

Tema 0: Planificación de la asignatura

Asignatura: Diseño de Interfaces Web



Planificación de la asignatura

- Impartida por Tacio Camba Espí
 - Contacto - tacio@ciclosmontecastelo.com
 - Tutorías - Concertar cita previa por mail
 - En remoto - A conveniencia de alumno y profesor
 - Presencial - En el centro, **viernes de 20:00 a 21:00**
 - Dudas, sugerencias y quejas
- 12 sesiones (10 ordinarias + 1 repaso + 1 examen)
 - **Viernes 17:00 a 20:00** - Desde el 22/09/2022 al 26/01/2023
 - Sesión de repaso - Viernes 19/01/2023
 - Examen 1ra convocatoria - Viernes 26/01/2024
 - Examen 2da convocatoria - Viernes 31/05/2024

Evaluación

- 3 tareas optionales (30% de la nota total)
 - Ejercicio teórico / prácticos sobre uno o varios temas impartidos en el momento de su publicación
 - Publicada como tarea en el *Classroom* de la asignatura
 - Entrega de la tarea
 - Fecha y hora límite - Ver planificación y enunciado (prioridad del enunciado)
 - Medio de entrega - Tarea de *Classroom* correspondiente
 - Fichero **apellido-nombre.zip** (no se corregirá de otro modo)
 - Retrasos en la fecha de entrega
 - Menos de 24 h → Se restan 1 punto
 - Entre 24 y 48 h → Se restan 2 puntos
 - Entre 48 y 72 h → Se restan 3 puntos
 - **Más de 72 h → Suspenso con un 0**
 - Publicación de calificaciones y soluciones en *Classroom*
 - Fecha de entrega oficial + 10 días

Evaluación

- Examen (1ra conv. - 70% / 2da conv. - 100%)
 - Prueba teórico-práctica sobre los contenidos de la asignatura al completo
 - 1ra convocatoria - Viernes 26/01/2024
 - 2da convocatoria - Viernes 31/05/2024
- Las calificación final de la asignatura se redondeará al entero más próximo
 - 8.4 ~ 8
 - 5.5 ~ 6
- La calificación del examen en 2da convocatoria será la media de la nota del examen oficial y el de recuperación. En caso de que la calificación sea superior a 4.5 pero inferior a 5, se calificará con un 5 (siempre y cuando el examen de recuperación esté aprobado con un 5 o más)
- Plagio (examen y tareas)
 - Calificación de 0 para el que copia
 - Calificación de 0 para el que es copiado
- Revisión de calificaciones
 - Se solicitará por mail al profesor en los **7 días siguientes** a la comunicación de la nota
 - La solicitud de revisión implica recalificación (a favor o en contra del alumno) en caso de errata

Planificación de sesiones, tareas y exámenes

Septiembre							Octubre							Noviembre						
L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D
				1	2	3							1			1	2	3	4	5
4	5	6	7	8	9	10	2	3	4	5	6	7	8	6	7	8	9	10	11	12
11	12	13	14	15	16	17	9	10	11	12	13	14	15	13	14	15	16	17	18	19
18	19	20	21	22	23	24	16	17	18	19	20	21	22	20	21	22	23	24	25	26
25	26	27	28	29	30		23	24	25	26	27	28	29	27	28	29	30			
						30	31													
Diciembre							Enero													
L	M	X	J	V	S	D	L	M	X	J	V	S	D							
				1	2	3	1	2	3	4	5	6	7							
4	5	6	7	8	9	10	8	9	10	11	12	13	14							
11	12	13	14	15	16	17	15	16	17	18	19	20	21							
18Ex	19	20	21	22	23	24	22	23	24	25	26Ex	27	28							
25	26	27	28	29	30	31	29	30	31											

	Sesión Teórica
	Sesión Repaso
	Examen
	Tarea

* Las clases se impartirán en horario de 17:00 a 20:00

Planificación de sesiones, tareas y exámenes

- Tarea 1:
 - Fecha de publicación: miércoles 11 de octubre de 2023
 - Fecha de entrega: 23:59 del miércoles 15 de noviembre de 2023
 - Fecha final de prórroga: 23:59 del sábado 4 de noviembre de 2023
 - Contenido: Tema 1 (Introducción) y Tema 2 (CSS básico)
 - Ponderación sobre la media: 20 %
- Tarea 2:
 - Fecha de publicación: miércoles 15 de noviembre de 2023
 - Fecha de entrega: 23:59 del miércoles 13 de diciembre de 2023
 - Fecha final de prórroga: 23:59 del domingo 26 de noviembre de 2023
 - Contenido: Tema 3 (CSS Avanzado)
 - Ponderación sobre la media: 35 %
- Tarea 3:
 - Fecha de publicación: miércoles 13 de diciembre de 2023
 - Fecha de entrega: 23:59 del lunes 29 de enero de 2024
 - Fecha final de prórroga: 23:59 del jueves 1 de febrero de 2024
 - Contenido: Tema 5 (Bootstrap) y Tema 6 (Bootstrap)
 - Ponderación sobre la media: 45 %

Materiales y recursos

- Apuntes, enunciados y videos
 - Classroom de la asignatura
 - Acceso por invitación enviada al correo del ciclo
 - Invitación no recibida → Mail al profesor
- Código
 - Repositorio Github de la asignatura
 - <https://github.com/tcamba-ciclos-montecastelo/diw-23>
- Todas las sesiones serán grabadas
 - Classroom

Temario

1. Introducción (~ 1 sesión)

- 1.1. [Una visión a alto nivel del desarrollo web](#)
- 1.2. [Configurando el editor de código](#)
- 1.3. [Un breve repaso a HTML\(5\)](#)
- 1.4. [Una pequeña introducción a CSS\(3\)](#)
- 1.5. [Colores](#)
- 1.6. [Herramientas de desarrollador](#)
- 1.7. [Referencias](#)

2. CSS Básico (~ 2 sesiones)

- 2.1. [Selectores](#)
- 2.2. [Herencia](#)
- 2.3. [Modelo de caja \(Box Model\)](#)
- 2.4. [Posicionamiento Absoluto](#)
- 2.5. [Pseudo-elementos](#)
- 2.6. [Los 3 modos de construir diseños \(Layouts\)](#)
- 2.7. [Referencias](#)

3. CSS Avanzado (~ 3 sesiones)

- 3.1. [Propiedad *Float*](#)
- 3.2. [Box Model, Border Box](#)
- 3.3. [Módulo *Flexbox*](#)
- 3.4. [Uso del Grid CSS](#)
- 3.5. [Media queries](#)
- 3.6. [Transiciones](#)
- 3.7. [Animaciones](#)
- 3.8. [Dimensiones CSS](#)
- 3.9. [Seminario - Píxeles CSS](#)
- 3.10. [Referencias](#)

4. Introducción a Bootstrap (~ 2 sesiones)

- 4.1. [¿Qué es Bootstrap\(5\)?](#)
- 4.2. [Breakpoints](#)
- 4.3. [Containers](#)
- 4.4. [BS Grid](#)
- 4.5. [Utilidades](#)
- 4.6. [Componentes](#)
 - 4.6.1. [Cards](#)
 - 4.6.2. [Dropdowns](#)
 - 4.6.3. [Navbars](#)
 - 4.6.4. [Carousel](#)
- 4.7. [Referencias](#)

5. Introducción a SASS (~ 2 sesiones)

- 5.1. [¿Qué es SASS?](#)
- 5.2. [Anidamiento](#)
- 5.3. [Variables e interpolación](#)
- 5.4. [At-Rules](#)
- 5.5. [Mixins e @include](#)
- 5.6. [Flow Control](#)

6. Índice de grabaciones

7. Índice de grabaciones 22-23

Tema 1: Introducción

Asignatura: Diseño de Interfaces Web



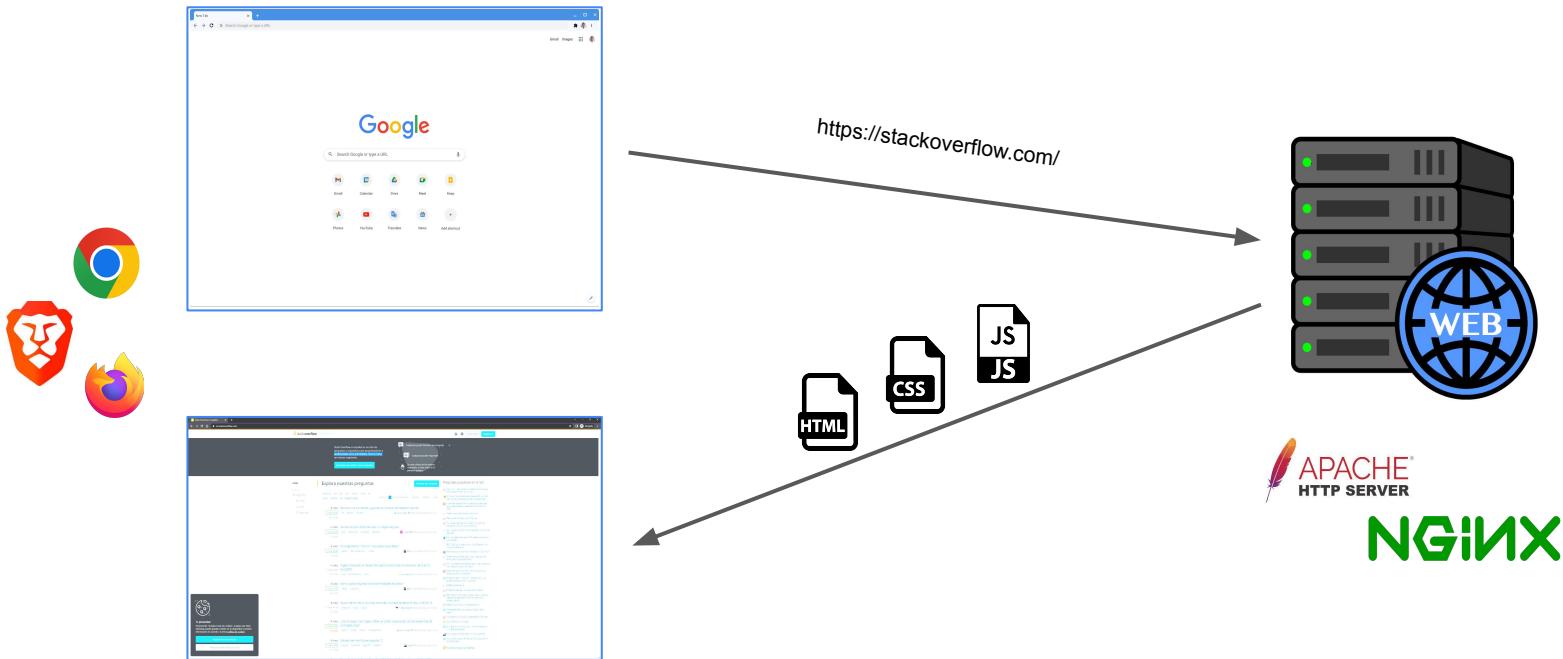
Contenidos

1. [Una visión a alto nivel del desarrollo web](#)
2. [Configurando el editor de código](#)
3. [Un breve repaso a HTML\(5\)](#)
4. [Una pequeña introducción a CSS\(3\)](#)
5. [Colores](#)
6. [Herramientas de desarrollador](#)
7. [Referencias](#)

Desarrollo front-end vs back-end

1. Un usuario solicita acceso a un sitio web desde su navegador
2. El servidor web (nginx, apache http...) responde al navegador web devolviéndole un conjunto de ficheros HTML, CSS y JS que aloja
3. El navegador recibe estos ficheros interpreta su código y renderiza la página web solicitada por el usuario

Desarrollo front-end vs back-end



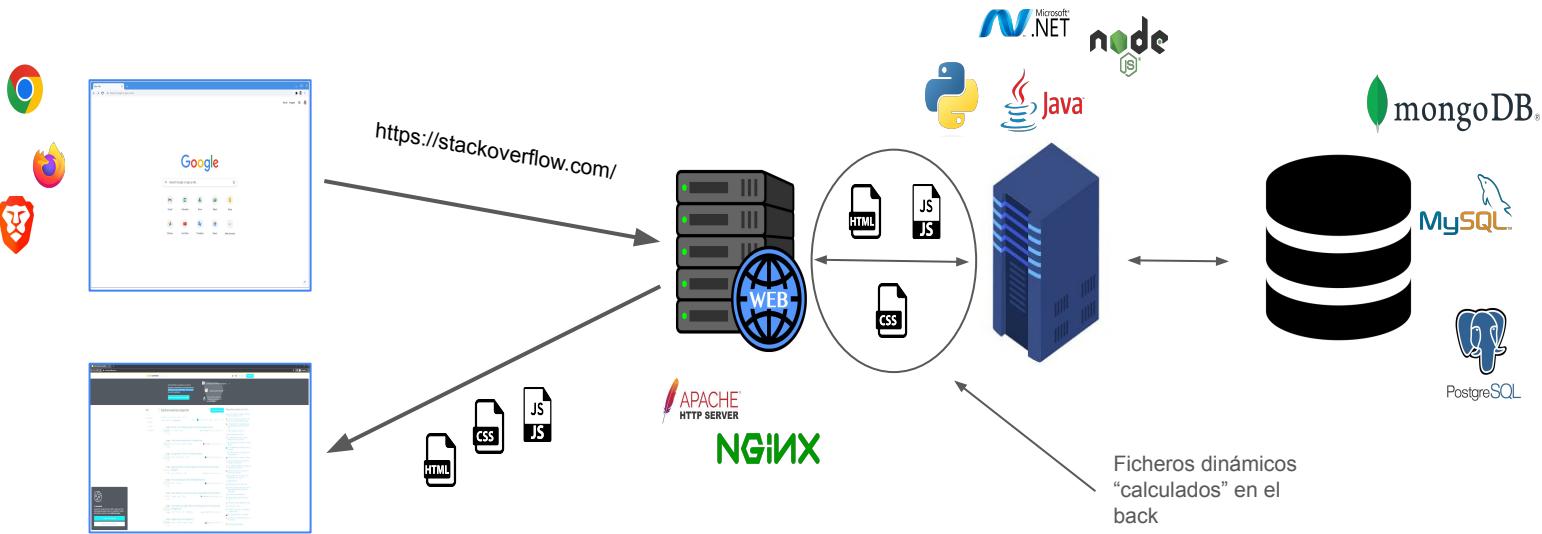
Desarrollo front-end vs back-end

- El proceso de escritura de estos ficheros **HTML, CSS y Javascript** que el navegador es capaz de interpretar recibe el nombre de **desarrollo Front-End**
- Las tecnologías **base** del desarrollo **front-end** son **HTML, CSS y Javascript**
- En el ejemplo anterior, los **ficheros** que componen el **sitio web** al que accede el usuario, se encuentran almacenados en un servidor web y son enviados tal cual al navegador, **sin ninguna transformación**. Tenemos entonces un sitio web **estático**...
- ... que no es lo habitual en la actualidad, como veremos en el ejemplo siguiente

Desarrollo front-end vs back-end

1. Un usuario solicita acceso a un sitio web desde su navegador
2. El **servidor web** devuelve al navegador del usuario los **ficheros** correspondientes, pero en esta ocasión, **transformados** por una **aplicación web** o servidor de **backend** (que se apoya en **servidores de BD**) de acuerdo a alguna lógica
3. El navegador web interpreta los ficheros recibidos mostrando al usuario un sitio web construido en tiempo real conforme a la lógica de la aplicación

Desarrollo front-end vs back-end



Desarrollo front-end vs back-end

- El proceso de escribir la **lógica de aplicación** con la que **ensamblar** los ficheros que constituyen el sitio web de forma **dinámica** se denomina **desarrollo Back-End**
- Es importante distinguir entre un servidor web y un servidor de aplicaciones o backend
 - El **servidor web** devuelve un conjunto de ficheros que **aloja** en respuesta a una solicitud de navegador
 - El **servidor** de aplicaciones o **backend** ejecuta la **lógica** que determina el comportamiento **dinámico** de un sitio web
- Python, Java, PHP, Node JS... son algunos ejemplos de lenguajes de programación backend

Los 3 lenguajes del front-end

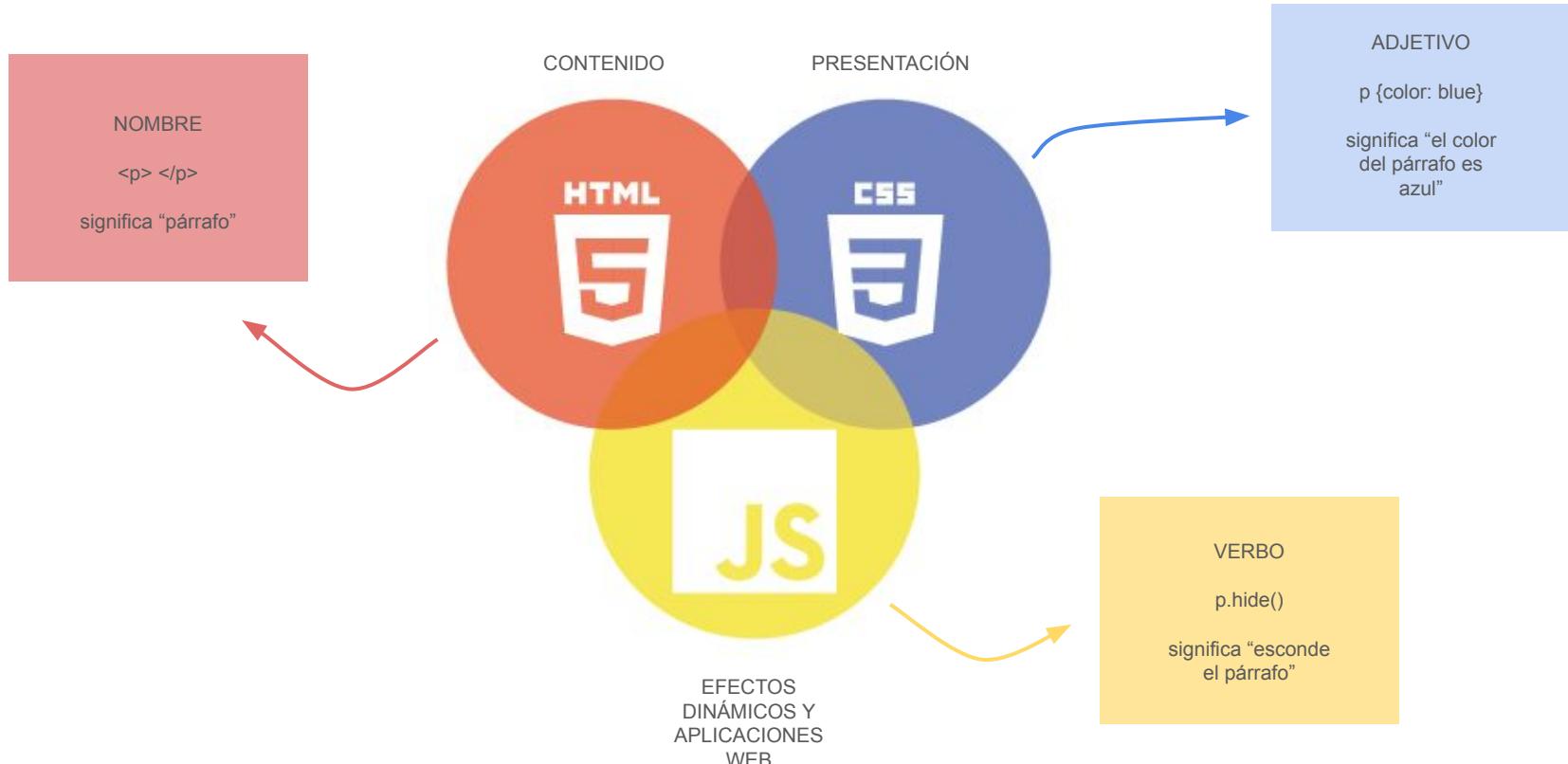
El desarrollo back-end no entra en el alcance de esta asignatura, de modo que nos centraremos en los tres lenguajes o tecnologías base del desarrollo front-end



