



# Introducción

- CSS: Cascading Style Sheets (Hojas de estilo en cascada)
- Describe la presentación del documento
- Define layout (disposición), colores, fuentes, variaciones según el dispositivo...
- Permite que múltiples documentos utilicen el mismo estilo
- Cascading: prioridad para ver qué regla se aplica

# Introducción

- Lo gestiona el World Wide Web Consortium (W3C)
- CSS1: Primera versión en 1996
- CSS3: Versión más actual, modular
- En realidad las versiones de CSS se conocen como niveles (Levels)
- Se va actualizando periódicamente
- Se hacen [snapshots anuales](#)
- [CSS Specifications](#)

# Introducción

- Cómo insertar CSS – Archivo externo

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <link rel="stylesheet" href="mystyle.css">
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

Archivo mystyle.css:

```
body {  
  background-color: lightblue;  
}  
h1 {  
  color: navy;  
  margin-left: 20px;  
}
```

# Introducción

- Cómo insertar CSS – Interno

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

# Introducción

- Cómo insertar CSS – En línea

```
<!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>
```

# ¿Cuándo usar cada forma?



# Introducción

- HTML nunca estuvo pensado para incluir etiquetas para dar formato
- HTML sirve para describir el contenido
- HTML 3.2 introdujo etiquetas como <font> y empezó la pesadilla
  - A raíz de esto la W3C creó CSS
- Nunca se debería insertar CSS en línea



# Introducción

- Comentarios

- Sintaxis: `/* ... */`
  - Pueden ser multilinea
  - Se pueden usar en el HTML dentro de la etiqueta `<style></style>`
- ```
p {  
    color: red;  /* Set text color to red */  
}  
/* This is  
a multi-line  
comment */
```

# Introducción

- CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos.

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

- `h1` es el **selector**.
  - Indica los elementos del HTML que van a ser afectados.
- `{color: red;}` es el bloque de **declaración**.
  - Cada bloque contiene una o más declaraciones separadas por ;
  - `color` es el nombre de una propiedad
  - `red` es el valor

# Selectores

- Simples
- Combinación
- Pseudo-clases
- Pseudo-elementos
- Atributos

# Selectores simples

- Basados en el nombre de un elemento (p, a, body, div...)
- Sintaxis: nombre\_elemento {}

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

# Selectores simples

- Basados en el atributo id de un elemento (que tiene que ser único)
- Sintaxis: #nombre\_id {}

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

-----

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

# Selectores simples

- Basados en el atributo class de un elemento (puede estar repetido)
- Sintaxis: `.nombre_class {}`

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

-----

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

# Selectores simples

- Basados en el atributo class de un elemento (puede estar repetido)
- Se puede especificar que solo afecte a un tipo determinado de elementos
- Sintaxis: `nombre_elemento.nombre_class { }`

```
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>
```

# Selectores simples

- Selector universal
- Selecciona todos los elementos del HTML
- Sintaxis: \* { }

```
* {  
  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>
```



# Selectores simples

- Agrupación
- Sintaxis: `nombre_selector1, nombre_selector2... { }`

```
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;  
}
```

# Selectores combinación

- CSS Combinators
- Relacionan selectores
- Tipos:
  - Descendiente
  - Hijo
  - Hermano adyacente
  - Hermano general

# Selectores combinación – Descendiente

- Selecciona todos los elementos descendientes del elemento especificado
- Sintaxis: `selector_simple1 selector_simple2 {}`

```
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>
```

# Selectores combinación – Hijo

- Selecciona todos los elementos que son hijos del elemento especificado
- Sintaxis: `selector_simple1 > selector_simple2 {}`

```
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>
```

# Selectores combinación – Hermano adyacente

- Selecciona todos los elementos que son hermanos (tienen el mismo padre) y que estén inmediatamente a continuación
- Sintaxis: `selector_simple1 + selector_simple2 {}`

```
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>
```

# Selectores combinación – Hermano general

- General sibling
- Selecciona todos los elementos que son hermanos y aparezcan después
- Sintaxis: `selector_simple1 ~ selector_simple2 { }`

```
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>
```

# Selectores pseudoclasses

- Definen un estado especial de un elemento
- [Lista con las pseudoclasses](#)
- Sintaxis: `selector:pseudo-class { }`

`:hover` → Selecciona al elemento que tiene el ratón encima

`:focus` → Selecciona al elemento que tiene el foco

`:first-child` → Selecciona al primer hijo

`:last-child` → Selecciona al último hijo

# Selectores pseudoelementos

- Identifican una parte especial de un elemento
- Sintaxis: `selector::pseudo-element { }`

`::after` → Para insertar algo antes. Usar con la propiedad `content`

`::before` → Para insertar algo después. Usar con `content`

`::first-letter` → Selecciona la primera letra

`::first-line` → Selecciona la primera línea

`::selection` → La selección del usuario



# Selectores de atributo

- Selecciona un elemento con un atributo concreto
- Puede estar asociado a un selector
- Sintaxis: *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>
```

# Selectores de atributo con valor

- Selecciona un elemento con un atributo y valor concretos
- Puede estar asociado a un selector
- Sintaxis: *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>
```

