>	<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>	<p> Lorem ipsum dolor sit amet.</p>

# Example – Columns

- Be careful with padding and margin!

```
<div class="column">  
  <h2>Title</h2>  
  <p>Lorem ipsum (...)</p>  
</div>  
  
<div class="column">  
  <h2>Title</h2>  
  <p>Lorem ipsum (...)</p>  
</div>
```

```
.column  
{ padding: 10px;  
float: left;  
width: 50%;  
}
```

## Título

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

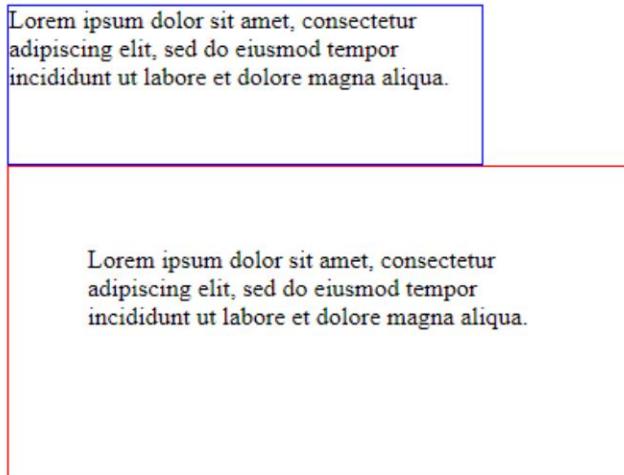
## Título

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Example – box-sizing

- By default, the height/width of a element does not include padding or margin

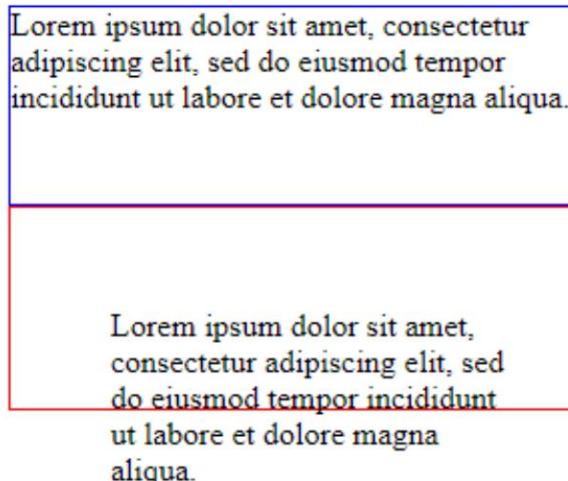
```
<div class="div1">Lorem (...)</div> <div  
class="div2">Lorem (...)</div>
```



```
.div1{ width:  
300px; height:  
100px; border: 1px solid blue;  
  
} .div2{ width:  
300px; height:  
100px; padding:  
50px; border: 1px solid red;  
}
```

## Example – box-sizing

- To avoid this we can use box sizing (more information) `<div class="div1">Lorem (...)</div> <div class="div2">Lorem (...)</div>`



```
* { box-sizing: border-box; }
.div1{
    width: 300px;
    height: 100px;
    border: 1px solid blue;
}

} .div2{
    width: 300px;
    height: 100px;
    padding: 50px;
    border: 1px solid red;
}
```

## Example – Footer

```
<div class="footer">  
  <h2>Footer</h2>  
</div>
```

```
.footer  
  { padding: 20px;  
    text-align: center;  
    background: #ddd;  
    margin-top: 20px;  
  }
```



**Footer**



# Example

- Link

## Ejemplo CSS

Link Link Link

**Title Heading**  
Title description  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

**Title Heading**  
Title description  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

**Title Heading**  
Title description  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

**About Me**  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit

**Popular Post**  
Image  
Image  
Image

**Follow Me**  
Some text..

**Footer**

# Responsive Web Design

- Make the website look good on all devices • Using only HTML and CSS
- Avoid horizontal scrolling



## Responsive Web Design – Viewport

- Use the following label so that the content is displayed well in mobiles:

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



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend ratione congue nihil imperdiet domino...