Los 3 lenguajes del front-end

- HTML es el lenguaje que permite definir la estructura y el contenido de una página web
- CSS permite describir la presentación, el aspecto y la disposición de dicho contenido
- Javascript dota de efectos dinámicos e interactivos al sitio web
- JS puede emplearse para construir aplicaciones web completas (back-end incluido)
- HTML y CSS **NO** son lenguajes de programación, Javascript **SI** lo es

Los 3 lenguajes del front-end



Los 3 lenguajes del front-end

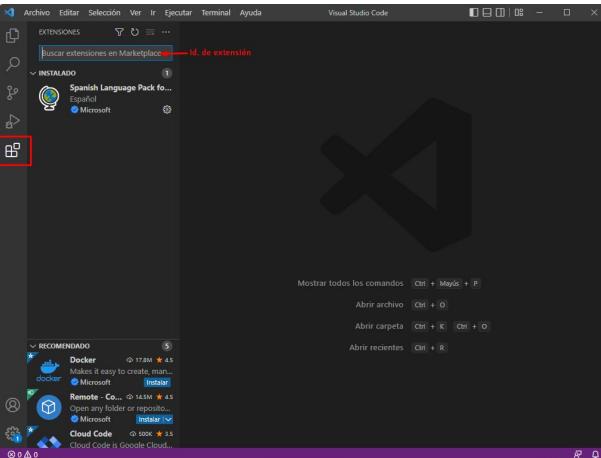
- Para construir un sitio o aplicación web se utiliza una combinación de HTML, CSS y Javascript
- El desarrollo front-end moderno se realiza utilizando “frameworks”, que son paquetes de software que proporcionan módulos de código reutilizable, tecnologías de front-end estandarizadas y bloques o componentes que hacen más fácil y rápido para el desarrollador la construcción de aplicaciones web o interfaces de usuario.
- Existen varios frameworks de desarrollo front-end y la mayoría se basan en Javascript. Entre los más populares se encuentran [React](#), [Vue](#), [Svelte](#), [Angular](#)...

Configurando el editor de código

- HTML, CSS (recordemos que no son lenguajes de programación) y JS no requieren sistemas en tiempo de ejecución, máquinas virtuales o compiladores específicos ya que son renderizados (HTML y CSS) e interpretados por el navegador web
- Un editor de texto plano y un navegador web son suficientes para el desarrollo web
- Existen IDEs (más bien editores de texto “potenciados”) que facilitan enormemente el proceso de desarrollo
 - [Visual Studio Code](#) - Desarrollado por Microsoft, es posiblemente el más empleado en desarrollo web en este momento. Es gratuito, ligero y cuenta con un ecosistema de plugins muy extenso. Será el empleado durante el curso.
 - [Sublime Text](#) - Similar a VSCode pero de pago (aunque dispone de una versión de evaluación)
 - [Webstorm](#) - IDE de pago (aunque gratuito para ciertos colectivos, como estudiantes o profesores), desarrollado por [Jetbrains](#). A diferencia de los anteriores no requiere configuración inicial alguna para ser utilizado de forma productiva.
- Durante el curso se utilizará [Visual Studio Code](#)

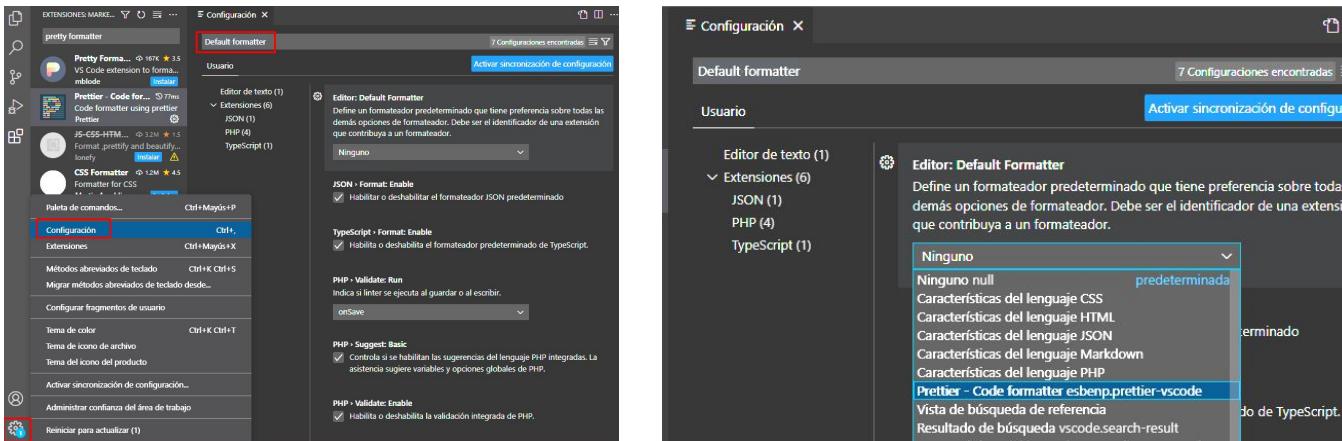
Configurando el editor de código

- Descargar e instalar VSCode desde su [web oficial](#). Dispone de versiones para los SO principales.
 - Dispone de una [versión de navegador](#) (con limitaciones respecto de la versión de escritorio), indicada para desarrollos rápidos o desde los repositorios de Azure
- Ejecutar VSCode e instalar la siguiente extensión



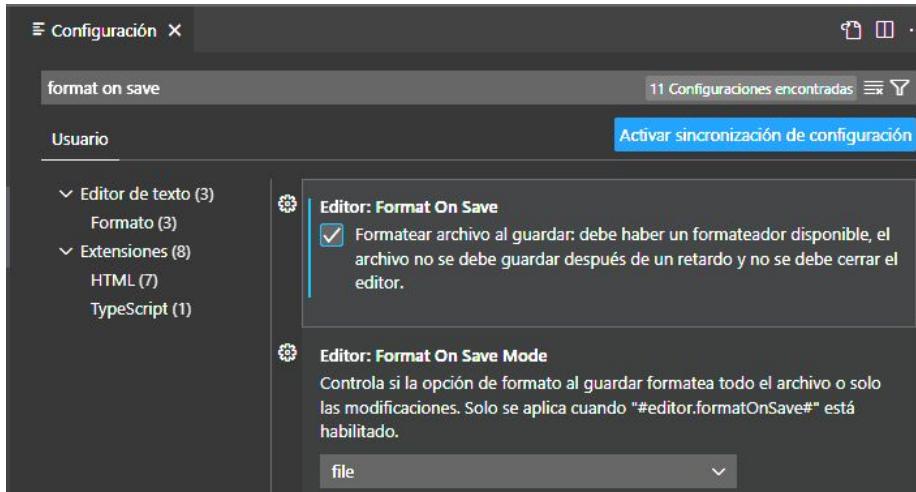
Configurando el editor de código

- **Prettier Code Formatter** - Formateo automático del código. Mantiene los proyectos limpios, uniformes y coherentes, a pesar del formato “opinionario”
 - Id. de extension - **esbenp.prettier-vscode**
- Configuración
 - Establecer *Prettier* como formateador por defecto (*default formatter*)



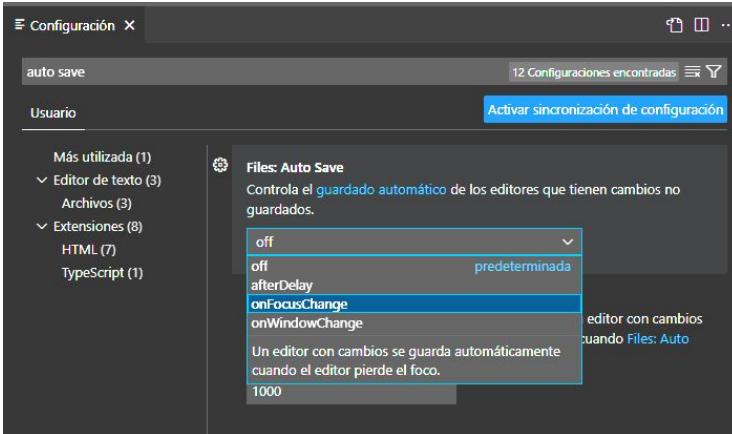
Configurando el editor de código

- Configuración
 - Establecer Prettier como formateador por defecto (*default formatter*)
 - Activar el formateo automático al guardar (*format on save*)



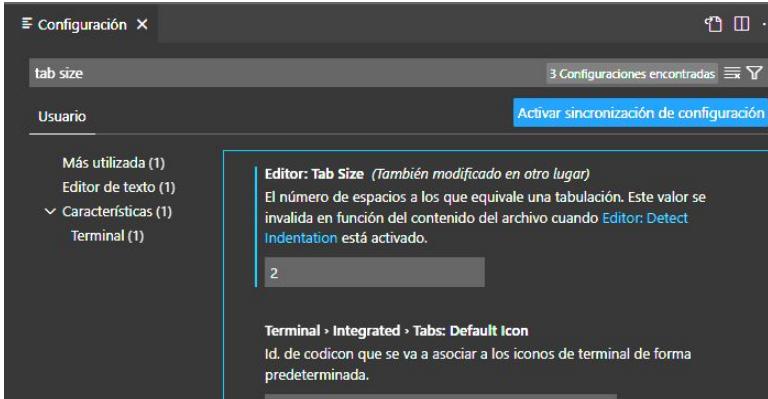
Configurando el editor de código

- Configuración
 - Establecer Prettier como formateador por defecto (*default formatter*)
 - Activar el formateo automático al guardar (*format on save*)
 - Activar el guardado automático cuando se cambia fichero o se abre una nueva pestaña (*auto save*)



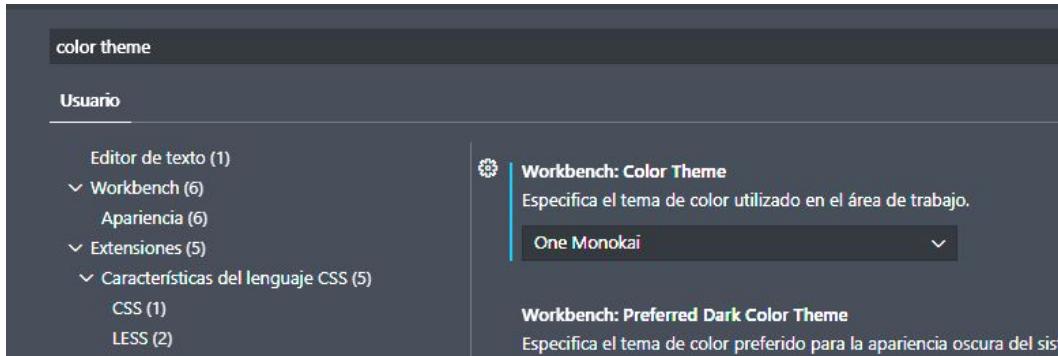
Configurando el editor de código

- Configuración
 - Establecer Prettier como formateador por defecto (*default formatter*)
 - Activar el formateo automático al guardar (*format on save*)
 - Activar el guardado automático cuando se cambia fichero o se abre una nueva pestaña (*auto save*)
 - Establecer el número de espacios por tabulación a 2 (*tab size*)



Configurando el editor de código

- Modificar el tema o estilo visual del editor
 - Se puede personalizar de forma manual
 - ... o utilizar alguno de los temas disponibles en el *marketplace* (funcionan como extensiones)
 - En este curso se empleará el tema ***One Monokai***
 - Id. de extensión - ***azemoh.one-monokai***
 - Configuración (*color theme*)



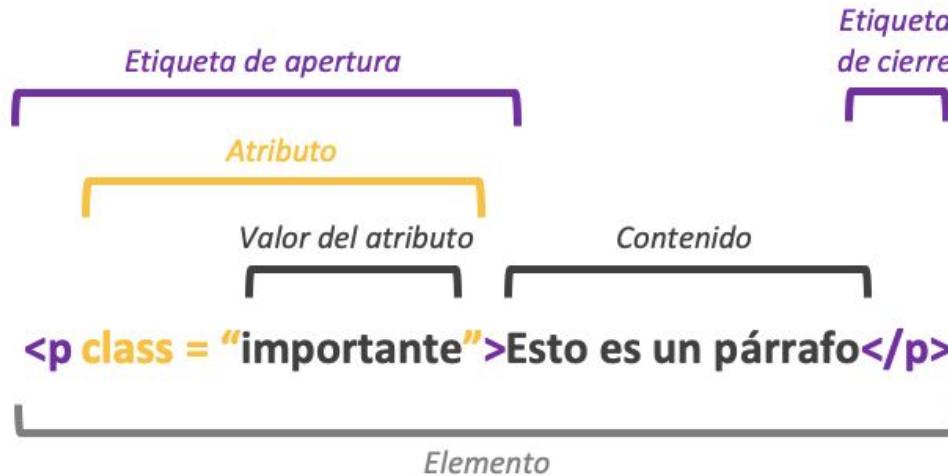
Configurando el editor de código

- Instalar las siguientes extensiones
 - **Image Preview** - Previsualización de imágenes.
 - Id. de extension - **kisstkondoros.vscode-gutter-preview**
 - **Color Highlight** - Visualización de colores en código
 - Id. de extension - **naumovs.color-highlight**
 - **Auto Rename Tag** - Renombrado automático de tags
 - Id. de extension - **formulahendry.auto-rename-tag**
 - **Live Preview**
 - Id. de extension - **ms-vscode.live-server**
 - **Live Server** - Servidor web embebido en el editor
 - Id. de extension - **ritwickdey.LiveServer**
 - <https://htmlcolorcodes.com/es/>
- Activar el cierre automático de tags
 - Configuración > **auto closing tags** > Habilitar el checkbox

Un breve repaso a HTML(5)

- HyperText Markup Language
- Es un lenguaje de marcas, que permite al desarrollador describir y estructurar el contenido de una página web (no es un lenguaje de programación, tercer aviso)
- Está formado por **elementos** que describen diferentes tipos de contenido: párrafos, enlaces, encabezados, imágenes, video...
- Renderizado por los navegadores web

Un breve repaso a HTML(5)



Un breve repaso a HTML(5)

- Elementos estructurales básicos

```
<!DOCTYPE html> <!-- Indica al navegador que se trata de un documento HTML -->
<html lang="en"> <!-- Inicio del documento HTML. Se especifica el lenguaje en el atributo lang -->
<head> <!-- Contiene metadatos. Los metadatos no se renderizan. Son "pistas" para el navegador -->
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge"> <!-- Compatibilidad con navegadores de Microsoft -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- viewport = ancho de la pantalla del dispositivo -->
  <title>Document</title> <!-- viewport = ancho de la pantalla del dispositivo --> <!-- Se muestra en la barra del navegador. Importante para SEO -->
</head>
<body> <!-- Define el cuerpo del documento, su contenido visualizable -->

</body>
</html>
```

- El **viewport** es el área útil donde se muestra el sitio web
 - Si no se especifica su tamaño, el navegador escala el contenido para que se visualice completo en la pantalla del dispositivo
 - Si especificamos su tamaño los contenidos se ajustarán a la pantalla de cada dispositivo de forma correcta

Un breve repaso a HTML(5)

- Construcción de una página web básica a partir del contenido sin formatear
 - Clonar el siguiente repositorio:
<https://github.com/tcamba-ciclos-montecastel/o/diw-23>
 - ... o descargar el **fichero ZIP** publicado en el **Classroom** de la asignatura en la sección correspondiente al **Tema 1 - Introducción**
 - Directorio [t1-introduccion/html/web-blog](#)

The Code Magazine

[Blog Challenges](#) [Explore CSS Grid](#)

The Basic Language of the Web: HTML



Posted by [Laura Jones](#) on Monday, June 21st 2021



All modern websites and web applications are built using three fundamental technologies: HTML, CSS and JavaScript. These are the language of the web. In this post, let's focus on HTML. We will learn what HTML is all about, and why you too should learn it.

What is HTML?

HTML stands for HyperText Markup Language. It's a markup language that web developers use to structure and describe the content of a webpage (not a programming language).

HTML consists of elements that describe different types of content: paragraphs, links, headings, images, video, etc. Web browsers understand HTML and render HTML code as websites.