# Selectores de atributo – Otros

`[attribute~="value"]`

- Selecciona un elemento con un atributo cuyo valor contiene una palabra específica

`[attribute|="value"]`

- Selecciona un elemento con un atributo cuyo valor empieza por una palabra concreta

`[attribute^=value]`

- Selecciona un elemento con un atributo cuyo valor empieza de una forma concreta

`[attribute$="value"]`

- Selecciona un elemento con un atributo cuyo valor termina de una forma concreta

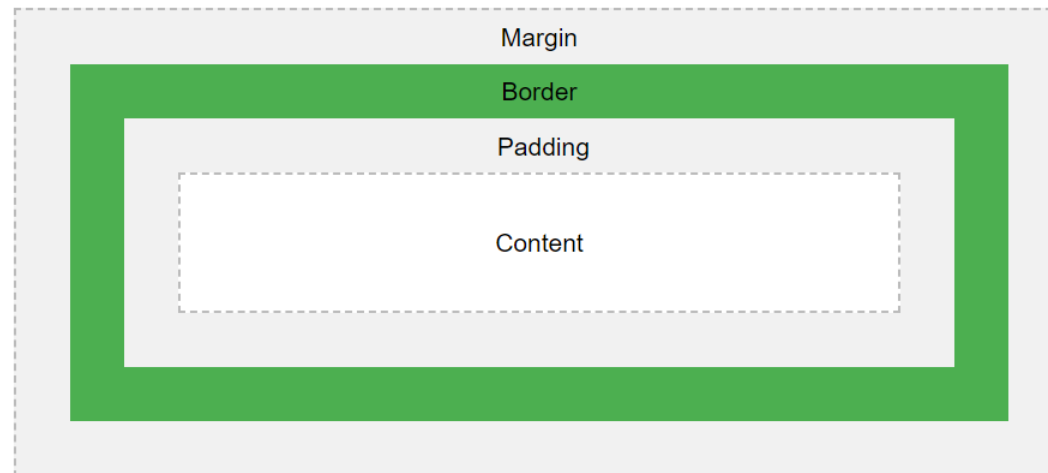
`[attribute*="value"]`

- Selecciona un elemento con un atributo cuyo valor contiene un valor específico

# Declaraciones

## Modelo de caja

- **Contenido:** El contenido de la caja como texto o imágenes
- **Padding:** Área vacía alrededor del contenido. Es transparente
- **Borde:** Borde alrededor del padding y del contenido
- **Margen:** Área vacía fuera del borde. Es transparente



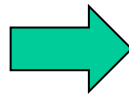
# Declaraciones – margin

- Espacio alrededor del elemento por fuera
- margin-top, margin-right, margin-bottom, margin-left
- Valores (pueden ser negativos):
  - auto
  - Longitud (px, pt, cm...)
  - % con respecto al ancho del elemento
  - inherit: heredado del padre
- [Más información](#)

# Declaraciones – margin

- Se puede colapsar
- Sintaxis:
  - `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;  
}
```

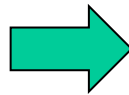
# Declaraciones – padding

- Espacio alrededor del contenido del elemento y dentro de los bordes
- padding-top
- padding-right
- padding-bottom
- padding-left
- Valores (no pueden ser negativos):
  - Longitud (px, pt, cm...)
  - % con respecto al ancho del elemento
  - inherit: heredado del padre
- [Más información](#)

# Declaraciones – padding

- Se puede colapsar
- Sintaxis:
  - 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;  
}
```



# Declaraciones

## width (ancho) y height (altura)

- Ancho y alto del contenido
- Luego además habrá que sumar el padding, el borde y el margen
- Valores:
  - auto
  - Longitud (px, pt, cm...)
  - % con respecto al ancho del bloque contenedor
  - initial
  - inherit: heredado del padre

# Declaraciones

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

- Máximo/mínimo ancho y alto del contenido
- Valores:
  - Longitud (px, pt, cm...)
  - % con respecto al ancho del bloque contenedor
  - none: no hay máximo/mínimo

# Declaraciones – text

- color
- background-color
- text-align: left/right/center/justify;
- direction
- vertical-align
- text-decoration
- text-transform: uppercase/lowercase/capitalize;
- text-indent: Sangrado de la primera línea

# Declaraciones – text

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

[Más información](#)

# Declaraciones – 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;`

[Más información](#)

# Declaraciones – position

- Tipo de posicionamiento usado para el elemento
- Valores:
  - static
    - Colocado según el flujo normal de la página. Por defecto.
  - relative
    - Relativo a su posición por defecto.
  - fixed
    - Relativo al viewport, no se mueve al hacer scroll.
  - absolute
    - Relativo al ancestro más cercano