# Responsive Web Design – Media Queries

- Allows you to establish breakpoints in the CSS according to:

- Width and height of the device
  - Orientation
  - Resolution

- @media tag

```
@media not|only mediatype and (mediafeature and|or|not mediafeature) {  
    CSS-Code;  
}
```

- They are always defined at the end of the CSS



## Responsive Web Design – Media Queries

```
@media not|only mediatype and (mediafeature and|or|not mediafeature) {
```

CSS-Code:

```
}
```

- mediatype: normally it will be screen
- mediafeature: normally this will be max-width or min-width
- Different CSS can be linked:

```
<link rel="stylesheet" media="screen and (min-width: 900px)"  
      href="widescreen.css">
```

```
<link rel="stylesheet" media="screen and (max-width: 600px)"  
      href="smallscreen.css">
```



## Responsive Web Design – Media Queries

- Example with 4 columns: .column

```
{ float: left;  
  
width: 25%;  
  
padding: 20px;  
  
} @media screen and (max-width: 992px) { width:  
.column {      50%; }  
  
} @media screen and (max-width: 600px) { width:  
.column {      100%; }  
}
```



# Example

- Link

## Ejemplo CSS

Link Link Link

**Title Heading**  
Title description  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

**Title Heading**  
Title description  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

**Title Heading**  
Title description  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

**About Me**  
Image

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit

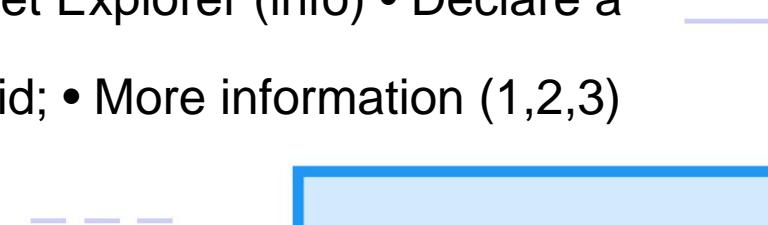
**Popular Post**  
Image  
Image  
Image

**Follow Me**  
Some text..

**Footer**

## Grid

- W3C Candidate Recommendation (not an official standard)
- Two dimensions (rows and columns)
- Divide the page into regions
- The evolution of table (which should not be used for layout)
- Allows a lot of versatility
- Not supported by Internet Explorer (info)
- Declare a container like: `display: grid;`
- More information (1,2,3)



# flexbox

- Flexible Box
  - [W3C Candidate Recommendation](#)
  - One dimension (rows or columns)
  - Facilitate the layout, alignment and distribution of elements • Alter the size of elements to fill the space • Not supported by Internet Explorer ([info](#)) • Declare a container like: `display: flex;` • More information ([1,2,3](#)) • Example
- — —
- 
- Change `flex-wrap: wrap;`

**Sample flexbox example**

**First article**

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry enniu pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

**Second article**

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry enniu pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

**Third article**

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry enniu pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat enniu. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed aqua chameleons put a bird on it. Listicle pickled man bun cornhole heliconia art party.

## sass

- Syntactically Awesome Style Sheets • CSS

Preprocessor • For working  
with large projects • Based on Ruby

- It allows:

- Variables
- Nesting
- Modules
- Mixins
- ...

- <https://sass-lang.com/>



## Less

- Leaner Style Sheets • CSS Preprocessor • Sass Inspired
- JavaScript Based
- It allows:
  - Variables
  - Nesting
  - Modules
  - ...
- <https://lesscss.org/>



# CSS FRAMEWORKS



# CSS Frameworks

- Characteristics:

- Bookstores
- They facilitate web development
- They make use of CSS
- They can have JS
- Reset the style sheet
- Grid layout (responsive)
- Mobile first
- Typography
- Sources
- Icons



## CSS Frameworks

- Disadvantages:
  - Learn to work with the framework
  - Lots of extra code that you won't use
  - Similarity of the websites that use it
  - Framework dependency