In HTML, each element is made up of 3 parts:

1. The opening tag
2. The closing tag
3. The actual element

You can learn more at [MDN Web Docs](#).

Why should you learn HTML?

There are countless reasons for learning the fundamental language of the web. Here are 5 of them:

- To be able to use the fundamental web dev language
- To hand-craft beautiful websites instead relying on tools like Wordpress or Wix
- To build web applications
- To impress friends
- To have fun! ☺

Hopefully you learned something new here. See you next time!

Related posts



[How to Learn Web Development](#)

By Jesus Schindlermann



[The Unicorn Power of CSS](#)

By Jim Dillon



[Why JavaScript is Awesome](#)

By Manida

Copyright © 2021 by The Code Magazine.

Un breve repaso a HTML(5)

```

<!-- Contenido de encabezado de la web. Tiene un carácter semántico -->
<header>
  <h1> The Code Magazine</h1> <!-- Elemento de título o encabezado que describe el tema de la sección que presenta -->

  <nav> <!-- Especifica un menú o elemento de navegación. Carácter semántico -->
    <!-- Anchor o enlace. Contiene un enlace a otra página o recurso web -->
    <a href="blog.html" target="_blank">Blog</a> <!-- href: URI del recurso web que se desea enlazar -->
    <a href="#">Challenges</a> <!-- href="#" es un enlace vacío -->
    <a href="#">Flexbox</a>
    <a href="#">CSS Grid</a>
  </nav>
</header>

<!-- Define una sección del contenido, un item en un e-commerce... Carácter semántico -->
<article>
  <header>

    <!-- De h1 a h6 con importancia y tamaño de fuente descendente. Se emplean para construir tablas de contenidos -->
    <h2> The Basic Language of the Web: HTML </h2>

    <!-- Elemento para incluir imágenes -->
    
      alt="Headshot of Laura Jones" <!-- Descripción de la imagen. Se muestra en caso de que la imagen no cargue. screen readers -->
      height="50" <!-- Se escalará la imagen a estas dimensiones -->
      width="50" <!-- Si se indican height y width puede romperse la relación de aspecto -->
    />

    <!-- <p> Distribuye el texto en párrafos -->
    <!-- <strong> Muestra el contenido en negrita. Preferible a <b> que semánticamente no tiene significado -->
    <p>Posted by <strong>Laura Jones</strong> on Monday, June 21st 2027</p>

  </header>

```

Un breve repaso a HTML(5)

```
<p>
  All modern websites and web applications are built using three
  <em>fundamental</em> <!-- <em> Aplica énfasis (cursiva o itálica) a su contenido. Preferible a <i> por razones semánticas
  -->
  technologies: HTML, CSS and JavaScript. These are the languages of the
  web.
</p>

...
<!-- Crea una lista ordenada de elementos -->
<ol>
  <li>The opening tag</li> <!-- Define un elemento en la lista que lo contiene -->
  <li>The closing tag</li>
  <li>The actual element</li>
</ol>

<p>
  You can learn more at
  <a
    href="
      https://developer.mozilla.org/en-US/docs/Web/HTML
    
    target="_blank" <!-- El enlace se abre en una nueva pestaña del navegador -->
  >
    MDN Web Docs
  </a>
</p>
```

Un breve repaso a HTML(5)

```
<aside> <!-- Otro elemento semántico de agrupación de contenido secundario o auxiliar de la web -->
  <h4>Related posts</h4>

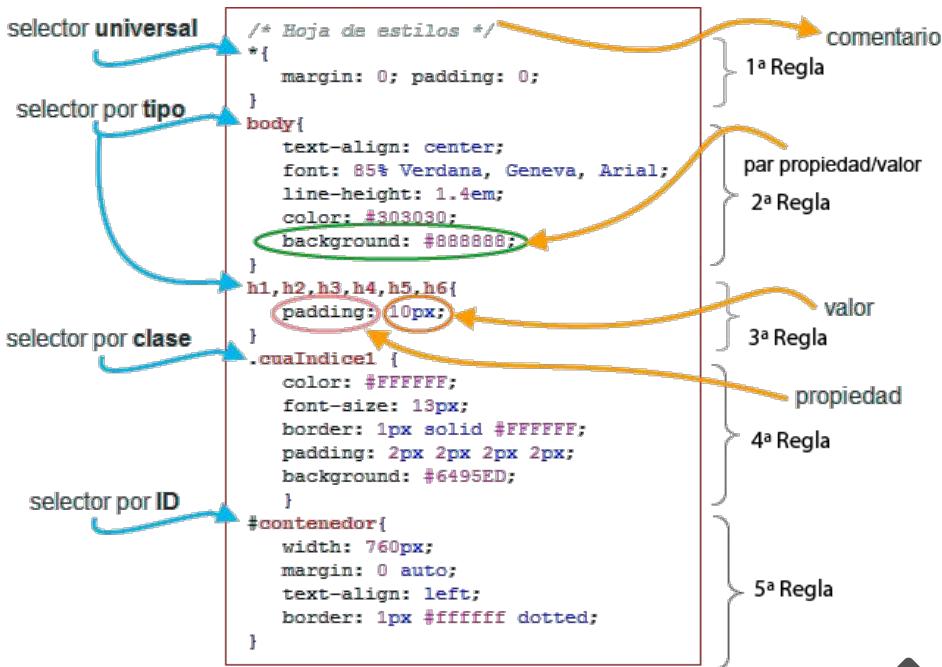
  <ul> <!-- Lista desordenada de elementos -->
    <li>
      
      <a href="#">How to Learn Web Development</a>
      <p>By Paschalis Aladdin</p>
    </li>

    ...
    </ul>
  </aside>

  <!-- Elemento semántico que define el pie de la página web -->
  <footer>Copyright © 2027 by The Code Magazine.</footer> <!-- © es una entidad HTML que renderiza al símbolo © -->
```

Una pequeña introducción a CSS(3)

- **Cascading Style Sheets**
- **CSS** describe el estilo visual y la presentación del contenido escrito en **HTML**
- **CSS** consiste en numerosas propiedades que los desarrolladores usan para formatear el contenido: propiedades sobre fuentes, texto, espacio, diseño, etc...



Una pequeña introducción a CSS(3)

- Construcción de una página web básica a partir del contenido sin formatear
 - Clonar el siguiente repositorio:
<https://github.com/tcamba-ciclos-montecastelo/diw-23>
 - ... o descargar el **fichero ZIP** publicado en el **Classroom** de la asignatura en la sección correspondiente al **Tema 1 - Introducción**
 - Directorio [t1-introduccion/css/web-blog](#)



The screenshot shows a blog post from The Code Magazine. The title is "The Basic Language of the Web: HTML". It's posted by Laura Jeeves on Monday, June 21st 2007. The post discusses the fundamental technologies of the web: HTML, CSS, and JavaScript. It highlights that HTML is a markup language used to structure and describe content. The sidebar contains sections on what HTML is, why you should learn it, and related posts.

Una pequeña introducción a CSS(3)

```
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>The Basic Language of the Web: HTML</title>

  <style> <!-- CSS interno (MAL). PRIORIDAD 3. Aplica a nivel de documento HTML. Acopla el HTML con el estilo CSS -->
    h1 {
      color: green;
    }
  </style>

  <!-- CSS externo (BIEN). PRIORIDAD 2. Aplica a nivel de documento HTML. -->
  <link rel="stylesheet" href="styles.css" /> <!-- Se enlaza con el elemento <link>. Se pueden enlazar múltiples ficheros CSS -->
</head>

<body>
  <header>
    <h1 style="color: blue">█ The Code Magazine</h1> <!-- CSS en línea (MUY MAL). PRIORIDAD 1. Sólo aplica al elemento en que se define -->

  <nav>
    ...
  </nav>
```

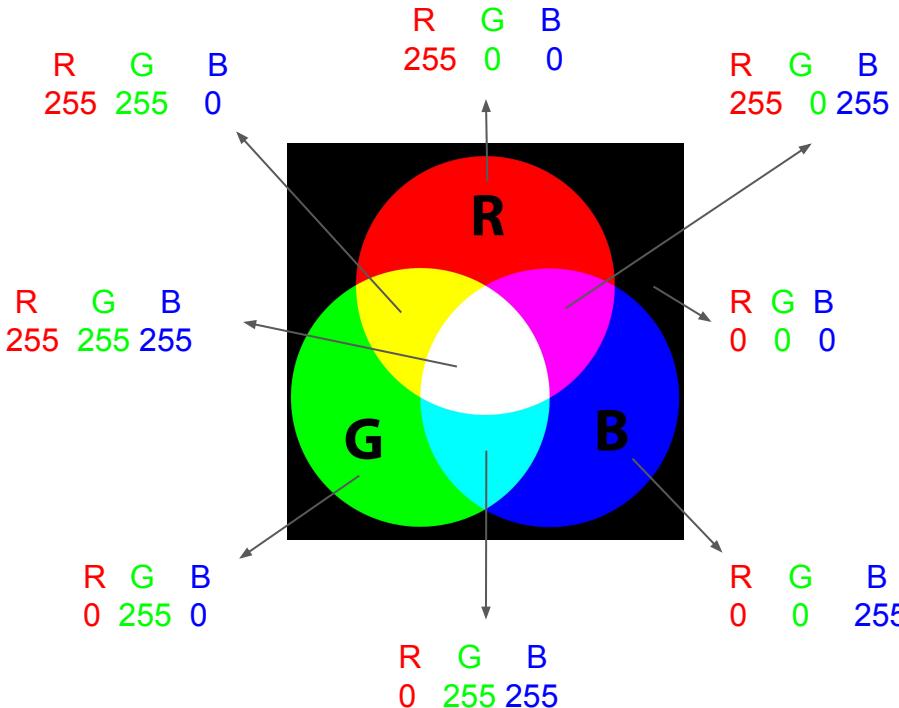


Una pequeña introducción a CSS(3)

```
h1 {  
    font-size: 26px; /* Tamaño de fuente en pixels */  
    font-family: sans-serif; /* Familia de fuente. No se emplea una específica por si el usuario no la tuviera disponible */  
    text-transform: uppercase; /* Todo el texto en mayúsculas */  
    font-style: italic; /* Todo el texto en cursiva */  
}  
  
h2 {  
    font-size: 40px;  
    font-family: sans-serif;  
}  
  
h3 {  
    font-size: 30px;  
    font-family: sans-serif;  
}  
  
h4 {  
    font-size: 20px;  
    font-family: sans-serif;  
    text-transform: uppercase;  
    text-align: center; /* Texto alineado en el centro horizontalmente */  
}  
  
/* Ver que aplica a elementos em, strong y a hijos ~ Herencia */  
p {  
    font-size: 22px;  
    font-family: sans-serif;  
    line-height: 1.5; /* Altura del espacio entre líneas. 1.5 veces el tamaño de la fuente, es decir 33px */  
}  
  
li {  
    font-size: 29px;  
    font-family: sans-serif;  
}
```

Colores

- **Modelo RGB:** Cada color puede ser representado como una combinación de ROJO, VERDE y AZUL
- Cada color base puede tomar valores entre 0 y 255
- RGB codifica por tanto 16.8 millones de colores diferentes



Colores

- Notación RGB
 - Modelo RGB estándar
 - rgb(0, 255, 255) 
 - Modelo RGB con transparencia
 - rgba(0, 255, 255, 0.3) 
- Normalmente se utiliza notación hexadecimal salvo cuando se necesita transparencia

#f4b33f

rgb(244, 179, 63)



Notación Hexadecimal

- En lugar de usar una escala de 0 a 255, vamos desde **0** a **FF** (hexadecimal)
 - #00ffff 
- Cuando todos los colores son pares idénticos se acorta a
 - #0ff

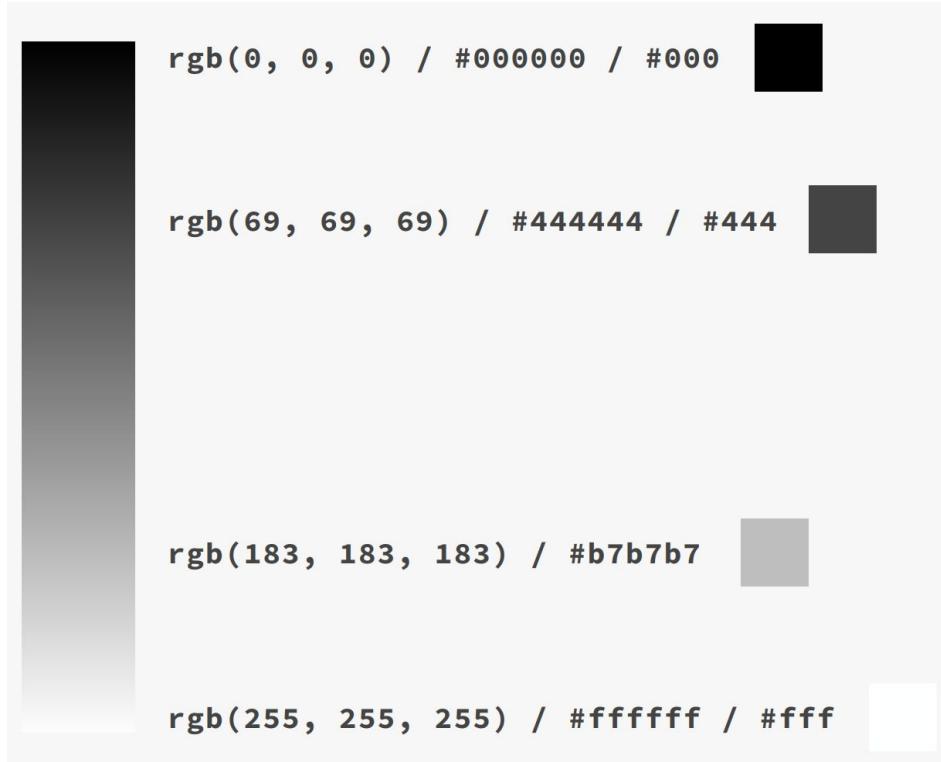
#f4b33fb3

rgba(244, 179, 63, 0.7)



Colores

- Cuando los colores en los 3 canales son los mismos obtenemos un tono de gris
- Hay 256 grises puros entre los que elegir



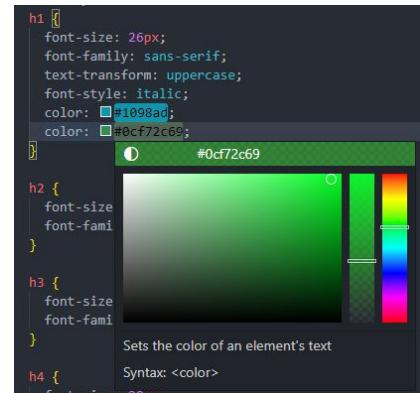
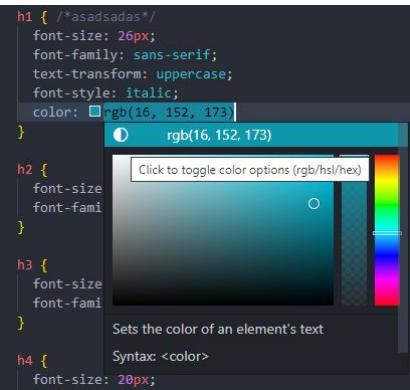
Colores

- Herramienta de selección de colores de VSCode
- ¿Qué color aplicará al texto de h1?

```

h1 {
    font-size: 26px;
    font-family: sans-serif;
    text-transform: uppercase;
    font-style: italic;
    color: #1098ad;
    color: #0cf72c69;
}

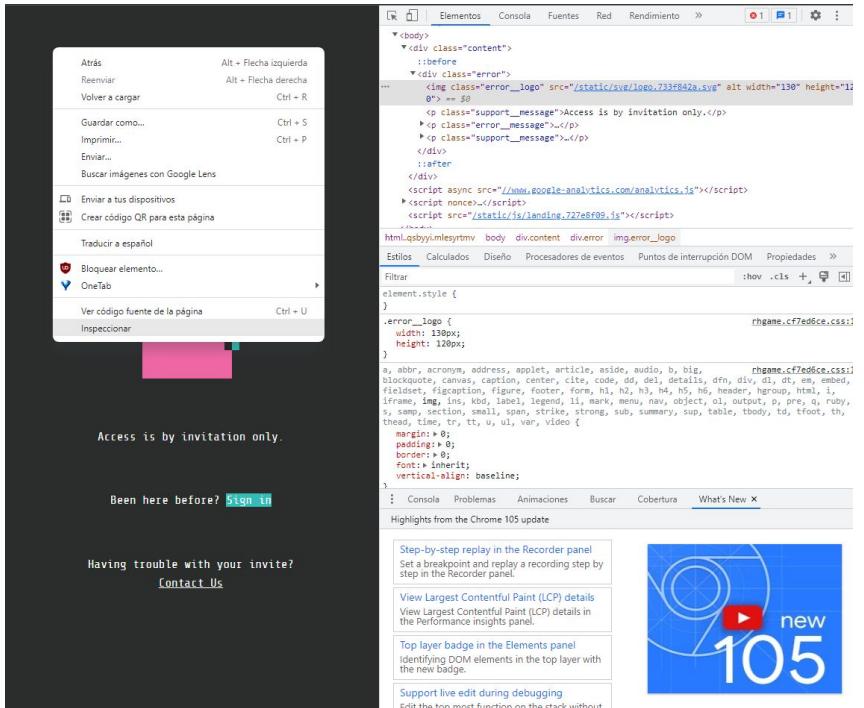
```



- El último en ser declarado

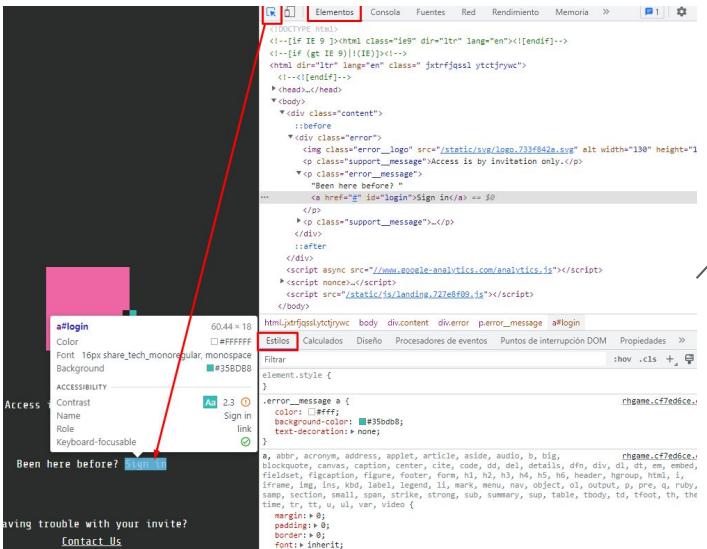
Herramientas de desarrollador

- Para seguir el curso se recomienda el uso del navegador Google Chrome
- Click derecho + Inspeccionar o F12
- Su manejo básico resulta casi imprescindible para desarrollo web



Herramientas de desarrollador

- Permite visualizar y editar el HTML y CSS de la página web o de un elemento a seleccionar


Modificado' element, which is selected in the Elements panel. The right-hand sidebar shows the element's properties: color (#0000ff), background-color (#10ff10), text-decoration (none), and font-size (5vh). Below the element, the 'Access' section shows 'Contrast' (A 2.3), 'Name' (Modificado), and 'Role' (link). A tooltip ' Been here before?' is shown above the element. At the bottom, there is a message 'Having trouble with your invite? Contact Us'."/>

```

<div class="support_message">Access is by invitation only.</div>
<div class="error_message">...</div>
<a href="#" id="login">Modificado</a> == $0
...
<p class="support_message">...</p>
</div>
:after
</div>
<script async src="//www.google-analytics.com/analytics.js"></script>
<script nonce=></script>
<script src="/static/islanding.727e8909.js"></script>
</body>
</html>

```

Access is by invitation only.