# Declaraciones

- Valores (cont.):
  - sticky
    - Se queda “pegado” al hacer scroll
- Para posicionar usar además:
  - top
  - right
  - bottom
  - left
  - z-index

[Más información](#)

# Valores – Unidades

- Absolutas
  - cm
  - mm
  - in: pulgadas( $1\text{ in} = 2.54\text{ cm}$ )
  - px: pixels, relativos al dispositivo
  - pt: puntos ( $1\text{ pt} = 1/72\text{ in}$ )



# Valores – Unidades

- Relativas
  - em: relativo al tamaño de fuente. 2em es dos veces el tamaño de fuente actual
  - rem: relativo al tamaño de fuente del elemento raíz. Suele equivaler a 16px.
  - ch: relativo al tamaño del cero
  - %: relativo al elemento padre

[Más información](#)

# Valores – Colores

- Nombres: [Lista de los nombres de colores](#)
- [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)`

[Más información](#)

# Varios

- Bordes redondeados
- Gradientes
- Transiciones
- Tooltips
- Múltiples columnas
- Media queries (responsive)

# Specificity

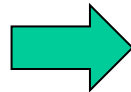
- Resolución de conflictos
- Cuanto más específico un elemento, mayor prioridad
- id > class > element
- En caso de igualdad, el que aparezca después en el archivo

[Más información](#)

# !important

- Añadir más importancia a una propiedad o valor
- Sobreescribe todas las reglas de esa propiedad

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



```
<p id="myid" class="myclass">  
    Texto de prueba  
</p>
```

# Resumen

- Referencia: <https://www.w3schools.com/cssref/default.asp>
- Interactive CheatSheet: <https://htmlcheatsheet.com/css/>

The screenshot displays the CSS CheatSheet website, which is organized into several sections:

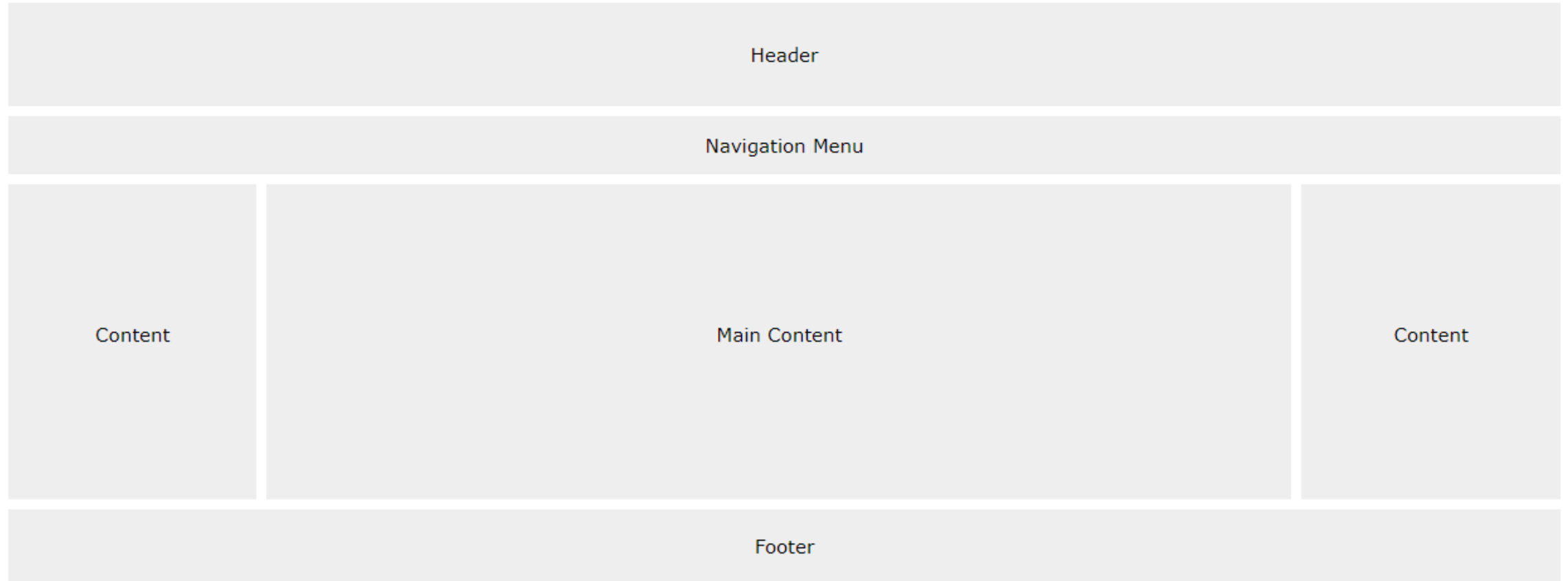
- Properties:** Lists various CSS properties with their descriptions. For example, `align-content` is described as "behavior of the flex-wrap property alignment for items inside the container".
- Background:** Includes fields for Image URL, Position, Repeat, Attachment, and Color, along with a Preview window.
- Basics:** Contains sections for Syntax, Internal styles, and Inline styles, providing examples of how to use CSS in HTML.
- Selectors:** Lists various CSS selectors, including pseudo-classes and attribute selectors, with their respective descriptions.
- Box Shadow:** Includes sliders for Right, Down, Spread, Blur, and Opacity, along with a checkbox for "Light backg" and a Preview window.
- Button Generator:** Includes fields for Color and Size, with a Preview window.