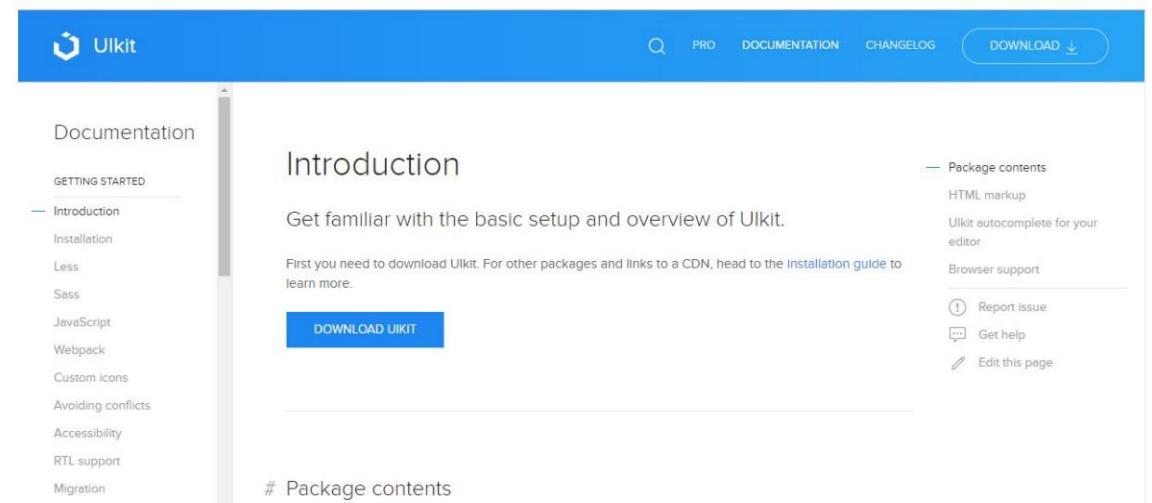
## CSS Frameworks - Examples



- Ulkit

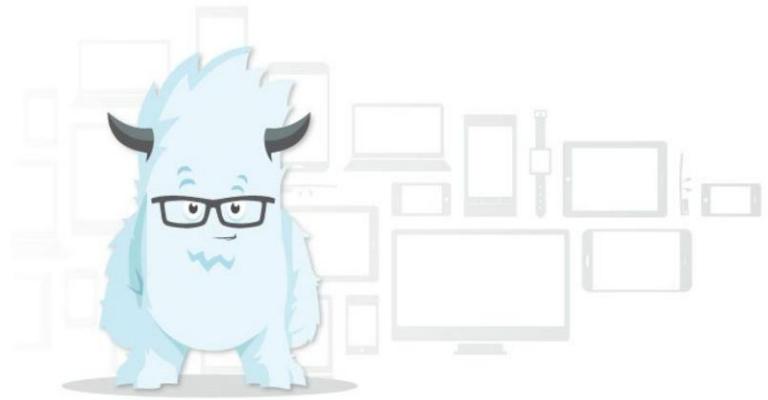
- Light
- Minimalist design
- Modular
- Self-contained (complicated to extend) –

<https://getuikit.com/>



The screenshot shows the UIkit documentation homepage. The header features the UIkit logo, a search bar, and navigation links for PRO, DOCUMENTATION, CHANGELOG, and DOWNLOAD. The main content area has a sidebar titled "Documentation" with sections for "GETTING STARTED" (Introduction, Installation, Less, Sass, JavaScript, Webpack, Custom icons, Avoiding conflicts, Accessibility, RTL support, Migration) and "Package contents" (HTML markup, UIkit autocomplete for your editor, Browser support). The main content area contains an "Introduction" section with text about basic setup and an "Installation" guide link, and a "DOWNLOAD UIKIT" button.

# CSS Frameworks - Examples



- Foundation

- “Responsive design gets a whole lot faster”
- Semantic
- Mobile First
- Customizable
- <https://get.foundation/>

**Foundation Framework**

Foundation for Sites  
6.8.1 changelog

Getting Started Expand

- Welcome
- Installation
- Starter Projects
- Compatibility
- Accessibility
- Kitchen Sink

Setup

- General
- Utilities
- Typography
- Controls
- Navigation
- Containers
- Media
- Plugins
- Sass

Find a page, component, variable, mixin, function...

Star

Showcase   Develop   Tutorials   Get Involved   Docs   Getting Started

**Welcome to Foundation 6**

We built Foundation for Sites to be the most advanced responsive front-end framework in the world.

**Welcome to the Foundation Docs**

ZURB

WELCOME TO FOUNDATION 6  
Installing Foundation  
New to Foundation for Sites?  
Experienced with Foundation for Sites?

**Installing Foundation**

There are a number of ways to install Foundation for Sites. Our installation page will help you find the best option for you.

**Install Foundation for Sites**

# CSS Frameworks - Examples



## • Bulma

- “The most advanced responsive front-end framework in the world”
- Responsive
- Modular
- <https://bulma.io/>

The screenshot shows the Bulma website's homepage. At the top, there is a navigation bar with the Bulma logo, links for 'Docs', 'Videos', 'Expo', and a 'Download' button. Below the navigation, there is a sidebar with sections for 'GUIDES' (Overview, Customize, Utilities) and 'CSS LIBRARY' (Columns, Elements, Components, Form, Layout, Helpers). The main content area features a section titled 'Documentation' with the sub-section 'Overview' highlighted. A description states: 'Everything you need to create a website with Bulma'. To the right, there is a 'My CSS ebook' section with a purple book thumbnail labeled 'CSS in 44 minutes'.