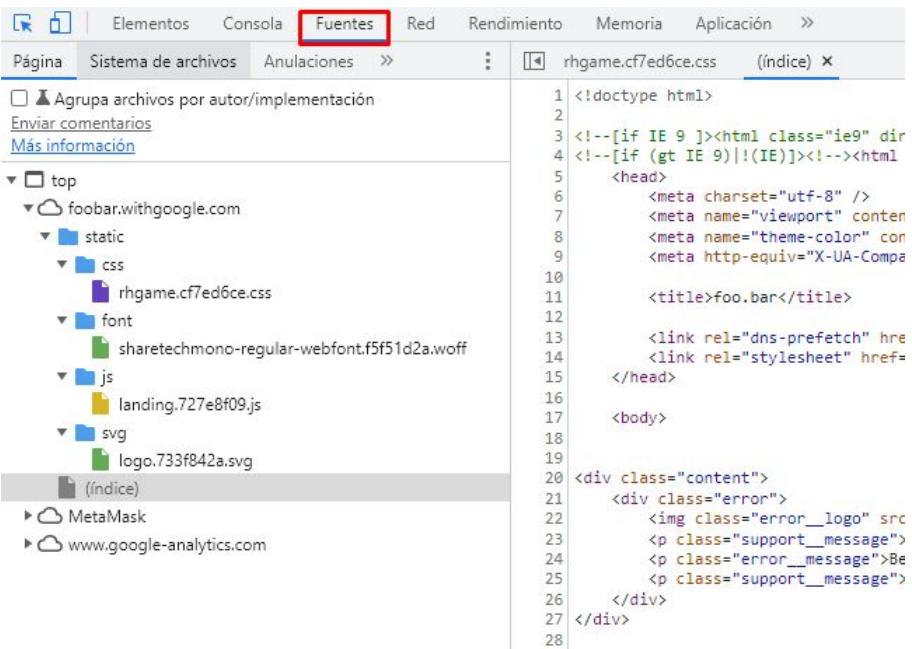
Been here before?

Modificado

Having trouble with your invite?
[Contact Us](#)

Herramientas de desarrollador

- Ficheros descargados por el navegador desde el servidor web para renderizar la página web
- index.html y otros ficheros como hojas de estilo, scripts JS, fuentes, imágenes, logos...



The screenshot shows the 'Fonts' tab selected in the developer tools. The left pane displays a tree view of network requests, including 'foobar.withgoogle.com' which has sub-folders for 'static', 'css', 'font', 'js', and 'svg'. Under 'css', there is a file named 'rhgame.cf7ed6ce.css'. The right pane shows the raw HTML code for the page, with line numbers from 1 to 28. The code includes doctype, meta tags, title, link tags for dns-prefetch and stylesheet, and a main content div.

```

1 <!doctype html>
2
3 <!--[if IE 9 ]><html class="ie9" dir="ltr">
4 <!--[if (gt IE 9 )|(IE)]><!--><html class="no-iframe" dir="ltr">
5   <head>
6     <meta charset="utf-8" />
7     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8     <meta name="theme-color" content="#000000" />
9     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
10    <title>foo.bar</title>
11
12    <link rel="dns-prefetch" href="https://www.google-analytics.com/>
13    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=ShareTechMono:ital,wght@0,400;0,700;1,400&subset=latin-ext&display=block"/>
14
15  </head>
16
17  <body>
18
19    <div class="content">
20      <div class="error">
21        
22        <p class="support__message">Este sitio web no responde a las peticiones de su navegador. Puede que el sitio esté caído o que no se pueda conectar con él. Si continúa teniendo problemas, por favor, inténtelo más tarde o contacte al administrador del sitio.</p>
23        <p class="error__message">Este sitio web no responde a las peticiones de su navegador. Puede que el sitio esté caído o que no se pueda conectar con él. Si continúa teniendo problemas, por favor, inténtelo más tarde o contacte al administrador del sitio.</p>
24
25      </div>
26    </div>
27  </body>
28

```

Referencias

- Visual Studio Code
 - [Emmet](#) - *CheatSheet* de Emmet. Emmet es una herramienta que extiende editores de texto como VSCode y proporciona entre otras cosas atajos o macros para facilitar el proceso de desarrollo
- HTML
 - [MDN Web Docs HTML](#) - Web de referencia de Mozilla con documentación sobre HTML5
 - [CSS Tricks HTML Entities](#) - Listado de entidades HTML disponibles
- CSS
 - [MDN Web Docs CSS](#) - Web de referencia de Mozilla con documentación sobre CSS3
 - [Codrops CSS](#) - Otra web de referencia muy completa con una explicación detallada de todos los elementos del lenguaje CSS
 - [CSS for People Who Hate CSS](#) - Guía sobre cómo escribir código CSS mejor, más limpio y reutilizable
- Colores
 - [InfinityInsight 3D Color Picker](#) - Herramienta que permite seleccionar colores sobre un cubo en 3d. Ayuda a sentar una intuición sobre el modelo RGB
 - [Otro selector de colores](#) - Herramienta para trastear con los colores
- Herramientas de desarrollador
 - [Chrome DevTools Docs](#) - Documentación oficial de las herramientas de desarrollador de Chrome



Tema 2: CSS Básico

Asignatura: Diseño de Interfaces Web



Contenidos

1. [Selectores](#)
2. [Herencia](#)
3. [Modelo de caja \(Box Model\)](#)
4. [Posicionamiento Absoluto](#)
5. [Pseudo-elementos](#)
6. [Los 3 modos de construir diseños \(Layouts\)](#)
7. [Referencias](#)

Selectores

- Definen sobre qué elementos HTML se aplican las reglas CSS
 - **Tipo** - Selecciona los elementos HTML por su **nombre**
 - **Clase** - Selecciona los elementos por el valor de su **atributo class**
 - **Id** - Selecciona los elementos por el valor de su **atributo id**. No puede haber 2 elementos con el mismo *id* dentro de un documento, su valor es único
 - **Universal** - Selecciona **todos los elementos**
 - **Atributo** - Selecciona los elementos basándose en el **valor de un atributo**
 - **Pseudoclase** - Selecciona los elementos en función de su **pseudoclase**, que es información de **estado** que no se encuentra en el **DOM**
 - **Pseudoelemento** - Permite seleccionar elementos que no tienen representación en el lenguaje HTML, como la primera línea de un párrafo, o los marcadores de una lista.



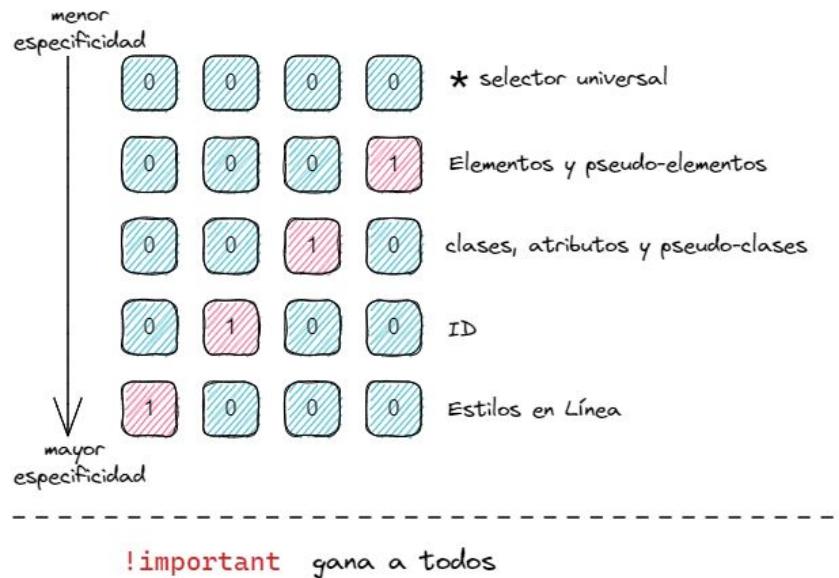
Selectores

- Combinación de selectores
 - Los selectores CSS se pueden combinar para seleccionar subconjuntos muy específicos de elementos HTML sobre los que aplicar las reglas CSS

```
selector #id .clase :pseudoclasse ::pseudoelemento [atributo] {  
    propiedad : valor ;  
    propiedad : valor  
}
```

Selectores

- Especificidad. Conflictos entre selectores
 - Cuando se combinan selectores es frecuente que acabemos aplicando estilos no deseados a determinados elementos.
 - Esto ocurre porque la **especificidad** del selector del estilo aplicado es superior a la de la regla deseada
 - El cálculo de la **especificidad** de un selector se resume en la siguiente figura



Selectores

- Algunos ejemplos de cálculo de especificidad

Selector	ID	Clases, atributos y pseudo-clases	Elementos y pseudo-elementos	Total
div#container #main ul li	2	0	3	203
table tbody tr.even td.odd	0	2	4	024
.wrapper h1 a	0	1	2	120
.navbar ul.menu li#first a:not(:visited)	1	3	3	133
.footer #menu li#sponsor a[rel=nofollow]::before	2	2	3	233



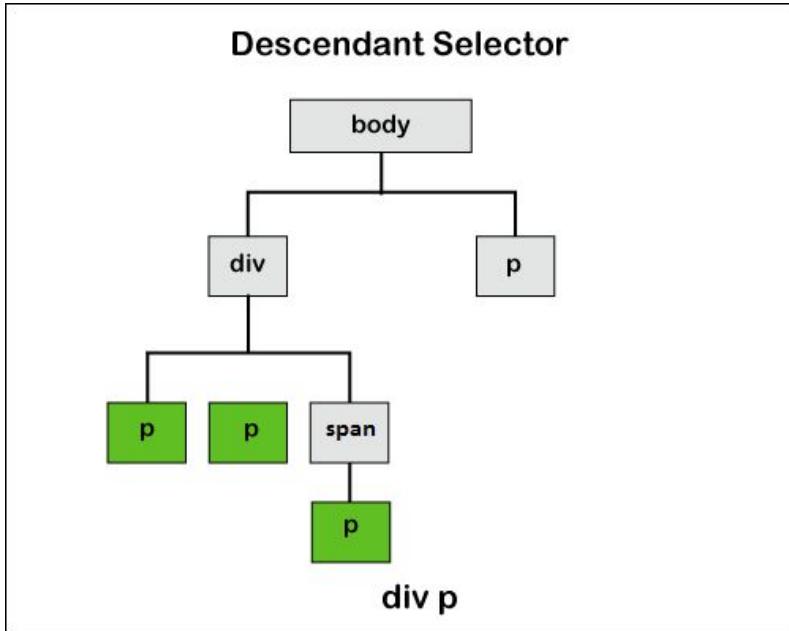
Selectores

- Combinadores CSS
 - Un combinador CSS es un símbolo que describe la relación entre selectores CSS
 - Junto con los tipos de selectores permiten construir reglas de selección complejas y muy específicas
 - Tipos de combinadores CSS
 - **Descendiente (espacio)** - Selecciona todos los elementos descendientes
 - **Hijo (>)** - Selecciona todos los elementos hijos (directos)
 - **Hermano adyacente (+)** - Selecciona el elemento inmediatamente contiguo
 - **Hermano general (~)** - Selecciona todos los elementos contiguos

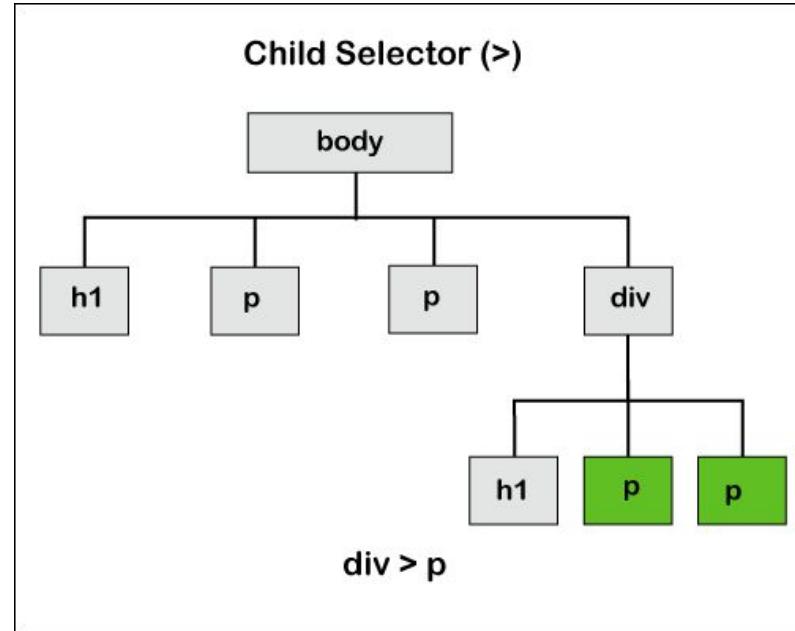


Selectores

- Combinador Descendiente

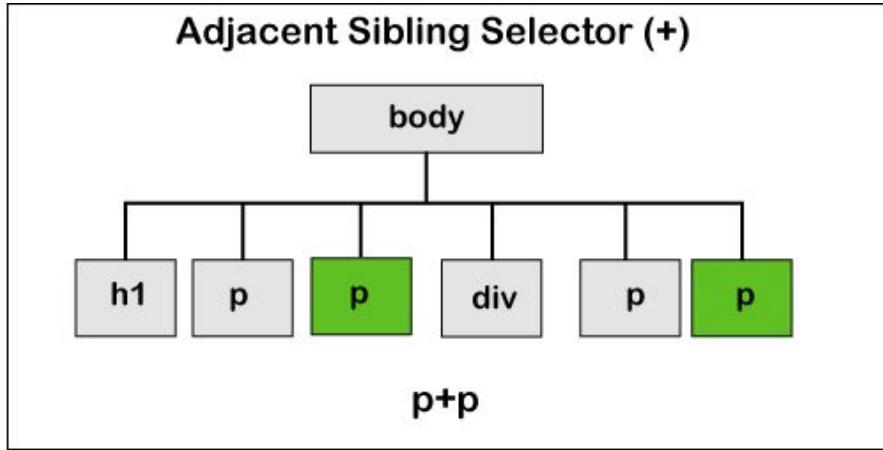
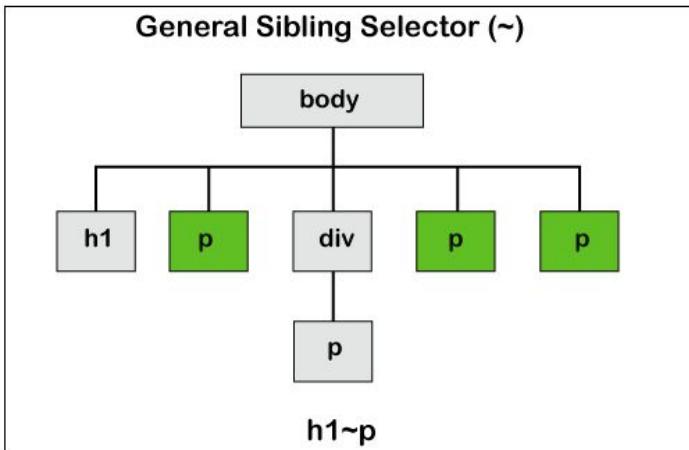


- Combinador Hijo



Selectores

- Combinador Hermano General
- Combinador Hermano Adyacente



Selectores

- Algún ejemplo espinoso

```
<div id="example3">  
    <p>First paragraph</p>  
    <p>Second paragraph</p>  
  
<div id="nestedDiv">  
    <p>Third paragraph</p>  
    </div>  
</div>
```

Selectores

```
#example3 p {
    color: red
}
```

Todos los p
descendientes de
#example3 a rojo

```
#example3 > p {
    color: blue
}
```

Todos los p hijos
directos de
#example3 a azul

```
#example3 > p {
    color: blue
}
```

Todos los p hijos
directos de
#example3 a azul

```
#example3 p {
    color: red
}
```

Todos los p
descendientes de
#example3 a rojo

CHILD SELECTOR ORDER

First paragraph

Second paragraph

Third paragraph

CHILD SELECTOR ORDER

First paragraph

Second paragraph

Third paragraph

Selectores

- En el caso anterior, dado que los combinadores no tienen impacto en la especificidad, el resultado es cuestión de orden
 - En el primer caso, ambos selectores aplican al tercer elemento de la lista, pero aplica el último declarado, es decir, el color azul
 - En el segundo caso, la situación es la inversa, el primer selector aplica a los dos primeros elementos de la lista, pero la segunda regla engloba a los 3 y sobreescribe el color azul aplicado en la regla anterior
 - Lo anterior rompe una regla básica del CSS, ir de los general a lo específico

Selectores

- Pseudo-clases

- Son palabras clave que pertenecen a un selector y que refieren el estado actual de un elemento
- Su sintaxis es `elemento:pseudo-clase`

```

/* Selecciona el PRIMER elemento li      /* Selecciona el 3er elemento li */
/*                                         li:nth-child(3) {
li:first-child {                           color: lightsalmon;
    color: goldenrod;                      }
}

/* Selecciona el ULTIMO elemento li      /* Selecciona el 2do elemento li
/*                                         empezando por el final */
li:last-child {                           li:nth-last-child(2) {
    color: blueviolet;                      color: orangered;
}
}

```

In HTML, each element is made up of 3 parts:

1. The opening tag
2. The closing tag
3. The actual element

You can learn more at the MDN Web Docs.