# Ejercicio



- Añade CSS al proyecto que empezaste de HTML
- Añade una cabecera a la página
- Añade un menú lateral
- Añade estilos variados, experimenta con diferentes selectores

# Ejemplo





# Ejemplo – Header

```
<div class="header">  
  <h1>Ejemplo CSS</h1>  
</div>
```

```
.header {  
  background-color: #F1F1F1;  
  text-align: center;  
  padding: 30px;  
}  
.header h1 {  
  font-size: 50px;  
}
```

**Ejemplo CSS**

# Ejemplo – Menú

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

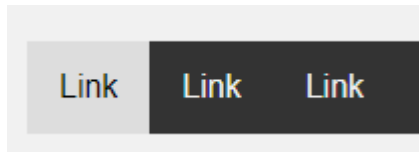
```
.topnav {
  overflow: hidden;
  background-color: #333;
}
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
.right{
  float:right;
}
```



# Ejemplo – Menú

- Añadir efecto al pasar el ratón por encima

```
.topnav a:hover {  
    background-color: #ddd;  
    color: black;  
}
```



# Ejemplo – Columnas

- De igual tamaño

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

```
.column {
  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.

# Ejemplo – Columnas

- Diferentes

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

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

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

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

Título	Título	Título
Lorem ipsum dolor sit amet.	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.	Lorem ipsum dolor sit amet.

# Ejemplo – Columnas

- ¡Cuidado con el padding y el margin!

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

```
<div class="column">  
  <h2>Título</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.

# Ejemplo – box-sizing

- Por defecto, el alto/ancho de un elemento no incluye el padding ni el margin

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

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

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

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

# Ejemplo – box-sizing

- Para evitarlo podemos usar box-sizing ([más información](#))

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

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

Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed  
do eiusmod tempor incididunt  
ut labore et dolore magna  
aliqua.

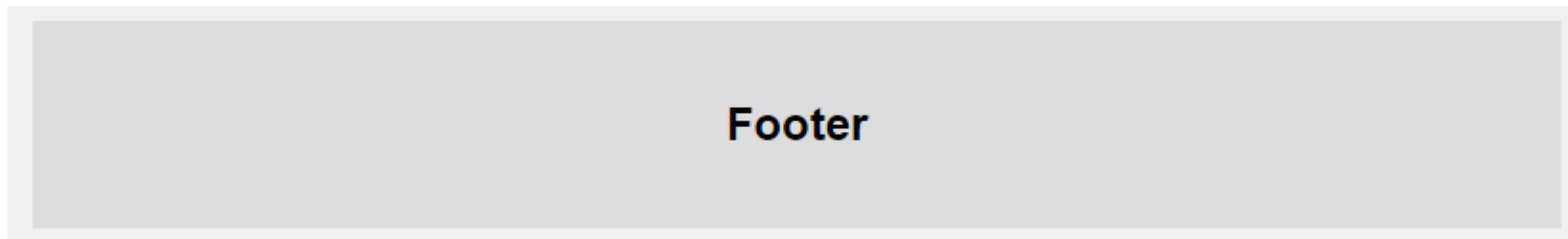
```
* {  
    box-sizing: border-box;  
}  
  
.div1{  
    width: 300px;  
    height: 100px;  
    border: 1px solid blue;  
}  
  
.div2{  
    width: 300px;  
    height: 100px;  
    padding: 50px;  
    border: 1px solid red;  
}
```