# CSS Frameworks - Examples



- **Materialize**

- “A modern responsive front-end framework”
- Based on Google Material Design ([link](#), [video](#))
- User Experience Focused
- Not updated since 2018
- <https://materializecss.com/>

**Getting Started**

Learn how to easily start using Materialize in your website.

---

**Download**

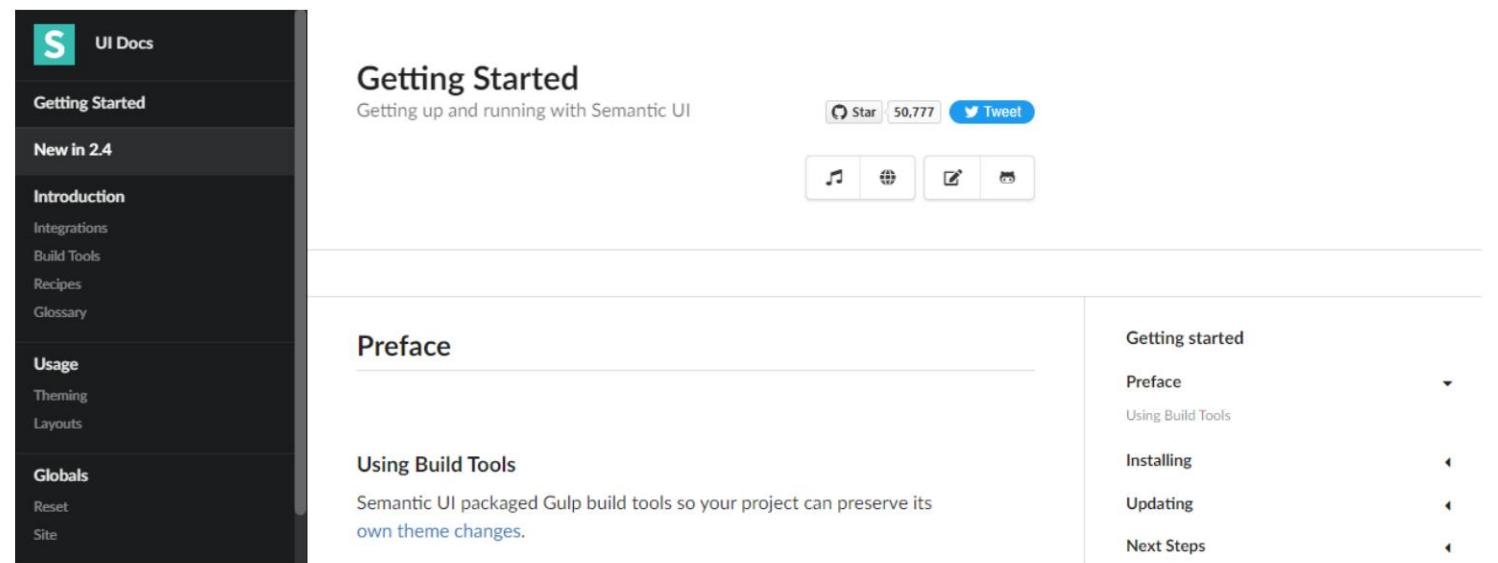
Materialize comes in two different forms. You can select which version you want depending on your preference and expertise. To start using Materialize, all you have to do is download one of the options below.

<a href="#">Materialize</a> <a href="#">Sass</a>	<a href="#">Download</a> <a href="#">Setup</a> <a href="#">Templates</a> <a href="#">Third-party Options</a> <a href="#">Sass</a>
---	---

## CSS Frameworks - Examples

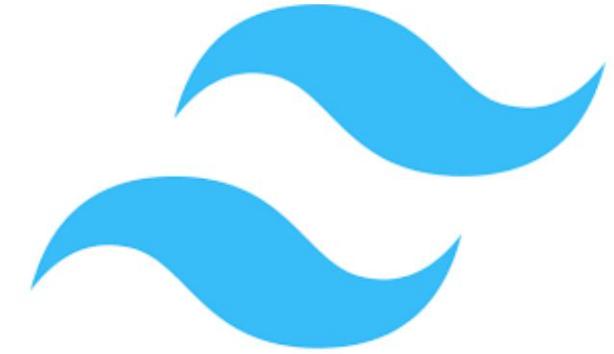


- Semantic UI
  - “User Interface is the language of the web”
  - Natural language
  - <https://semantic-ui.com/>