Why should you learn HTML?

There are countless reasons for learning the fundamental language of t

- To be able to use the fundamental web dev language
- To hand-craft beautiful websites instead of relying on tools like'
- To build web applications
- To impress friends
- To have fun 😊

Hopefully you learned something new here. See you next time!

Selectores

- Pseudo-clases

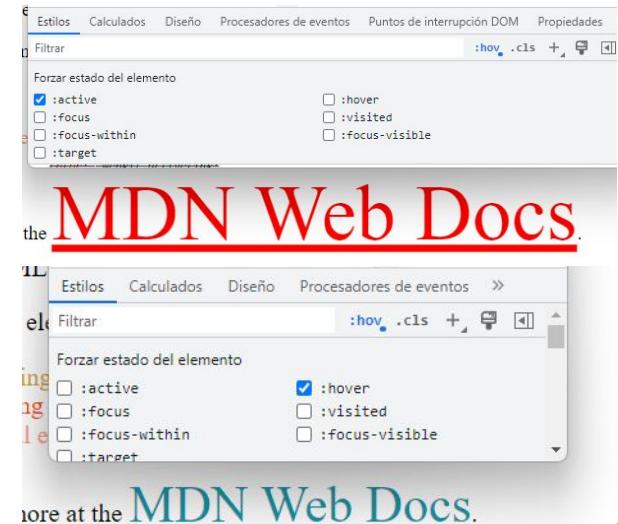
- Son palabras clave que pertenecen a un selector y que refieren el estado actual de un elemento
- Su sintaxis es `elemento:pseudo-clase`

```
a:link {
    text-decoration: none;
}

/* Selecciona el elemento a cuando
el cursor se coloca sobre él */
a:hover {
    color: #0f97ac;
    text-decoration: underline;
    font-size: calc(2 * 1em);
}

/* Selecciona el elemento a cuando
ha sido visitado */
a:visited {
    color: #042930;
    text-decoration: dashed
}

/* Selecciona el elemento a en el
instante del click */
a:active {
    color: red;
    font-size: calc(4 * 1em);
}
```



Herencia

- Es una característica de CSS que hace que algunas de las propiedades de los elementos padres sean heredadas por sus hijos
- No todas las propiedades CSS se heredan. Principalmente pasan de padres a hijos las propiedades relacionadas con texto, como las de la familia font-*
- Lista de propiedades que se heredan:
<https://gist.github.com/dcneiner/1137601>

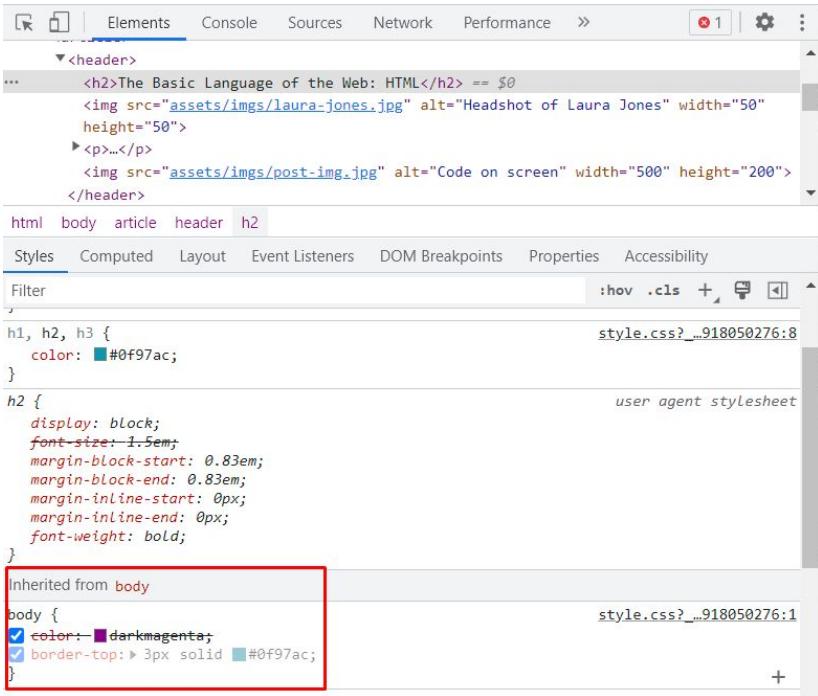
Herencia

```

body {
    color: darkmagenta;
    border-top: 3px solid #0f97ac;
}

h1,
h2,
h3 {
    color: #0f97ac;
}

```



The screenshot shows the Chrome DevTools Elements tab with the header element selected. The h2 color is being inspected. The 'Inherited from body' section is highlighted with a red box, showing the applied styles: color: darkmagenta; and border-top: 3px solid #0f97ac;

Herencia

- Selector universal
 - Queremos aplicar propiedades de forma global a todo el documento
 - No queremos que se hereden
 - Que sean fácilmente sobreescritas

```
* {
  color: darkmagenta;
  border-top: 3px solid #0f97ac;
}

h1,
h2,
h3 {
  border-top: none; /* Sobreescribe la propiedad
  de arriba */
  color: #0f97ac;
}
```

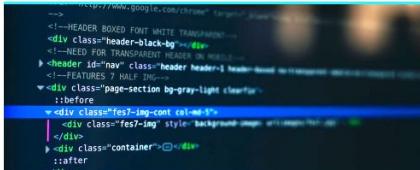
THE CODE MAGAZINE

Blog Challenges Flexbox CSS Grid

The Basic Language of the Web: HTML



Posted by Laura Jones on Monday, June 21st 2027



```
<!-HEADER BOXED FONT WHITE TRANSPARENT-->
<div class="header-black-pg"></div>
<!-NEED FOR TRANSPARENT HEADER-->
<header id="nav" class="header header-1 header-transparent">
<!-FEATURES 7 HALF IMG-->
<div class="page-section bg-gray-light clearfix">
  <div>
    <div class="feat-img-cont col-4-5">
      <div class="feat7-img" style="background-image: url('https://...');"></div>
    </div>
    <div class="container"></div>
  </div>
  <div>
```

All modern websites and web applications are built using three fundamental technologies: HTML, CSS and JavaScript. These are:

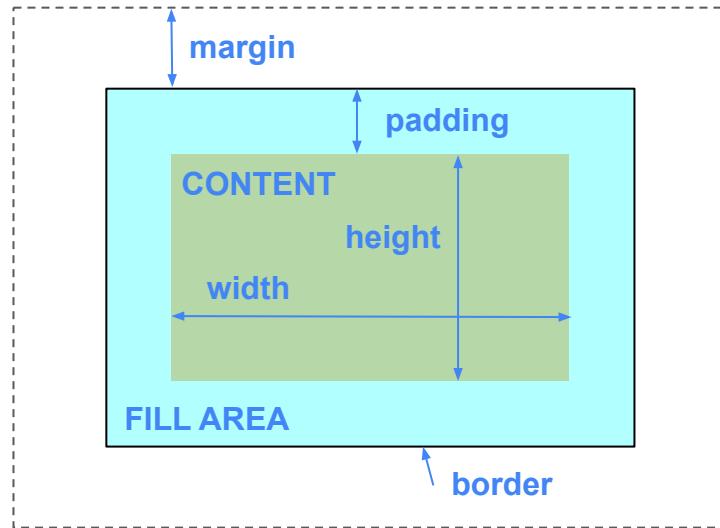
In this post, let's focus on HTML. We will learn what HTML is all about, and why you too should learn it.

WHAT IS HTML?



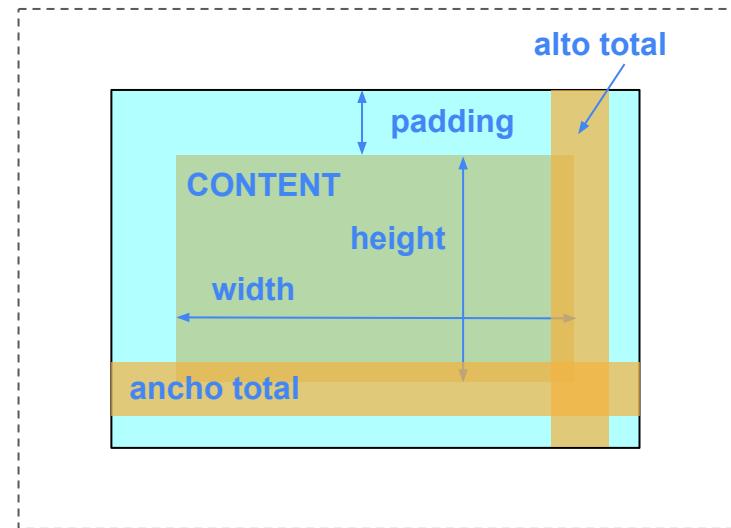
Modelo de caja

- **Content:** Texto, imágenes, etc...
- **Border:** Línea que bordea el elemento.
Se considera **DENTRO** del elemento
- **Padding:** Espacio invisible en torno al contenido. **DENTRO** del elemento
- **Margin:** Espacio invisible **FUERA** del elemento. Espacio entre elementos
- **Fill area:** Área que ocupan las imágenes o a la que aplican propiedades como **background-color**



Modelo de caja

- **Ancho final de un elemento:** left border + left padding + width + right padding + right border
 - **Alto final de un elemento:** top border + top padding + height + bottom padding + bottom border
-
- Estos valores se pueden especificar con propiedades css
 - Los cálculos son por defecto, pero se puede modificar



Modelo de caja

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Ejemplos de clase</title>

    <link rel="stylesheet" href="css/styles.css" />
  </head>
  <body>
    <div class="container">
      <div class="box"></div>
    </div>
  </body>
</html>
```

styles.css

```
.container {
  width: 25%;
  margin: 25% auto;
}

.box {
  background-color: lightpink;
  background-image: url(../assets/imgs/univcat.jpg);
  background-size: 400px 200px;
  background-repeat: no-repeat;
  background-position: center;
  border: 10px solid lightcoral;
  height: 200px;
  padding: 10px;
}
```

- Alto contenido = 200px

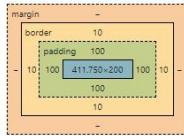
- Alto caja = 200px

- Alto FINAL = ¿200px?

420px!!!!!!

```
.box {
  background-color: lightpink;
  background-image: url(../assets/imgs/univcat.jpg);
  background-size: 400px 200px;
  background-repeat: no-repeat;
  background-position: center;
  border: 10px solid lightcoral;
  height: 200px;
  padding: 10px;
}

div {
  display: block;
}
```



Modelo de caja

```
index.html


<p>Posted by <strong>Laura Jones</strong> on Monday, June 21st 2027</p>


</header>
```

- Si se declaran dimensiones de imagen en HTML y en CSS sólo se declara una de ellas se puede romper la relación de aspecto
 - Se declaran dimensiones sólo en CSS
 - Se declaran dimensiones sólo en HTML
 - O se declaran en HTML y en CSS pero entonces se establece una de ellas a **auto**

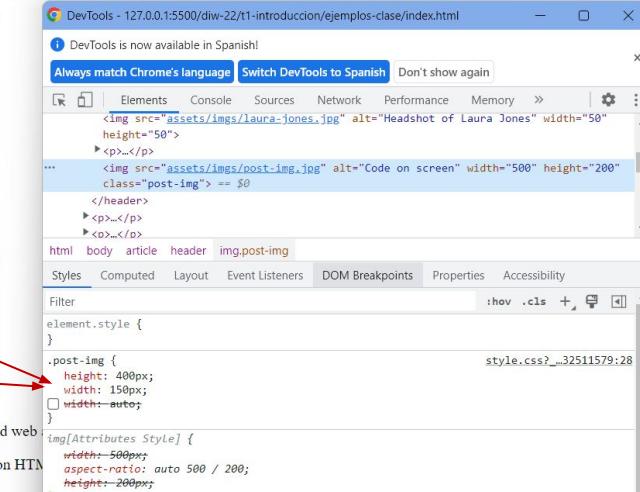


Posted by **Laura Jones** on Monday, June 21st 2027



```
styles.css
.post-img {
  height: 400px;
  width: 150px;
  /*width: auto;*/
}
```

All modern websites and web
In this post, let's focus on HTM



DevTools - 127.0.0.1:5500/di2-22/t1-introducción/ejemplos-clase/index.html

Always match Chrome's language Switch DevTools to Spanish Don't show again

Elements Console Sources Network Performance Memory

```

<p>_</p>

</header>
<p>_</p>
<p>_</p>
```

html body article header img.post-img

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter element.style { }

```
.post-img {
  height: 400px;
  width: 150px;
  /*width: auto;*/
}
```

style.css?_32511579:28

img[Attributes Style] {
 width: 500px;
 height: 200px;
 aspect-ratio: auto 500 / 200;
}

Modelo de caja

```
index.html

<title>The Basic Language of the Web: HTML</title>

<link rel="stylesheet" href="css/style.css" />
</head>
<body>
  <div class="container">
    <header class="main-header">
      <h1>■ The Code Magazine</h1>

      <nav>
        <a href="#" target="_blank">Blog</a>
        <a href="#" target="_blank">Challenges</a>
        <a href="#" target="_blank">Flexbox</a>
        <a href="#" target="_blank">CSS Grid</a>
      </nav>
    </header>

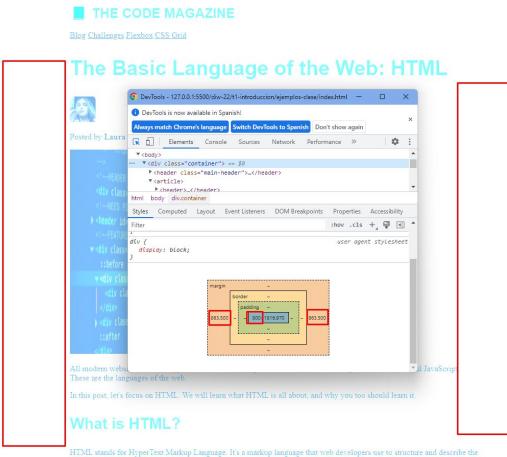
    <article>
      <header>
        <h2>The Basic Language of the Web: HTML</h2>
      </header>
```

```
styles.css

.container {
  width: 800px;
  margin: 0 auto;
}

...
```

- Cómo centrar el contenido en la página
 - Metemos contenido en contenedor seleccionable
 - Fijamos ancho del contenedor
 - **Fijamos márgenes verticales a 0**
 - **Fijamos márgenes horizontales a auto (los calcula el navegador)**



Modelo de caja

- **Inline boxes**

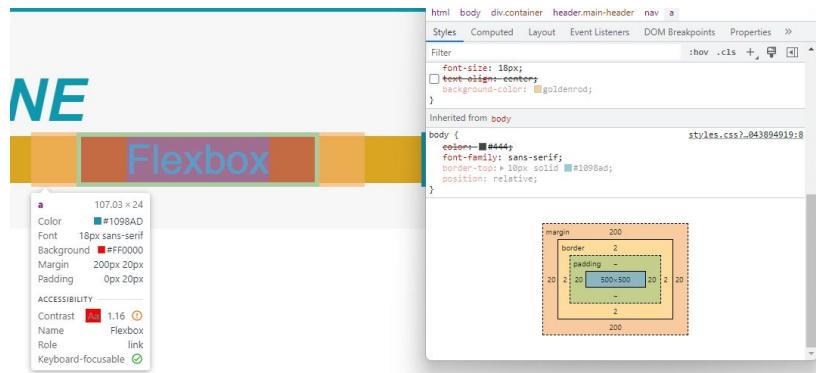
- Ocupa el espacio de su contenido
- Sin saltos de línea
- No aplica ni width ni height
- Paddings y margins sólo horizontales

Elementos: a, img, strong, em, button, etc.

display: inline

```
nav a:link {
    width: 500px;
    height: 500px;

    margin: 200px 20px;
    padding: 0px 20px;
    border: 2px solid black;
    background-color: red;
    color: black;
}
```



Modelo de caja

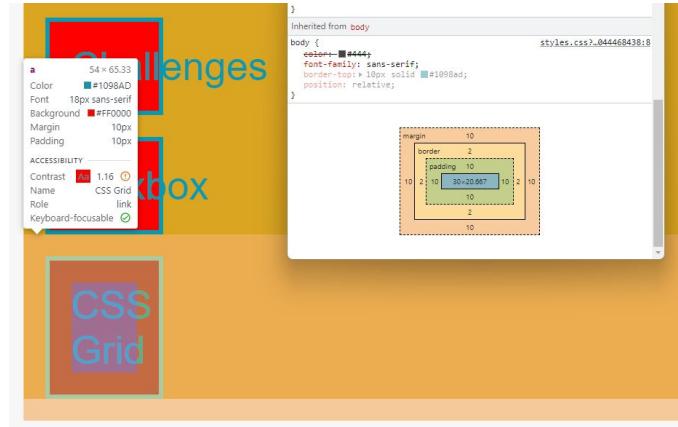
- Block Boxes

- Formateo visual en cajas
- 100% del ancho del padre
- Verticalmente se apilan
- El box-model aplica normalmente

Elementos: body, main, header, footer, section, nav, aside, div, h1-h6, p, ul, ol, li, etc

display: block