# Ejemplo – Footer

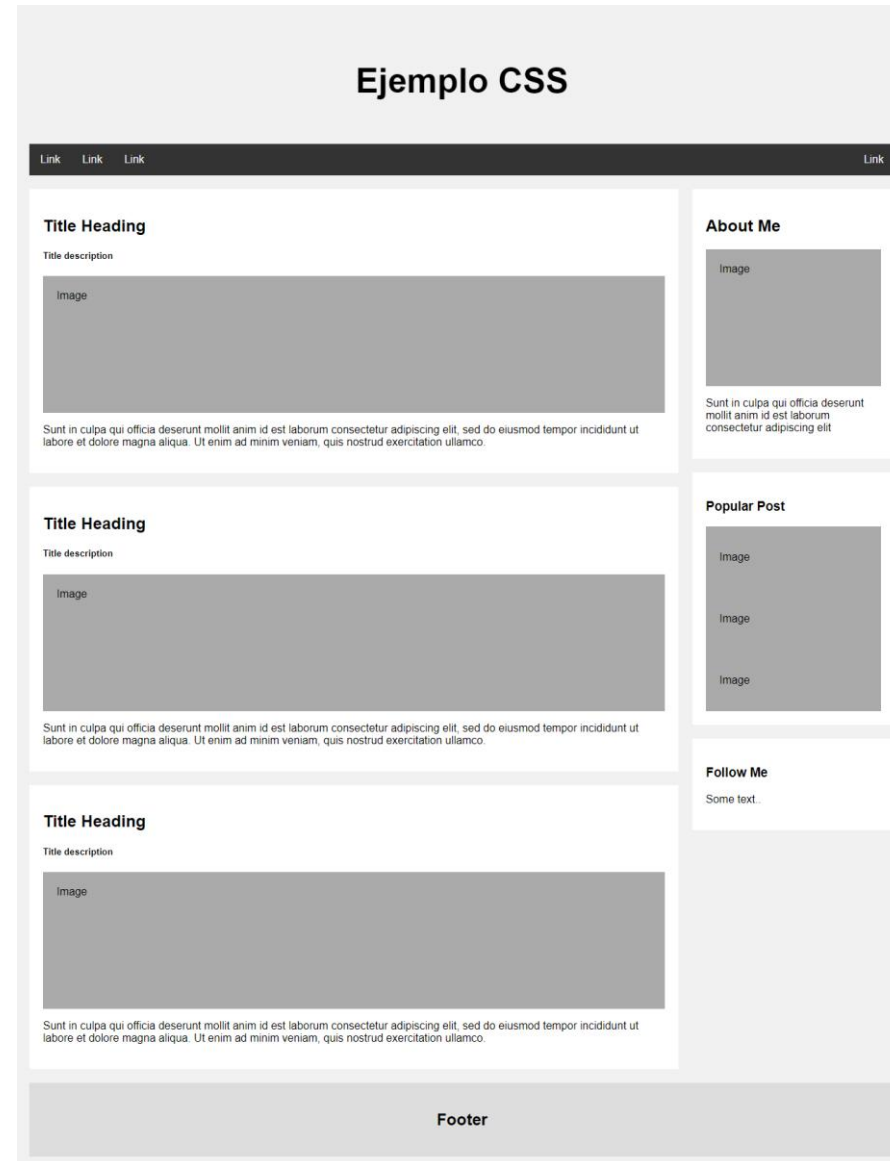
```
<div class="footer">  
  <h2>Footer</h2>  
</div>
```

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



# Ejemplo

- [Link](#)



# Responsive Web Design

- Que la página web se vea bien en todos los dispositivos
- Haciendo uso solo de HTML y CSS
- Evitar scroll horizontal



# Responsive Web Design – Viewport

- Usar la siguiente etiqueta para que se visualice bien el contenido en móviles:

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



Lorem ipsum dolor sit amet, consectetur 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 orion conase nihil imperdiet domine...

# Responsive Web Design – Media Queries

- Permite establecer breakpoints en el CSS según:
  - Ancho y alto del dispositivo
  - Orientación
  - Resolución
- Etiqueta @media

```
@media not|only mediatype and (mediafeature and|or|not mediafeature) {  
    CSS-Code;  
}
```
- Siempre se definen al final del CSS

# Responsive Web Design – Media Queries

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

- mediatype: normalmente será screen
- mediafeature: normalmente será max-width o min-width
- Se pueden enlazar diferentes CSS:

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

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

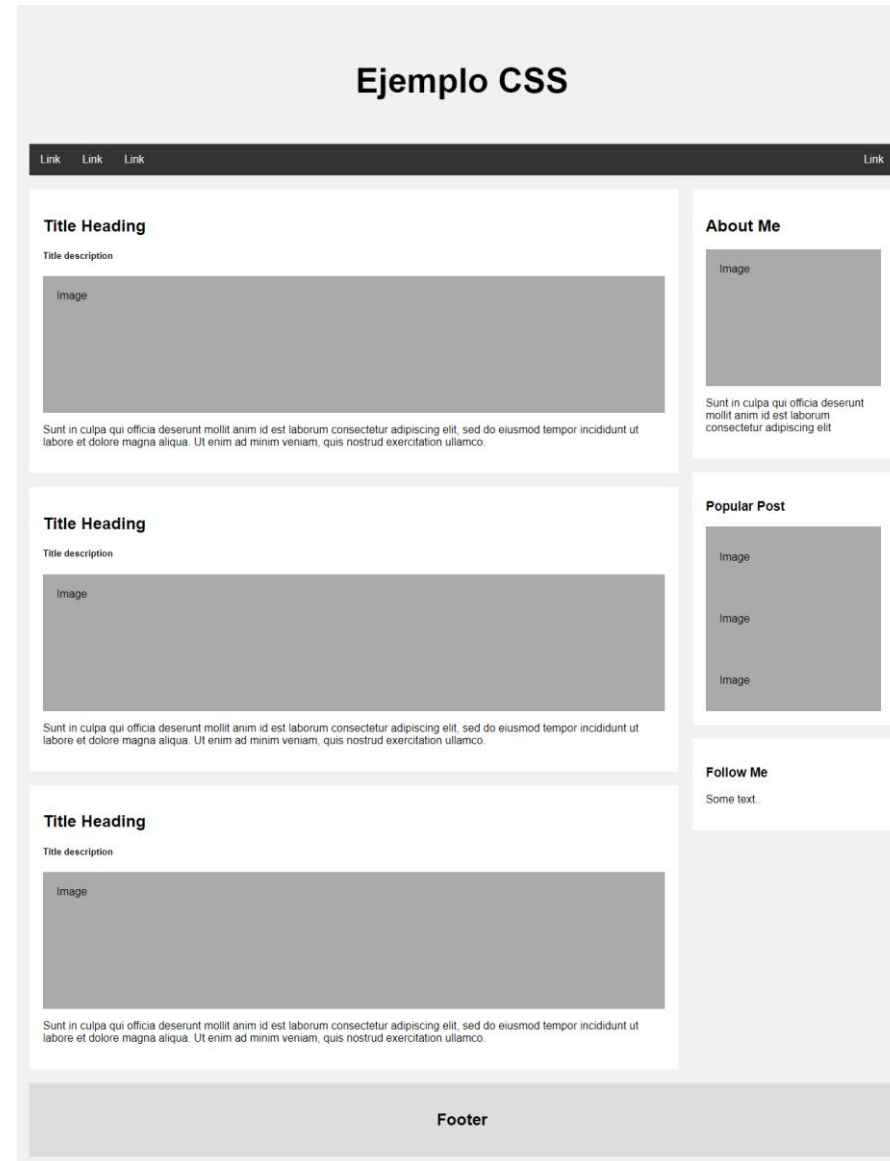
# Responsive Web Design – Media Queries

- [Ejemplo con 4 columnas:](#)

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

# Ejemplo

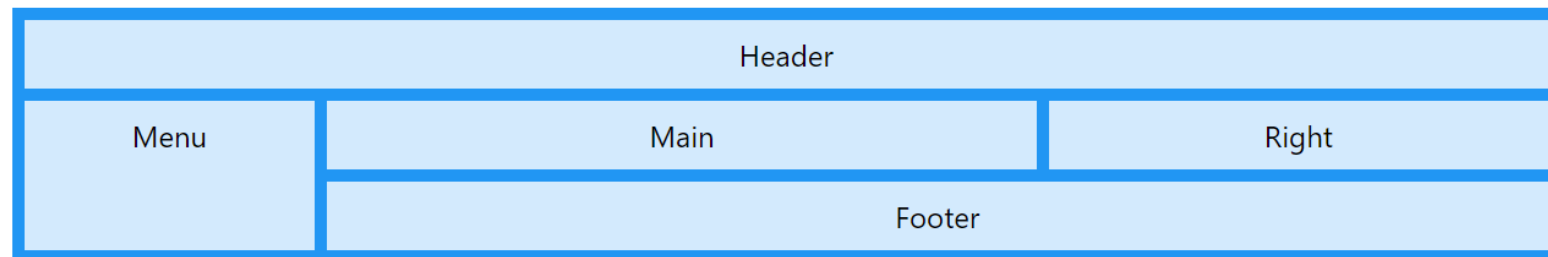
- [Link](#)





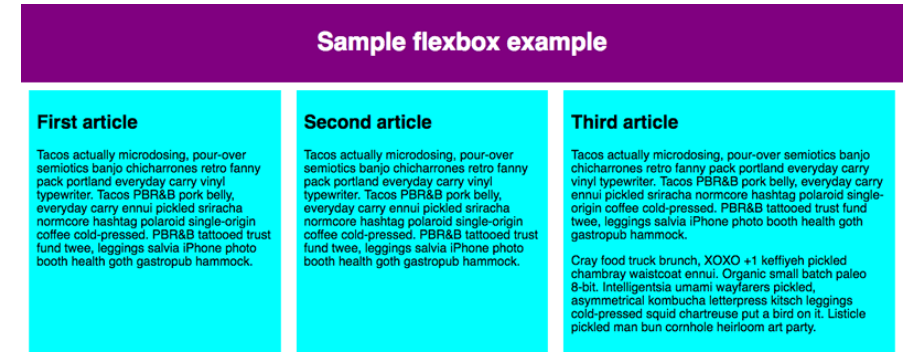
# Grid

- [W3C Candidate Recommendation](#) (no es un estandar oficial)
- Dos dimensiones (filas y columnas)
- Dividir la página en regions
- La evolución de la table (que no debería usarse para el layout)
- No está soportado por Internet Explorer ([info](#))
- Declarar un contenedor como: `display: grid;`
- Más información ([1](#),[2](#),[3](#))