The screenshot shows the homepage of the Semantic UI documentation. The left sidebar has a dark theme with a teal header featuring a white 'S' icon and the text "UI Docs". The sidebar includes sections like "Getting Started", "New in 2.4", "Introduction" (with links to "Integrations", "Build Tools", "Recipes", and "Glossary"), "Usage" (with links to "Theming" and "Layouts"), and "Globals" (with links to "Reset" and "Site"). The main content area has a white background. It features a "Getting Started" section with the sub-headline "Getting up and running with Semantic UI", a "Star" button with 50,777 stars, a "Tweet" button, and social sharing icons. Below this is a "Preface" section. Further down is a "Using Build Tools" section with the text "Semantic UI packaged Gulp build tools so your project can preserve its own theme changes." On the right side, there's a sidebar with a teal header and a list of navigation items: "Getting started", "Preface", "Using Build Tools", "Installing", "Updating", and "Next Steps".

# CSS Frameworks - Examples



## • tailwindcss

- “A utility-first CSS framework for rapidly building custom user interfaces.”
- 
- Lightweight – Scalable
- <https://tailwindcss.com/>

The screenshot shows the Tailwind CSS website homepage. The header includes the Tailwind logo and version v3.3.3. The navigation bar has links for Docs, Components, Blog, and Showcase. Below the navigation is a search bar with placeholder "Quick search..." and a "Ctrl K" keybinding. A sidebar on the left contains links for Documentation, Components, Templates, Screencasts, Playground, Resources, and Community. Under "Getting Started", "Installation" is highlighted. The main content area features a large heading "Get started with Tailwind CSS". It explains that Tailwind CSS works by scanning HTML files, JavaScript components, and templates for class names, generating styles, and writing them to a static CSS file. It highlights the framework's speed, flexibility, and reliability with zero-runtime. At the bottom, there are links for Tailwind CLI, Using PostCSS, Framework Guides, and Play CDN.

# CSS Frameworks - Examples



- **Bootstrap**

- “Build fast, responsive sites”
- Rapid prototyping
- Great ecosystem
- It is managed by Twitter
- Many components
- Icons
- Less and Sass support
- [Market with official themes](#)
- <https://getbootstrap.com/>

The screenshot shows the official Bootstrap website. At the top is a purple header bar with the Bootstrap logo, navigation links for Docs, Examples, Icons, Themes, and Blog, a search bar, and social media links. The main content area has a light gray background. On the left is a sidebar with sections for 'Getting started' (including Introduction, Download, Contents, Browsers & devices, JavaScript, Webpack, Parcel, Vite, Accessibility, RFS, RTL, Contribute) and 'Customize' (Overview, Sass). The main content area features a large heading 'Get started with Bootstrap' with a 'View on GitHub' button. Below it is a paragraph about Bootstrap's power and versatility. A 'Quick start' section follows, with a note about creating an index.html file and a code snippet in a code editor.

# BOOTSTRAP



# Bootstrap

- Use

- Add the link to the CSS CDN within head
- Add meta viewport tag to make it responsive
- Optionally add the JS to the end of body

```
<head>  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link  
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"  
    integrity="sha384-T3c6Coli6uLrA9TneNEoa7RxnatjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"  
    crossorigin="anonymous">  
</head>
```



# Bootstrap

- Content

```
bootstrap/
  └── css/
      ├── bootstrap-grid.css
      ├── bootstrap-grid.css.map
      ├── bootstrap-grid.min.css
      ├── bootstrap-grid.min.css.map
      ├── bootstrap-reboot.css
      ├── bootstrap-reboot.css.map
      ├── bootstrap-reboot.min.css
      ├── bootstrap-reboot.min.css.map
      └── bootstrap.css
          ├── bootstrap.css.map
          └── bootstrap.min.css
              └── bootstrap.min.css.map
  └── js/
      ├── bootstrap.bundle.js
      ├── bootstrap.bundle.js.map
      ├── bootstrap.bundle.min.js
      ├── bootstrap.bundle.min.js.map
      ├── bootstrap.js
      ├── bootstrap.js.map
      ├── bootstrap.min.js
      └── bootstrap.min.js.map
```