```
nav a:link {
    display: block;
    height: 50px;
    width: 30px;
    margin: 10px 10px;
    padding: 10px 10px;
    border: 2px solid black;
    background-color: red;
    color: black;
}
```



Modelo de caja

- **Inline-block boxes**
 - Parece de tipo inline pero se comporta como el tipo block
 - Solo ocupa el ancho de su contenido
 - No produce saltos de línea
 - El box-model aplica normalmente

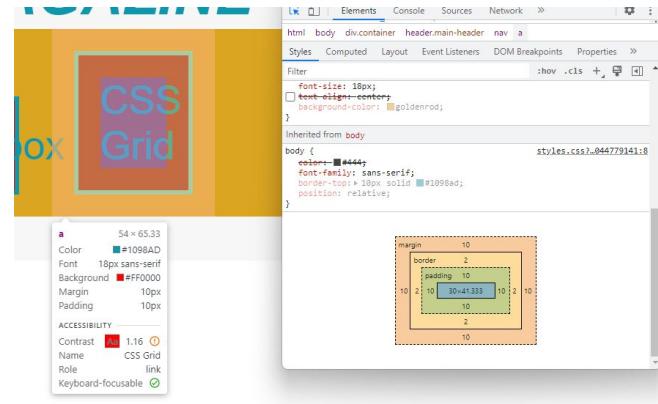
`display: inline-block`

```
nav a:link {
    display: block;

    height: 50px;
    width: 30px;

    margin: 10px 10px;
    padding: 10px 10px;

    border: 2px solid black;
    background-color: red;
    color: black;
}
```



Posicionamiento absoluto

- Flujo normal

- Posicionamiento por defecto
- El elemento está “en el flujo”
- Los elementos se renderizan de acuerdo a su tipo (bloque, en línea...)
- y su orden en el documento HTML

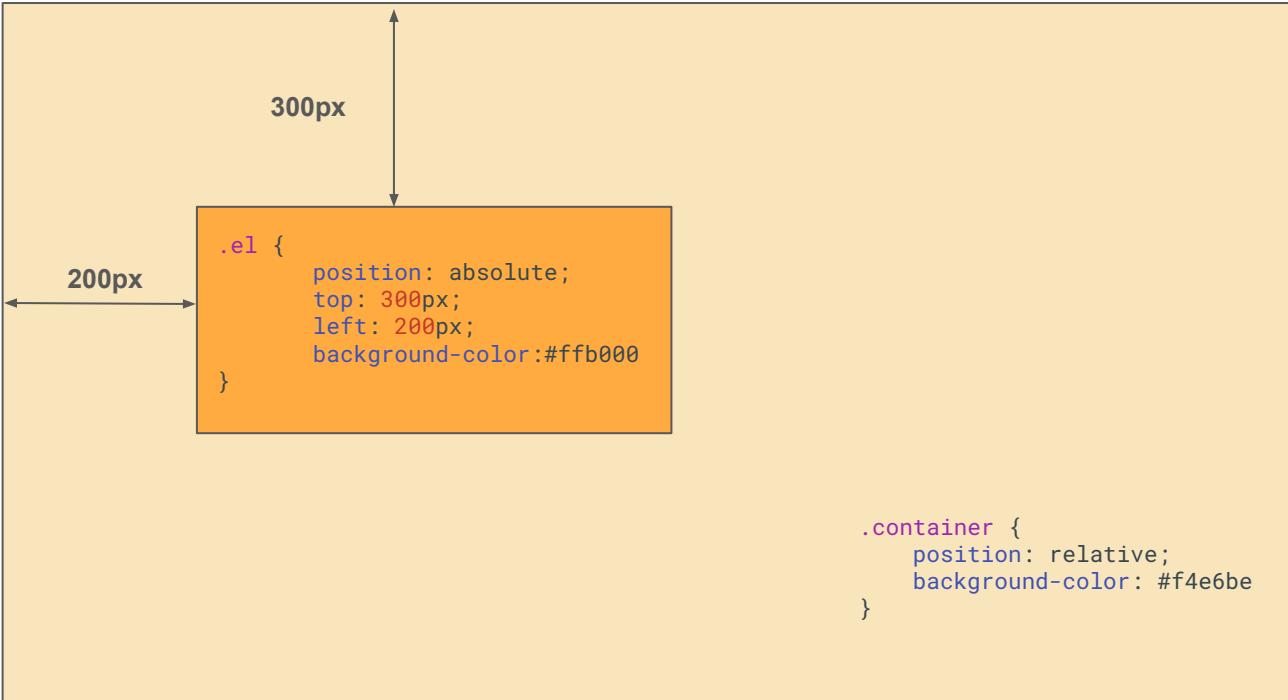
```
position: static
```

- Posicionamiento absoluto

- Los elementos se extraen del flujo normal
- Elemento “fuera del flujo”
- Los elementos que lo rodean no sufren ningún efecto o distorsión, simplemente lo superponen
- Se establece un *offset* desde el *top*, *bottom*, *left* y *right* del primer parente posicionado como relativo

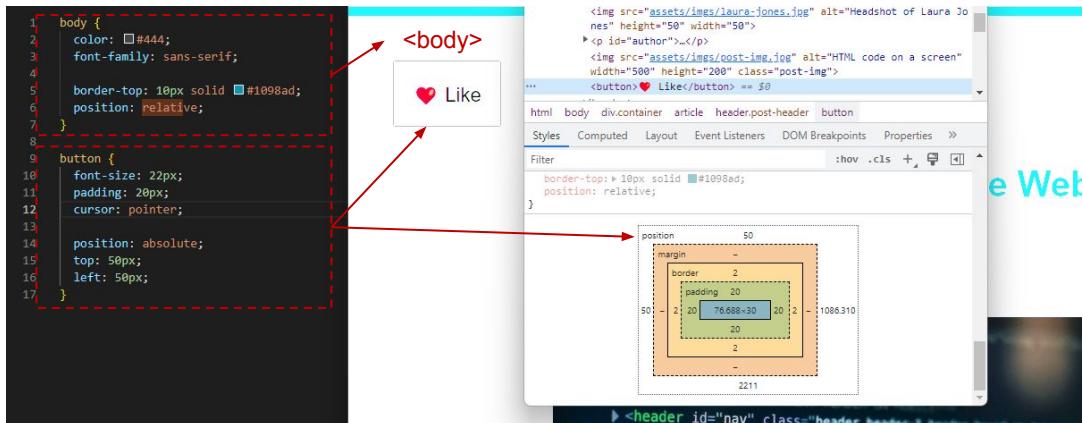
```
position: absolute
```

Posicionamiento absoluto



Posicionamiento absoluto

- Posicionamiento absoluto
 - Sacamos el botón del “flujo normal” con ***position: absolute***
 - Establecemos el **body** como su referencia al marcarlo con ***position: relative***
 - Posicionamos el botón 50px a la derecha del borde izquierdo del **body**
 - Posicionamos el botón 50px hacia abajo del borde superior del **body**



Posicionamiento absoluto

- Posicionamiento absoluto
 - Si hacemos scroll hacia abajo “perdemos” el botón, se mantiene en su posición fija respecto del **body**
 - Si reducimos el ancho de la pantalla, el botón, que recordemos está “fuera del flujo”, se superpone al resto de elementos
 - En el caso en que el elemento con posicionamiento relativo no tenga ningún parente con la propiedad **position: absolute** toma como marco de referencia la ventana gráfica



Pseudo-elementos

- Selector CSS que permite aplicar estilos a “subconjuntos” o “partes” de un elemento HTML.

```
h2::first-letter {
  color: red;
  font-size: 100px;
  font-weight: bold;
  font-style: oblique;

  margin-right: 50px;
}
```

```
p::first-line {
  color: red;
  font-size: 25px;
  font-weight: bold;
}
```

```
h3 + p::first-line {
  color: red;
  font-size: 25px;
  font-weight: bold;
}
```

The Basic HTML



Posted by **Laura Jones** on Monday,

All modern websites and web applications are built using three fundamental technologies: HTML, CSS and JavaScript. These are the languages of the web.

In this post, let's focus on HTML. We will learn what HTML is all about, and why you too should learn it.

What is HTML?

HTML stands for HyperText Markup Language. It's a markup language that web developers use to structure and describe the content of a webpage (not a programming language).

HTML consists of elements that describe different types of content: paragraphs, links, headings, images, video, etc. Web browsers understand HTML and render HTML code as websites.

In HTML, each element is made up of 3 parts:

All modern websites and web applications are built using three fundamental technologies: HTML, CSS and JavaScript. These are the languages of the web.

In this post, let's focus on HTML. We will learn what HTML is all about, and why you too should learn it.

What is HTML?

HTML stands for HyperText Markup Language. It's a markup language that web developers use to structure and describe the content of a webpage (not a programming language).

HTML consists of elements that describe different types of content: paragraphs, links, headings, images, video, etc. Web browsers understand HTML and render HTML code as websites.

In HTML, each element is made up of 3 parts:

1. The opening tag
2. The closing tag
3. The actual element

You can learn more at MDN Web Docs.

Why should you learn HTML?

There are countless reasons for learning the fundamental language of the web. Here are 5 of them:

- To be able to use the fundamental web technologies



Pseudo-elementos

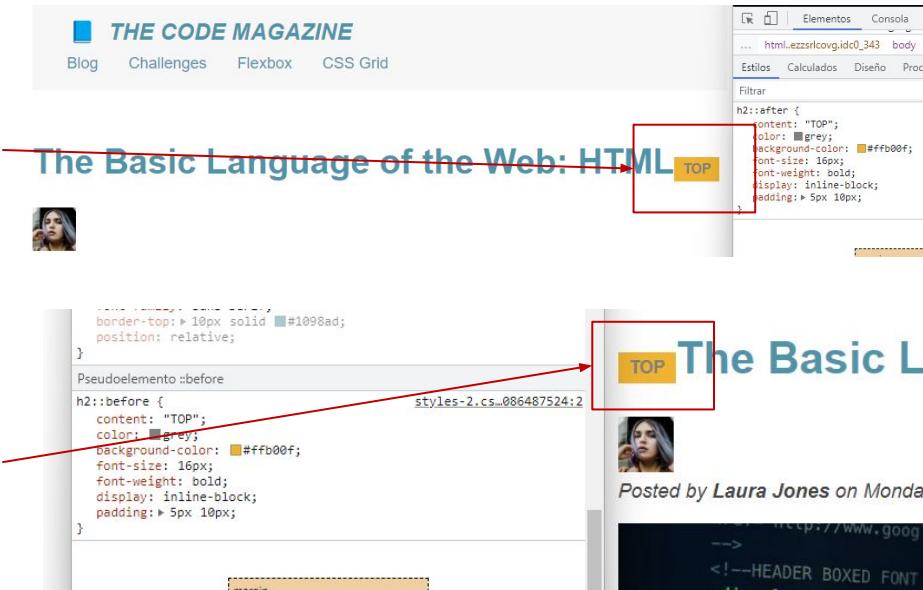
- Los pseudo-elementos **::before** y **::after** permiten añadir contenido “virtual”, no presente en el HTML

```

h2::after {
    content: "TOP";
    color: grey;
    background-color: #ffb00f;
    font-size: 16px;
    font-weight: bold;
    display: inline-block;
    padding: 5px 10px;
}

h2::before {
    content: "TOP";
    color: grey;
    background-color: #ffb00f;
    font-size: 16px;
    font-weight: bold;
    display: inline-block;
    padding: 5px 10px;
}

```

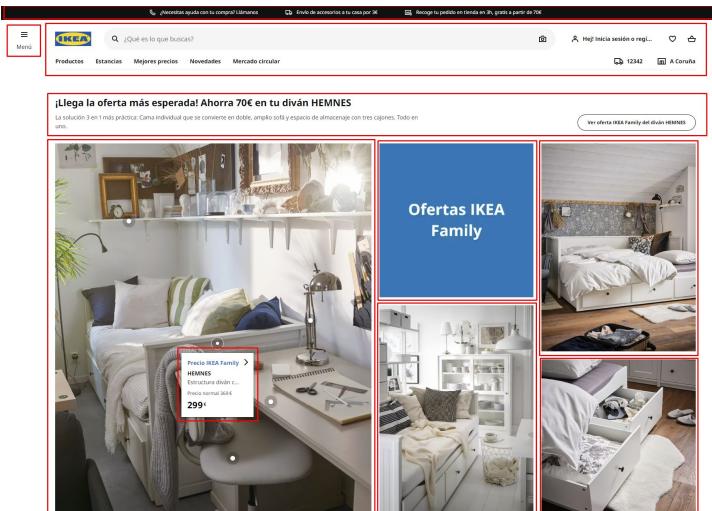


Los 3 modos de construir diseños (Layouts)

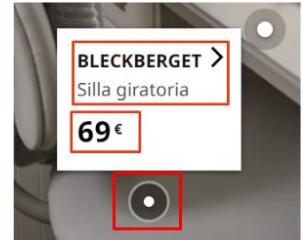
- *Layout* es la forma en que texto, imágenes y otro contenido se dispone y ordena en una página web
- Otorga a la página una estructura visual en la que colocar nuestro contenido
- **Construir un *layout* o diseño:** ordenar elementos de la página en una estructura visual, en lugar de simplemente colocarlos uno detrás de otro (flujo normal)

Los 3 modos de construir diseños (Layouts)

Diseño de página



Diseño de componentes



Los 3 modos de construir diseños (Layouts)

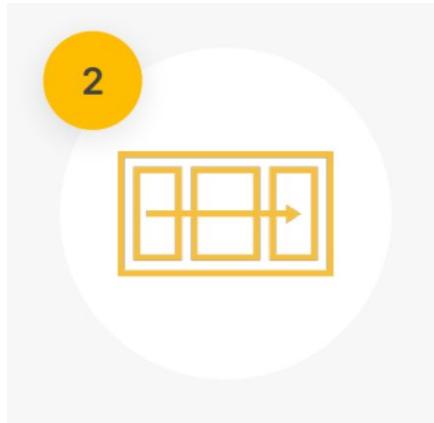


Float Layouts

Modo antiguo de construir diseños de todos los tamaños.

Se basa en la propiedad CSS *float*.

En desuso.



FLEX

Forma actual de estructurar elementos en filas o columnas uni-dimensionales sin usar *floats*.

Perfecto para diseño de componentes.



CSS GRID

Se usa para distribuir elementos en un *grid* o rejilla bi-dimensional.

Ideal para diseños de página y de componentes complejos

Referencias

- Selectores
 - [30 selectores CSS](#) - Referencia con 30 selectores CSS fundamentales para el desarrollo web
 - [CSS Diner](#) - Juego CSS para practicar con selectores
 - [Specificity Smashing Magazine](#) - Artículo detallado sobre especificidad de selectores
- Box model
 - [Fjolt CSS Box Model](#) - Artículo detallado sobre el modelo de caja CSS con ejemplos interactivos
- Pseudo-elementos
 - [Cosas quays hechas con pseudo-elementos](#) - Ejemplos de cosas sorprendentes hechas con pseudo-elementos

Tema 3: CSS Avanzado

Asignatura: Diseño de Interfaces Web



Contenidos

1. [Propiedad Float](#)
2. [Box Model. Border Box](#)
3. [Módulo Flexbox](#)
4. [Uso del Grid CSS](#)
5. [Media queries](#)
6. [Transiciones](#)
7. [Animaciones](#)
8. [Dimensiones CSS](#)
9. [μSeminario - Píxeles CSS](#)
10. [Referencias](#)

Propiedad Float

- Flujo normal

- Posicionamiento por defecto
- El elemento está “en el flujo”
- Los elementos se renderizan de acuerdo a su tipo (bloque, en línea...) y su orden en el documento HTML

- Posicionamiento absoluto

- Los elementos se extraen del flujo normal: “fuera del flujo”
- Los elementos que lo rodean no sufren ningún efecto o distorsión, simplemente lo superponen
- Se establece un *offset* desde el *top*, *bottom*, *left* y *right* del primer parente posicionado como relativo

- Float

- Los elementos se extraen del flujo normal: “fuera del flujo”
- Los elementos que lo rodean “lo envuelven”
- El elemento contenedor no ajustará su tamaño a la altura del elemento flotante, pudiendo “colapsar”

```
position: static
```

```
position: absolute
```

```
float: left  
float: right
```

Propiedad Float

- La imagen flotante de la autora altera la posición de todos los elementos adyacentes, que ahora la “rodean” o “envuelven”

```
.author-img {
  float: left;
}

.author {
  background-color: yellow;
}

.author + p {
  background-color: red;
}
```

The Basic Lang



Posted by **Laura Jones** on M

```
--> http://www.google.com
<!--HEADER BOXED FONT
<div class="header-bl
<!--NEED FOR TRANSPAREN
```

Keyboard-focusable



Posted by **Laura Jones**

Lore ipsum dolor sit amet, consectetur adipisicing elit nisi aliquid ut saepe consequuntur perspiciatis quas aperian eveniet dicta exercitationem, ipsa eligendi ab id accusantium.

The Basic Language of the Web

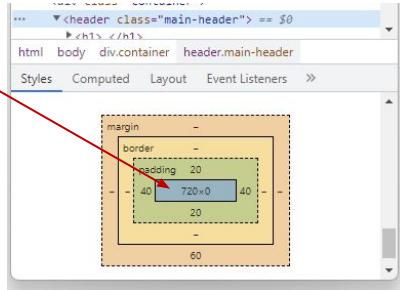


Posted by **Laura Jones** on Monday, June 21st 2027

Lore ipsum dolor sit amet, consectetur adipisicing elit nisi aliquid ut saepe consequuntur perspiciatis quas aperian eveniet dicta exercitationem, ipsa eligendi ab id accusantium.

Propiedad Float

- El título **h1** flota a la izquierda
- El **nav** flota a la derecha
- El elemento **header** que los contiene pierde su altura al quedar sus únicos hijos “fuera del flujo normal”
- El relleno verde se mantiene por el *padding*, pero si lo ponemos a 0, entonces resultará invisible
- Decimos que el elemento **header** “ha colapsado” ¿Cómo podemos resolverlo?