# Flexbox

- Flexible Box
- [W3C Candidate Recommendation](#)
- Una dimensión (filas o columnas)
- Facilitar el layout, alineamiento y distribución de elementos
- Alterar el tamaño de los elementos para rellenar el espacio
- No está soportado por Internet Explorer ([info](#))
- Declarar un contenedor como: `display: flex;`
- Más información ([1](#),[2](#),[3](#))
- [Ejemplo](#)
  - Cambiar `flex-wrap: wrap;`



# Sass

- Syntactically Awesome Style Sheets
- Preprocesador CSS
- Para trabajar con proyectos grandes
- Basado en Ruby
- Permite:
  - Variables
  - Nesting
  - Módulos
  - Mixins
  - ...
- <https://sass-lang.com/>



# Less

- Leaner Style Sheets
- Preprocesador CSS
- Inspirado en Sass
- Basado en JavaScript
- Permite:
  - Variables
  - Nesting
  - Módulos
  - ...
- <https://lesscss.org/>



# CSS FRAMEWORKS

# CSS Frameworks

- Características:
  - Librerías
  - Facilitan el desarrollo web
  - Hacen uso de CSS
  - Pueden tener JS
  - Resetean la hoja de estilos
  - Layout en forma de grid (responsive)
  - Mobile first
  - Tipografía
  - Fuentes
  - Iconos

# CSS Frameworks

- Desventajas:
  - Aprender a trabajar con el framework
  - Mucho código extra que no usaras
  - Similitud de las webs que lo usen
  - Dependencia del framework

# CSS Frameworks - Ejemplos

- **UIKit**
  - Ligero
  - Diseño minimalista
  - Modular
  - Self-contained (complicado de extender)
  - 16K+ stars on GitHub
  - <https://getuikit.com/>





# CSS Frameworks - Ejemplos

- **Foundation**

- “Responsive design gets a whole lot faster”
- Semantic
- Mobile First
- Customizable
- 20K+ stars in GitHub
- <https://get.foundation/>



# CSS Frameworks - Ejemplos

- **Bulma**

- “The most advanced responsive front-end framework in the world”
- Responsive
- Modular
- 40K+ stars in GitHub
- <https://bulma.io/>



# CSS Frameworks - Ejemplos

- **Materialize**

- “A modern responsive front-end framework”
- Basado en Material Design de Google (link, [video](#))
- User Experience Focused
- No se actualiza desde 2018
- 38K+ stars in GitHub
- <https://materializecss.com/>



# CSS Frameworks - Ejemplos

- **Semantic UI**
  - “User Interface is the language of the web”
  - Lenguaje natural
  - No se actualiza desde 2018
  - 48K+ stars in GitHub
  - <https://semantic-ui.com/>



# CSS Frameworks - Ejemplos

- **Bootstrap**
  - “Build fast, responsive sites”
  - Prototipado rápido
  - Gran ecosistema
  - Lo gestiona Twitter
  - Muchos componentes
  - Iconos
  - Soporte Less y Sass
  - [Mercado con temas oficiales](#)
  - 150K+ stars in GitHub
  - <https://getbootstrap.com/>



# BOOTSTRAP

# Bootstrap

- Uso
  - Añadir dentro de head el enlace al CDN del CSS
  - Añadir etiqueta meta viewport para que sea responsive
  - Opcionalmente añadir los JS al final de body

<head>

```
<meta name="viewport" content="width=device-width, initial-scale=1>
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css"
```

```
rel="stylesheet" integrity="sha384-
```

```
F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU"
```

```
crossorigin="anonymous">
```

```
</head>
```

# Bootstrap

- Contenido

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	Included
bootstrap-reboot.css bootstrap-reboot.min.css	Not included	<a href="#">Only Reboot</a>	Not included	Not included



# Bootstrap

- Archivos .min
  - Versión reducida del archivo original
  - Se eliminan saltos de línea y espacios innecesarios
  - Se usa en producción
  - bootstrap.css (200KB) → bootstrap.min.css (159KB)
- Archivos .map
  - Source maps
  - Para trabajar con herramientas de desarrollo
  - Conversión que se ha usado para generar el archivo minimizado (.min)
  - [Más información](#)
- Archivos rtl (Right to Left): soporte para escritura de derecha a izquierda

# Bootstrap

- Containers

- Contenedores
- Elemento de layout más básico
- Requisito para usar Grid

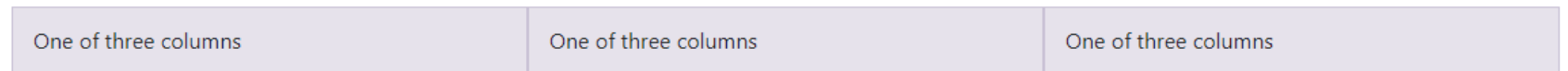
```
<div class="container">  
  <!-- Content here -->  
</div>
```