css files	Layout	Content	Components	Utilities
bootstrap.css bootstrap.min.css	Included	Included	Included	Included
bootstrap-grid.css bootstrap-grid.min.css	<a href="#">Only grid system</a>	Not included	Not included	<a href="#">Only flex utilities</a>
bootstrap-utilities.css bootstrap-utilities.min.css	Not included	Not included	Not included	<a href="#">Included</a>
bootstrap-reboot.css bootstrap-reboot.min.css	Not included	<a href="#">Only Reboot</a>	Not included	Not included



# Bootstrap

- .min files
  - Reduced version of the original file –  
Unnecessary line breaks and spaces are removed –  
Used in production –  
bootstrap.css (200KB) → bootstrap.min.css (159KB)
- .map files –
  - Source maps
    - To work with development tools –  
Conversion that has been used to generate the minimized file (.min)
      - [More information](#)
- RTL (Right to Left) files: support for writing from right to left

# Bootstrap

- Containers

- Containers
- Most basic layout element
- Requirement to use Grid

```
<div class="container">  
    <!-- Content here -->  
</div>
```

	<b>extra small</b> <576px	<b>Small</b> ≥576px	<b>Medium</b> ≥768px	<b>Large</b> ≥992px	<b>X-Large</b> ≥1200px	<b>XX-Large</b> ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	720px	960px	1140px	1320px	
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

# Bootstrap – Grid

- Grid

- Flexbox based
- Predefined classes to work with columns

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```



# Bootstrap - Components

## ● Alerts

- Contextual message
- Result of a user action

```
<div class="alert alert-primary" role="alert"> Message </div> <div class="alert alert-secondary" role="alert"> Message </div> <div class="alert alert-success" role="alert"> Message </div> <div class="alert alert-danger" role="alert"> Message </div> <div class="alert alert-warning" role="alert"> Message </div> <div class="alert alert-info" role="alert"> Message </div> <div class="alert alert-light" role="alert"> Message </div> <div class="alert alert-dark" role="alert"> Message </div>
```

A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

A simple danger alert—check it out!

A simple warning alert—check it out!

A simple info alert—check it out!

A simple light alert—check it out!

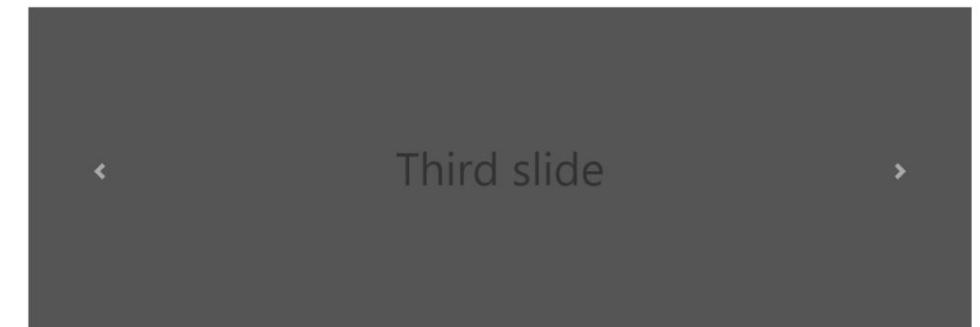
A simple dark alert—check it out!

# Bootstrap - Components

- Carousel

- Cycle elements (images or text)
- Should it be used?

```
<div id="carouselExampleSlidesOnly" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
</div>
```