```
.main-header {
  background-color: green;
}

h1 {
  background-color: aquamarine;
  float: left;
}

nav {
  float: right;
  background-color: yellow;
}
```



THE CODE MAGAZINE

Blog

Challenges

Flexbox

CSS Grid

Propiedad Float

- ¿Cómo resolver el colapso de un elemento?
 - a. Añadimos un elemento vacío al padre con la propiedad **clear**
 - b. Creamos un pseudo-elemento con la propiedad **clear**

```
.clearfix::after {  
  clear: both;  
  content: "";  
  display: block;  
}
```



The screenshot shows a dark-themed header for 'THE CODE MAGAZINE'. The header includes a logo, the site name, and navigation links for 'Blog', 'Challenges', 'Flexbox', and 'CSS Grid'. The HTML code on the left shows a `<header>` element with a `clearfix` class applied. The CSS code on the right defines the `.clearfix::after` pseudo-element.

```
<body>  
<div class="container">  
  <header class="main-header clearfix">  
    <h1> THE CODE MAGAZINE </h1>  
    <nav>  
      <a href="#">Blog</a>  
      <a href="#">Challenges</a>  
      <a href="#">Flexbox</a>  
      <a href="#">CSS Grid</a>  
    </nav>  
  </header>  
</div>
```

```
.clearfix::after {  
  clear: both;  
  content: "";  
  display: block;  
}
```

Propiedad Float

- Código inicial disponible en <https://github.com/tcamba-ciclos-montecastelo/diw-22/tree/master/t1-introduccion/css-avanzado/inicial>
- Usando
 - display: block, display:inline-block...
 - pseudo-elementos
 - float
- Desarrollar la versión de la web que se muestra en la siguiente imagen



THE CODE MAGAZINE

The Basic Language of the Web: HTML

TOP

Posted by **Laura Jones** on Monday, June 21st 2027

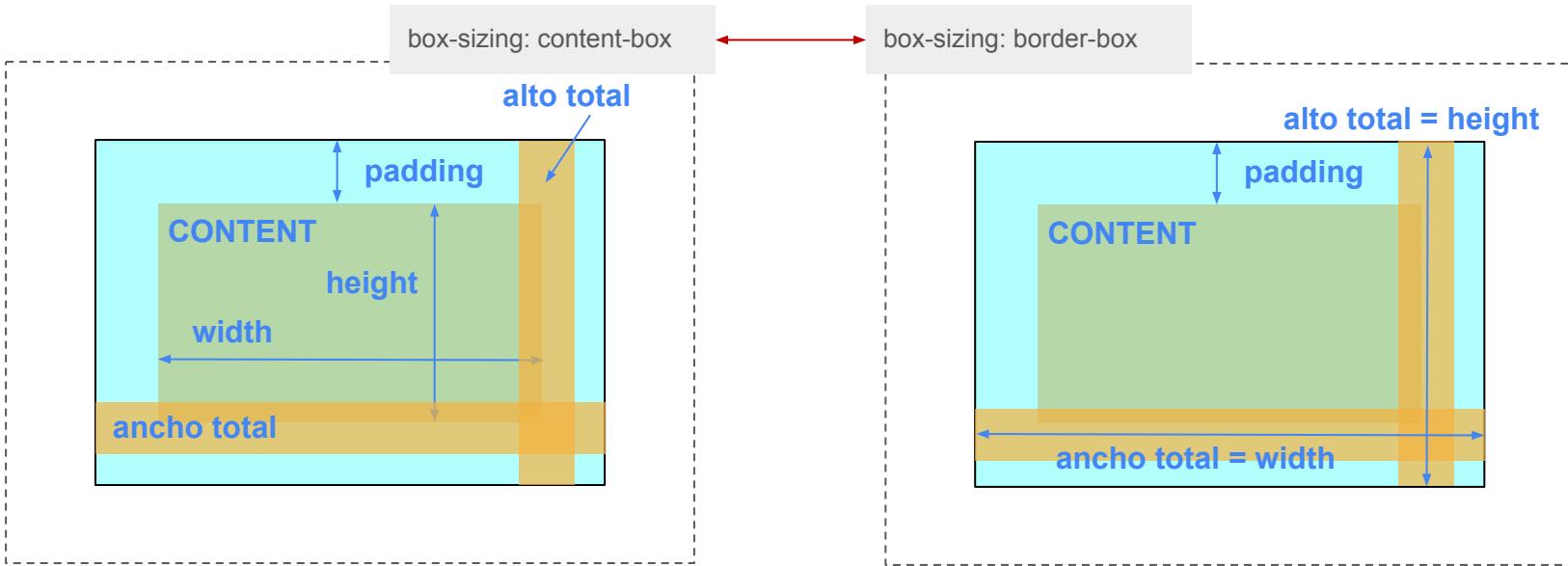
RELATED POSTS

- How to Learn Web Development By Jonas Schmedemann
- The Unknown Powers of CSS By Jim Dillon
- Why JavaScript is Awesome By Matilda

All modern websites and web applications are built using three *fundamental* technologies: HTML, CSS and JavaScript. These are the languages of the web.

In this post, let's focus on HTML. We will learn what HTML is all about, and why you too should learn it.

Box Model. Border Box



- **Ancho final de un elemento:** ~~left border + left padding+ width + right padding + right border~~
- **Alto final de un elemento:** ~~top border + top padding+ height + bottom padding+ bottom border~~

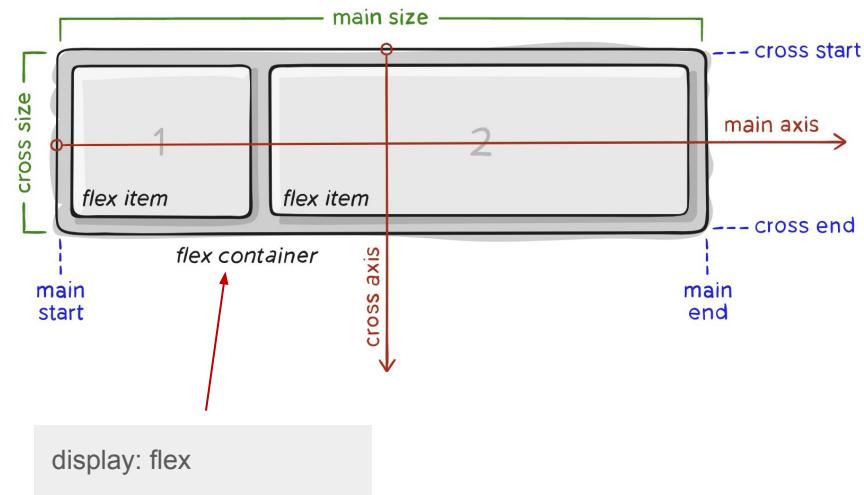
Flexbox

- **Flexbox** es un conjunto de propiedades CSS relacionadas para construir diseños 1-dimensionales
- La idea principal tras **Flexbox** es que el espacio vacío dentro de un elemento contenedor puede ser dividido automáticamente entre sus elementos hijos
- **Flexbox** facilita el alineamiento automático de los elementos de un lado a otro dentro de un contenedor padre, tanto vertical como horizontalmente
- **Flexbox** resuelve problemas comunes como el centrado vertical y la creación de columnas de la misma altura
- **Flexbox** es perfecto para reemplazar floats, permitiéndonos escribir código HTML y CSS más limpio



Flexbox

- **Main axis:** Los **flex-items** se dispondrán siguiendo esta dirección hasta ocupar todo el **main size** del **flex-container**
- **Cross Axis:** Dirección secundaria que dirige el flujo de los **flex-items** cuando ocupan todo el **main size** del **flex-container**
- El sentido / orden que siguen los **flex-items** sobre cada eje viene determinado por los puntos de inicio / fin de cada eje
 - *main start/end*
 - *cross start/end*
- La dirección y sentido de los ejes no tiene porque ser la indicada en la imagen y depende de las propiedades ***flex-direction*** y **writing-mode**



Flexbox

FLEX CONTAINER

1. **gap: 0 | <length>**
 Crea espacio entre ítems, sin usar **margin**
2. **justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly**
 Alinea los ítems a lo largo del *main axis* (horizontalmente, por defecto)
3. **align-items: stretch | flex-start | flex-end | center | baseline**
 Alinea los ítems a lo largo del *cross axis* (verticalmente, por defecto)
4. **flex-direction: row | row-reverse | column | column-reverse**
 Define cuál es el *main axis*
5. **flex-wrap: nowrap | wrap | wrap-reverse**
 Permite a los ítems saltar de línea si son demasiado grandes
6. **align-content: stretch | flex-start | flex-end | center | space-between | space-around**
 Solo aplica cuando hay varias líneas (**flex-wrap: wrap**)

FLEX ITEMS

1. **align-self: auto | stretch | flex-start | flex-end | center | baseline**
 Sobreescribe **align-items** a nivel de flex-item
2. **flex-grow: 0 | <integer>**
 Permite a los ítems crecer (0 NO permite, 1+ SI permite)
3. **flex-shrink: 1 | <integer>**
 Permite a los ítems encogerse (0 NO permite, 1+ SI permite)
4. **flex-basis: auto | <length>**
 Para definir el ancho de un ítem, sin usar **width**
5. **flex: 0 1 auto | <int> <int> <len>**
 Atajo **recomendado** para *flex-grow*, *-shrink*, *-basis*
6. **order: 1 | <integer>**
 Controla el orden de los items, -1 lo pone el primero, 1 el último

Flexbox

- A partir del código disponible aquí
<https://github.com/tcamba-ciclos-montecastelo/diw-22/tree/master/t1-introduccion/css-avanzado/inicial>
- Usando
 - flexbox
- Desarrollar la versión de la web que se muestra en la siguiente imagen



THE CODE MAGAZINE

The Basic Language of the Web: HTML

TOP

Posted by Laura Jones on Monday, June 21st 2027