	<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%	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
  - Basado en Flexbox
  - Clases predefinidas para trabajar con columnas

```
<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
  - Mensaje contextual
  - Resultado de una acción del usuario

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

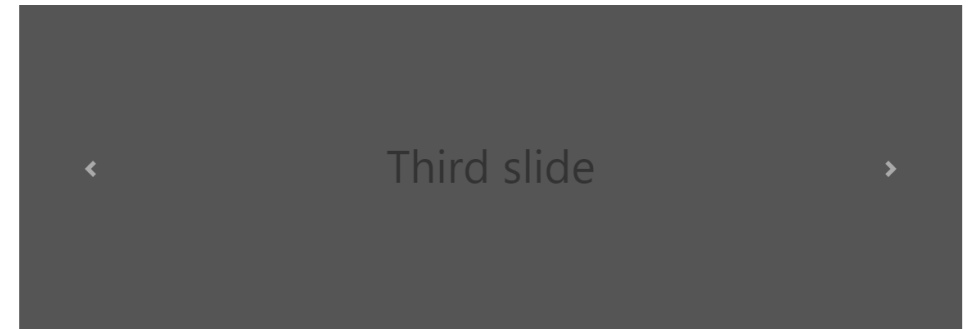
A simple light alert—check it out!

A simple dark alert—check it out!

# Bootstrap - Components

- Carousel
  - Ciclar elementos (imágenes o texto)
  - ¿Se debería usar?

```
<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>
```



# Bootstrap - Components

- Forms
  - Formularios

```
<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

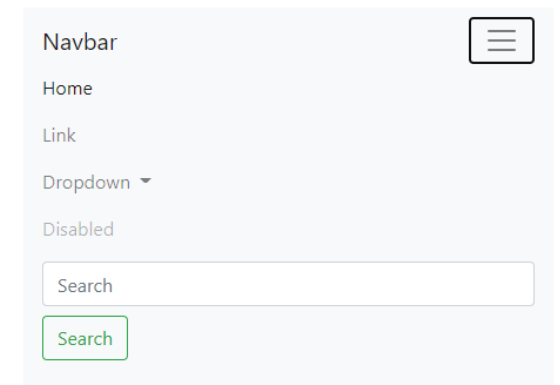
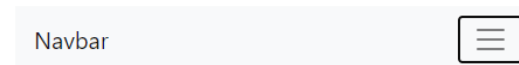
- Navbar

- Cabecera de navegación
- Colapsa con pantallas pequeñas y se puede desplegar



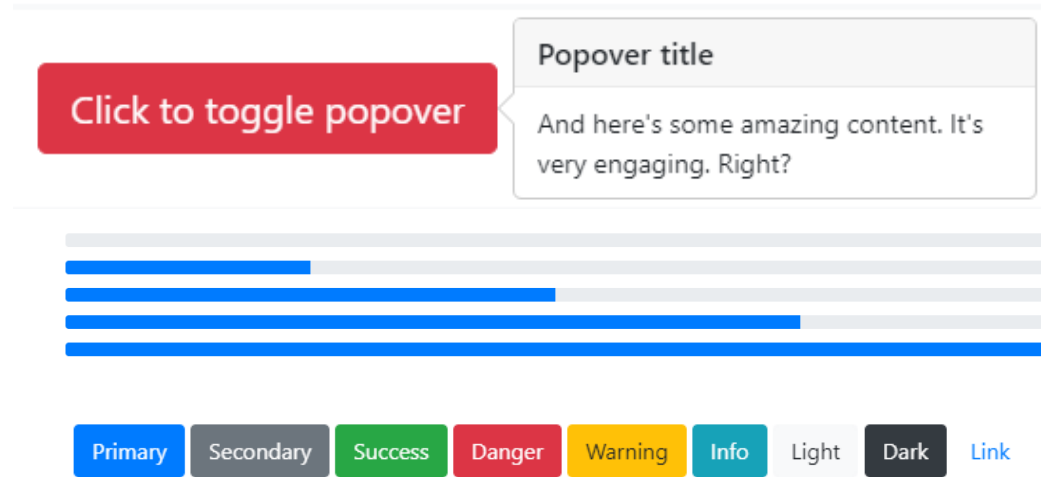
```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">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

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



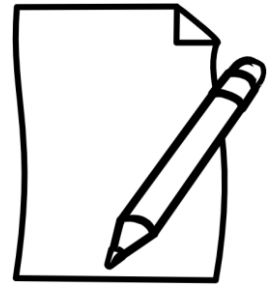
# Bootstrap - Components

- Popovers
- Barras de progreso
- Botones
- Iconos
- Y mucho más...





# Bootstrap - Ejercicio



- Un nuevo proyecto
- Partir del [template de Bootstrap](#) (Starter Template)
- Haz una web
- Usa componentes variados y experimentar con el Grid

# Referencias

- Luján Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario.
- Martín, A. R., & Martín, M. J. R. (2014). Aplicaciones web. Ediciones Paraninfo, SA.
- Luján Mora, S. (2001). Programación en Internet: clientes web. Editorial Club Universitario.
- Luján Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario.

# Referencias

- 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>