## Bootstrap - Components

- Forms

- Forms

```
<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label> <input type="email"
      class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"> <div id="emailHelp" class="form-text">We'll
      never share your email with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Password</label> <input
      type="password" class="form-control" id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1"> <label
      class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Email address

We'll never share your email with anyone else.

Password

Check me out

**Submit**



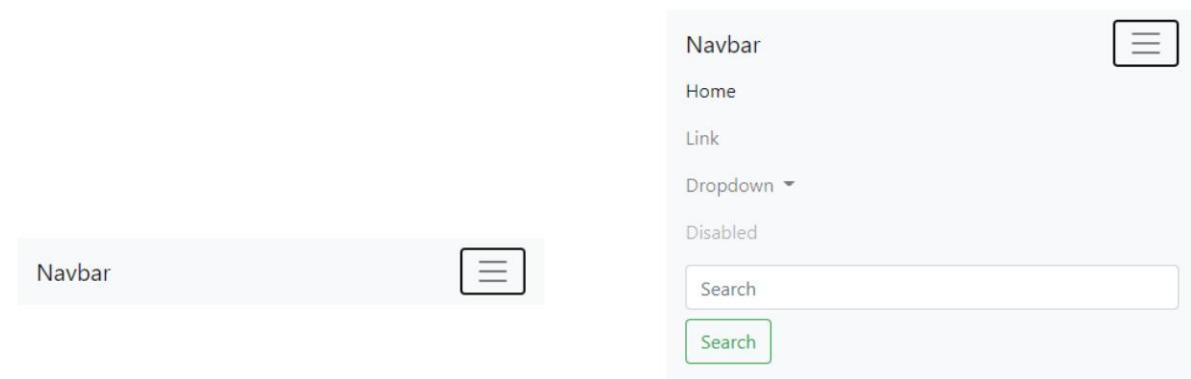
## Bootstrap - Components

- Navbar

- Navigation header
- Collapses with small screens and can be deployed

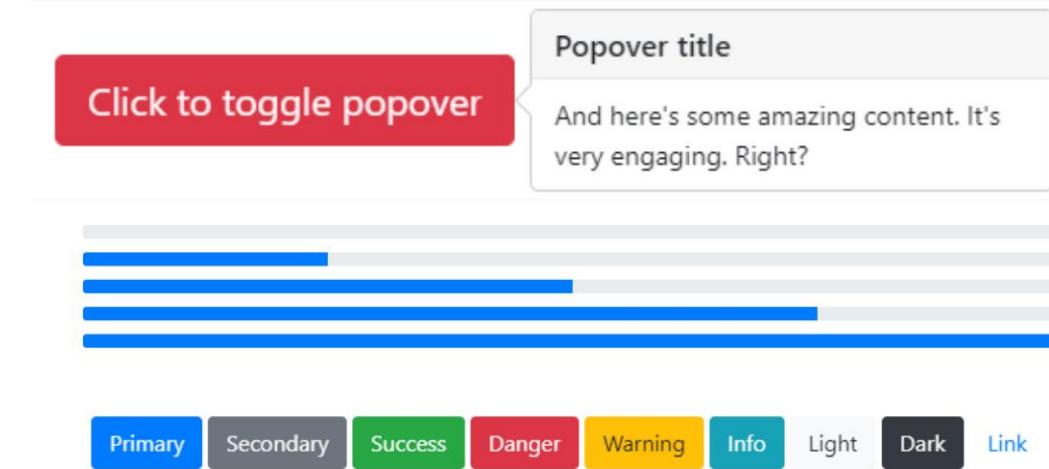
```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon">
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    (...) </div>
</nav>
```

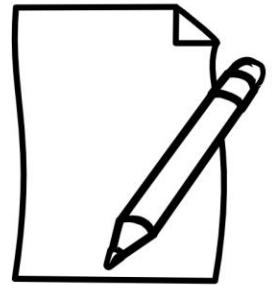


# Bootstrap - Components

- Popovers
- Progress bars
- Buttons
- Icons
- And much more...



## Exercise 2 – Bootstrap



- A new project • Starting from the Bootstrap template ([Quick start](#))
- Make a website
- Use varied components and experiment with the Grid

## References

- Luján Mora, S. (2002). Web application programming: history, basic principles and web clients. University Club Editorial.
- Martín, AR, & Martín, MJR (2014). Web applications. Paraninfo Editions, SA.
- Luján Mora, S. (2001). Internet programming: web clients. Editorial University Club.
- Luján Mora, S. (2002). Web application programming: history, basic principles and web clients. University Club Editorial.



# References

- HTML - Hypertext Markup Language – <https://www.w3.org/html/> – <https://www.w3.org/TR/html52/> – <https://www.w3.org/Protocols/Classic.html> • CSS

## Cascading Style Sheets

- <https://www.w3.org/Style/CSS/>
- <https://www.w3.org/Style/Examples/011/firstcss>
- <https://www.w3.org/Style/CSS/learning>

## • Tutorials

- <https://www.w3schools.com/html/>
- <https://www.w3schools.com/css/> •

## Other

- <https://github.com/andrews1022/web-development-2020-course-list> –
- <https://css-tricks.com/hottest-front-end-tools-in-2021/>