```
<!--HEADER BOXED FONT WHITE TRANSPARENT-->
<div class="header-black-bg"></div>
<!--NEED FOR TRANSPARENT HEADERS ON MOBILE-->
<header id="nav" class="header header-1 header-transparent">
<!--FEATURES 7 HALF IMG-->
<div class="page-section bg-gray-light clearFix">
  :before
  <div class="fes7-img-cont col-md-9">
    <div class="fes7-img" style="background-image: url('https://www.google.es/cse/img/nav/nav_bg_1.jpg');"></div>
  </div>
  <div class="container"></div>
  :after
</div>
```

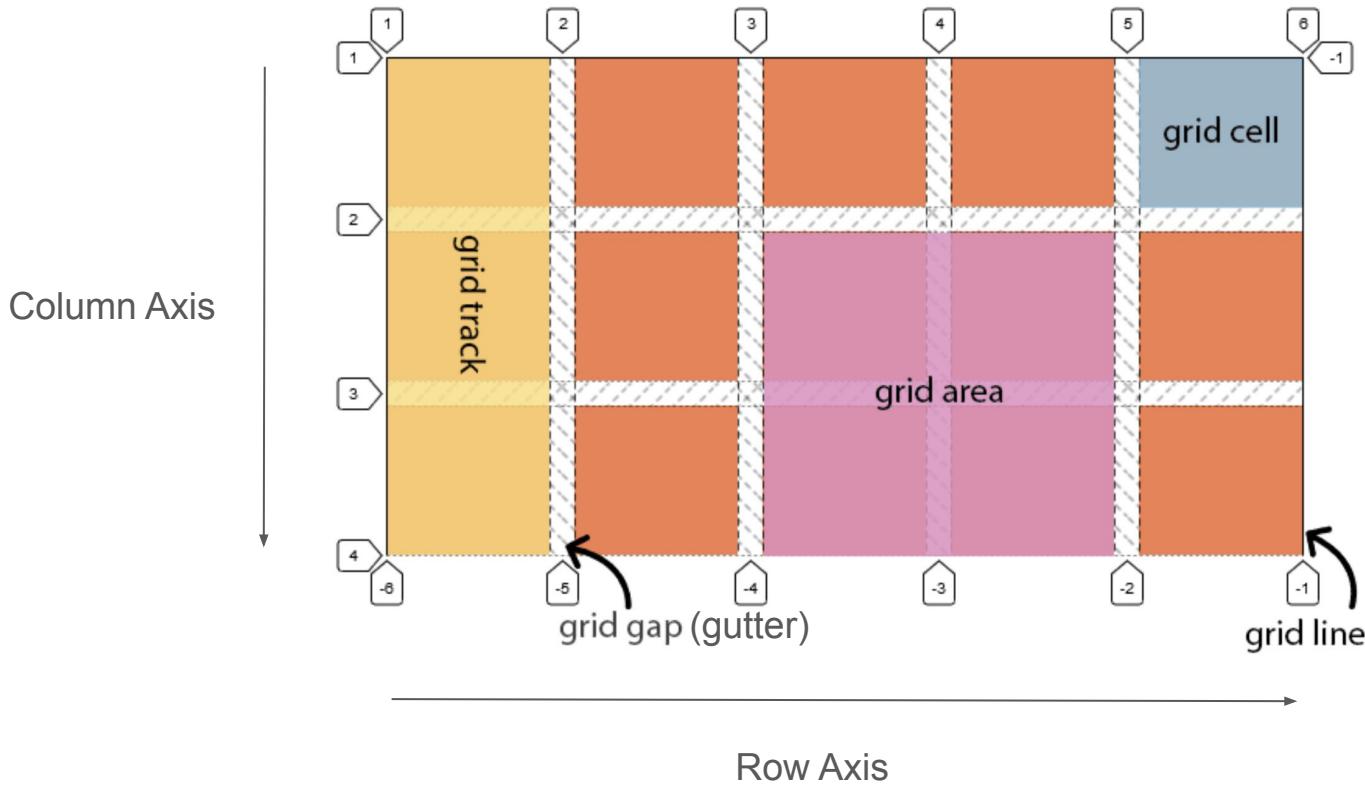
RELATED POSTS

- How to Learn Web Development By Jonas Schmedemann
- The Unknown Powers of CSS By Jim Dillon
- Why JavaScript is Awesome By Matilda

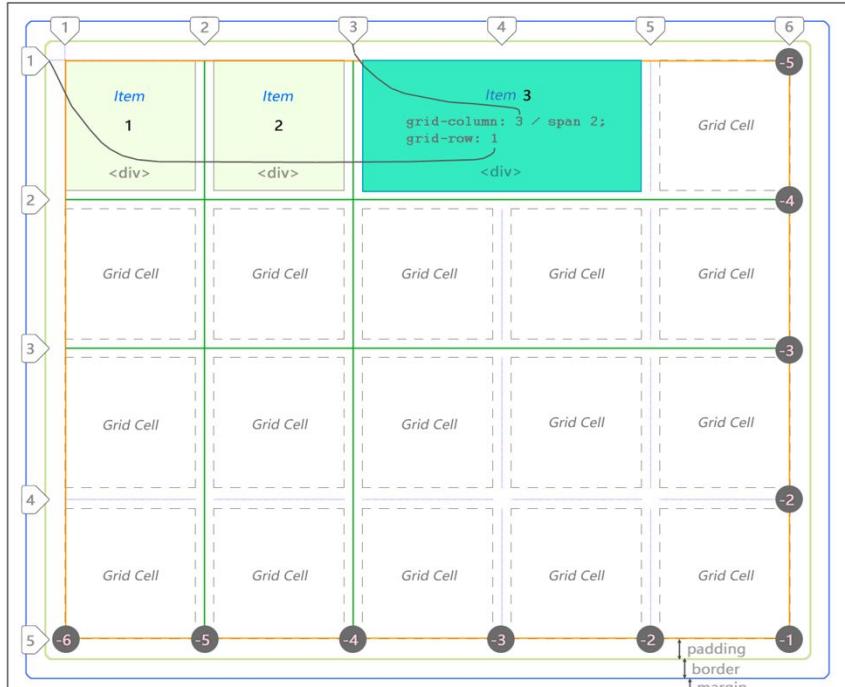
CSS Grid

- CSS Grid es un conjunto de propiedades CSS para construir diseños 2-dimensionales
- La idea principal tras CSS Grid es la división de un elemento contenedor en filas y columnas que pueden llenarse con elementos hijos
- En contextos 2-dimensionales, CSS Grid nos permite escribir menos código HTML anidado y CSS más fácil de leer
- ¡CSS Grid no es un sustituto de Flexbox! Es más, funcionan muy bien juntos

CSS Grid



CSS Grid

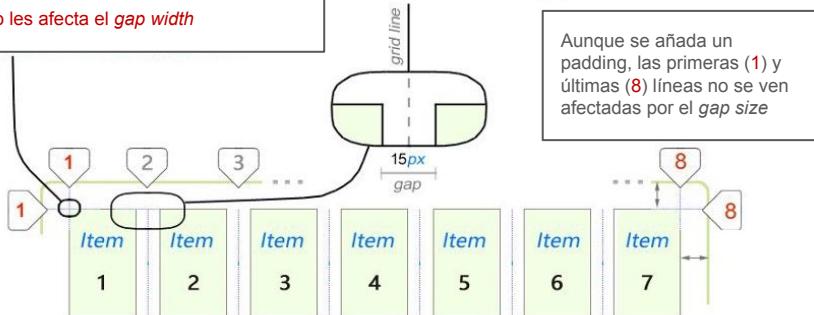


Las líneas del Grid marcan los límites de cada *grid-track* y sirven para posicionar los ítems en el grid

Por defecto el Grid no tiene bordes.

Las líneas 1 vertical y 1 transversal comienzan en la esquina superior derecha del primer elemento

No les afecta el *gap width*



Aunque se añada un padding, las primeras (1) y últimas (8) líneas no se ven afectadas por el *gap size*

CSS Grid



Posicionamiento implícito

- Los ítems 3 y 4 se posicionan automáticamente en la 2da fila (en la primera no queda espacio)
- Los ítems ajustan su ancho para ajustarse a lo declarado en `grid-template-columns`



Posicionamiento explícito

- Grid definido de forma explícita : 2×2 cells
- Los ítems se ajustan a las dimensiones declaradas en `grid-template-columns` y `grid-template-rows`

CSS Grid

auto

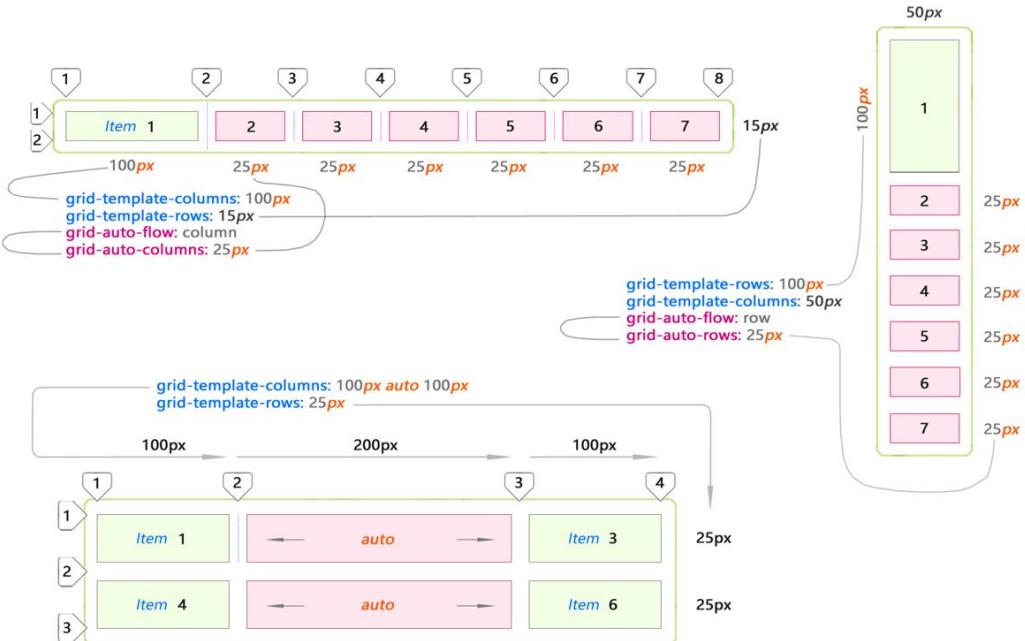
El valor *auto* en las propiedades *grid-template-** hace que las celdas afectadas se repartan el tamaño disponible del Grid container

grid-auto-flow: column | row

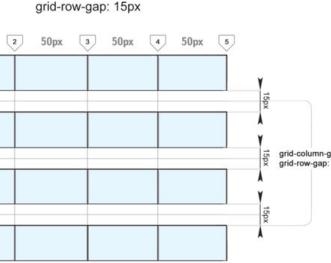
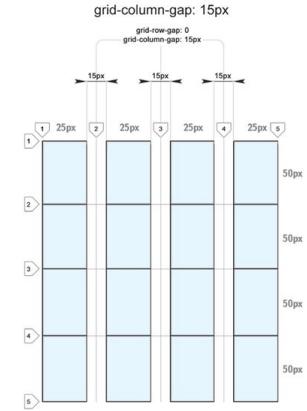
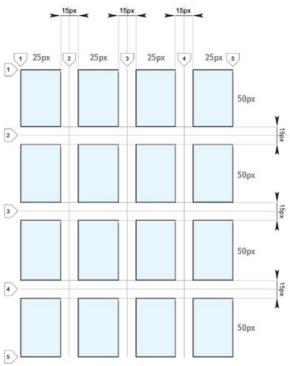
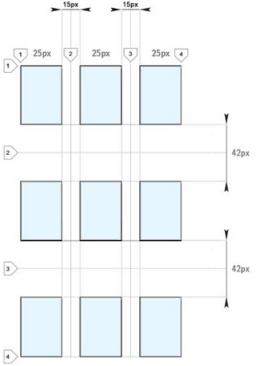
Determina la dirección en la que se posicionan los ítems cuando se quedan sin espacio en el *grid-track* correspondiente

grid-auto-[columns | rows]: size

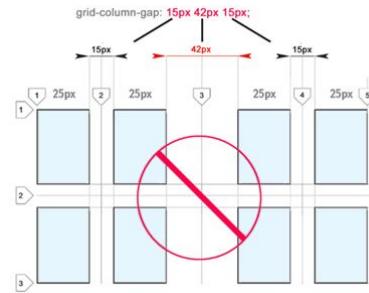
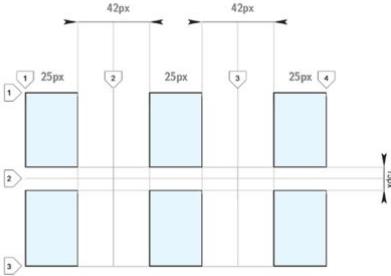
Define el tamaño por defecto de las filas / columnas que se incluyen automáticamente en el Grid



CSS Grid



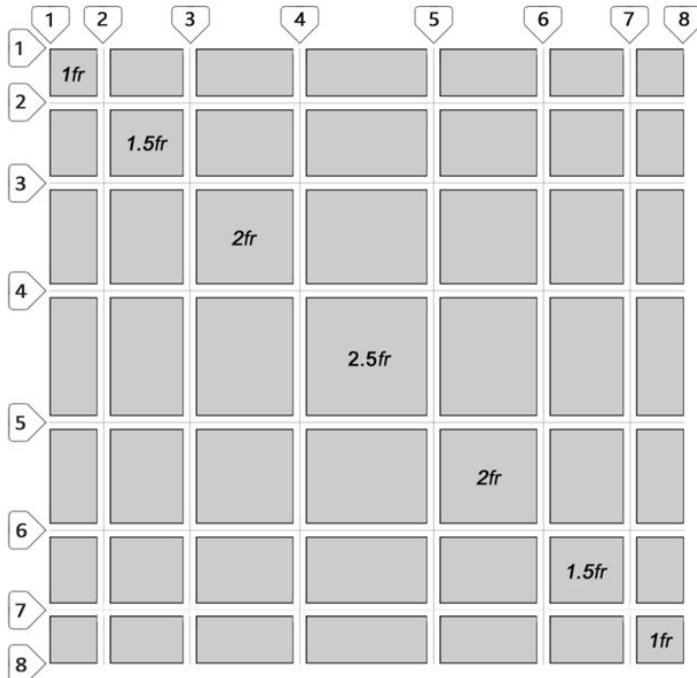
- **NO:** *gaps* diferentes en la misma dirección
- **SI:** *gaps* diferentes en direcciones distintas



CSS Grid

Unidades fraccionarias

- Los tamaños de las celdas se pueden especificar usando **unidades fraccionarias *fr***
- Cada *fr* corresponde a una unidad del espacio disponible en el *grid-container*
- **OJO**, se ve afectado por los *gaps*
- Es habitual acortar la declaración de filas / columnas repetidas usando la función ***repeat***
 - *grid-template-columns: 100px 1fr 1fr 1fr*
 - *grid-template-columns: 100px repeat(3, 1fr)*



grid-template-rows: 1fr 1fr 1.5fr 2.0fr 2.5fr 2.0fr 1.5fr 1fr
grid-template-columns: 1fr 1fr 1.5fr 2.0fr 2.5fr 2.0fr 1.5fr 1fr



CSS Grid

GRID CONTAINER

1. grid-template-rows: <track size>*
grid template-columns: <track size>*

Establece los **row y column tracks** del grid. Una unidad de longitud por cada track. Se puede usar cualquier unidad pero **fr (fraction)** rellena el espacio sin usar.

2. row-gap: 0 | <length>
column-gap: 0 | <length>

Crea espacio vacío entre *tracks*

3. justify-items: stretch | start | center | end
align-items: stretch | start | center | end

Alinea los *items* dentro de las filas / columnas (horizontal / verticalmente)

4. justify-content: stretch | start | center | end | ...
align-content: stretch | start | center | end | ...

Alinea el **grid entero dentro del grid container**. Solo aplica si el *grid container* es más grande que el grid

GRID ITEMS

1. grid-column: <start line> / <end line> | span <number>
grid-row: <start line> / <end line> | span <number>

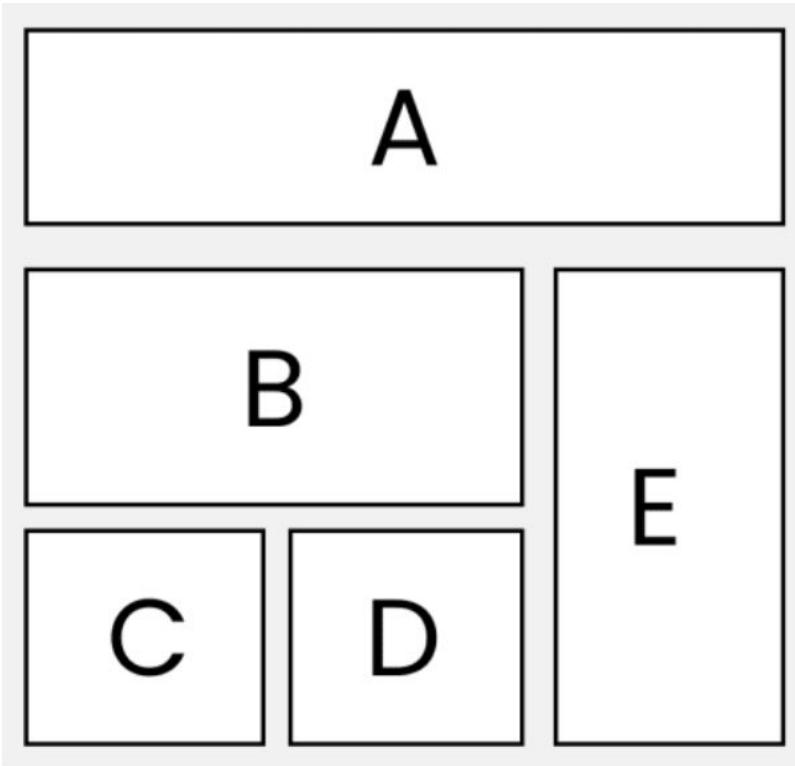
Coloca el ítem en una celda específica, basado en los números de línea. La palabra clave **span** se puede usar para expandir un ítem a lo largo de más celdas.

2. justify-self: stretch | start | center | end
align-self: stretch | start | center | end

Sobreescribe justify-items / align-items para un elemento específico

CSS Grid

- Usando
 - CSS Grid
- Crea el siguiente diseño pensado para pantallas grandes

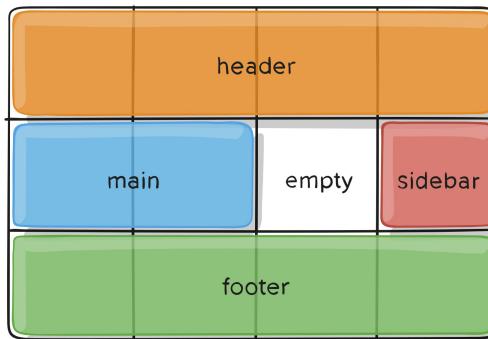


CSS Grid

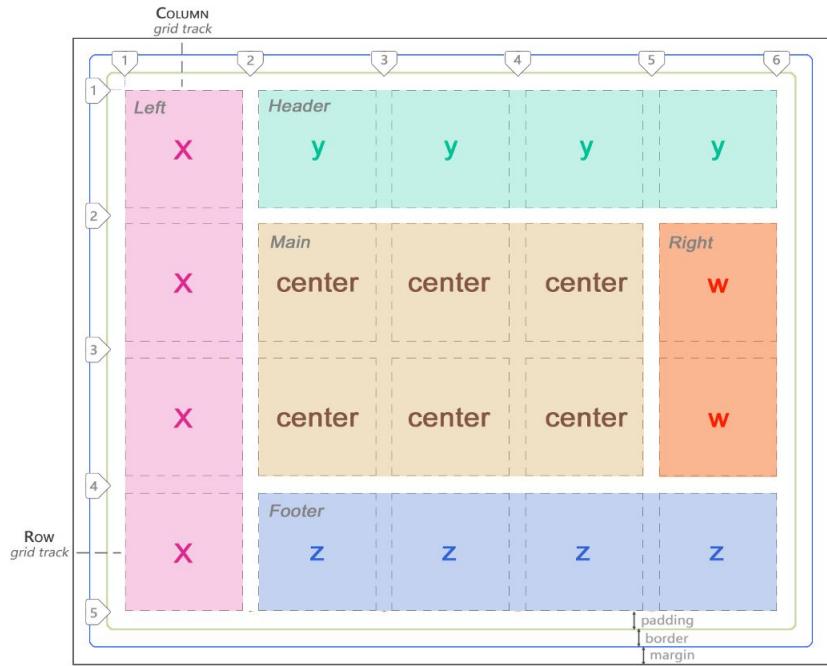
Grid Areas

- Permite definir un Grid a partir de nombres o etiquetas que representan sus celdas
- Un área se compone de una o varias de estas etiquetas contiguas
- Un “.” representa una celda vacía
- Todas las columnas deben contener el mismo número de celdas

```
.container {  
    display: grid;  
    grid-template-columns: 50px 50px 50px 50px;  
    grid-template-rows: auto;  
    grid-template-areas:  
        "header header header header"  
        "main main . sidebar"  
        "footer footer footer footer";
```



CSS Grid



```

1 <head>
2   <style type = "text/css">
3     #grid { grid-template-areas:
4       'x y y y y'
5       'x center center center w'
6       'x center center center w'
7       'x z z z z';
8     }
9   </style>
10 </head>
11 <body>
12   <div id = "grid">
13     <div style = "grid-area: x">Left</div>
14     <div style = "grid-area: y">Header</div>
15     <div style = "grid-area: z">Footer</div>
16     <div style = "grid-area: w">Right</div>
17     <div style = "grid-area: center">Main</div>
18   </div>
19 </body>
20

```

CSS Grid

- A partir del código disponible aquí
<https://github.com/tcamba-ciclos-montecastelo/diw-22/tree/master/t1-introduccion/css-avanzado/inicial>
- Usando
 - CSS Grid Template Areas
- Desarrollar la versión de la web que se muestra en la siguiente imagen



The screenshot shows a blog post titled "The Basic Language of the Web: HTML" by Laura Jones. The post content discusses the fundamental technologies of the web: HTML, CSS, and JavaScript. It highlights that modern websites and web applications are built using these three technologies. The sidebar on the right lists "RELATED POSTS" with titles like "How to Learn Web Development", "The Unknown Powers of CSS", and "Why JavaScript is Awesome". Navigation links at the top include "Blog", "Challenges", "Flexbox", and "CSS Grid".

Media queries

- Permiten incluir o aplicar un bloque de reglas CSS sólo si una determinada condición es cierta
- Aplican estilos para un tipo de medios concreto (pantalla, impreso, ...)
- ... si se verifica una condición sobre una propiedad del dispositivo (generalmente su altura / anchura mínima /máxima)

```
@media not | only mediatype and | not | only (media feature) {  
    selector {...}  
}
```



Media queries

```
Media Type           Condición a testear
↓                         ↗
@media screen and (max-width: 525px) {
    img[class="hide"] {
        display: none !important
    }
}
```

“En pantallas con anchos inferiores a 525 px oculta las imágenes con la clase *hide*”

Media queries

Tipo de medio	Significado
screen	Monitores o pantallas de ordenador
print	Documentos de medios impresos o pantallas de previsualización de impresión
speech	Lectores de texto para invidentes
all	Todos los dispositivos o medios. Por defecto

Media queries

Tipo de propiedad	Valores	¿Cuándo se aplica?
width	size	Si el dispositivo tiene el tamaño indicado exactamente
min-width	size	Si el dispositivo tiene un tamaño de ancho mayor al indicado
max-width	size	Si el dispositivo tiene un tamaño de ancho menor al indicado
aspect-ratio	aspect-ratio	Si el dispositivo encaja con la relación de aspecto indicada
orientation	landscape portrait	Si el dispositivo está orientado vertical u horizontalmente

Media queries

Se usa solo para que navegadores antiguos no procesen media query

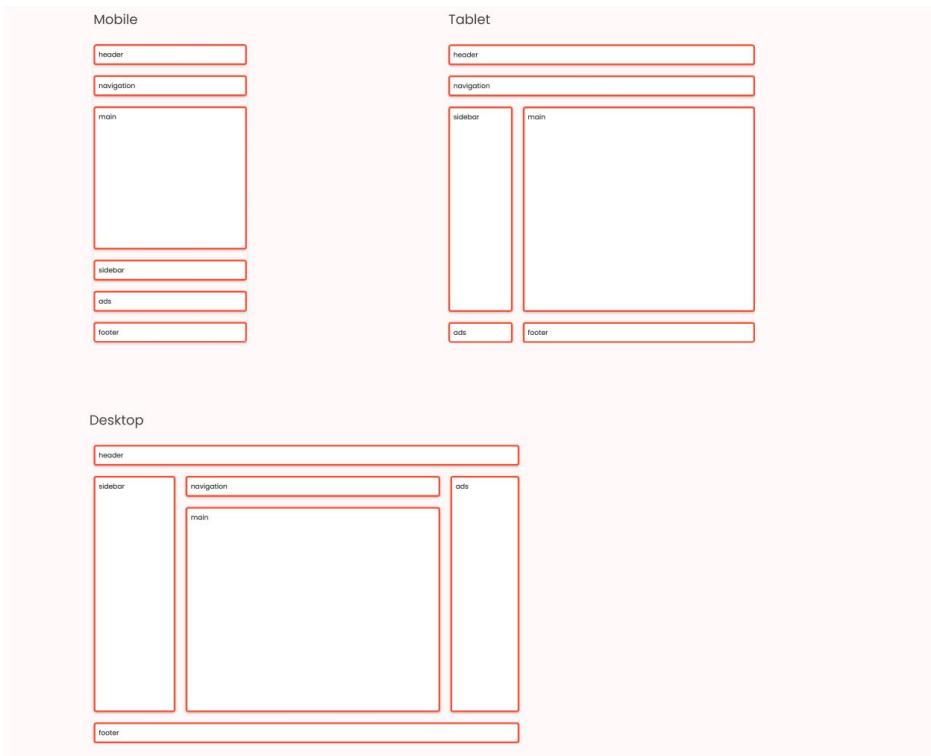
“En pantallas con anchos inferiores a 400 px y superiores a 800 px...”

```
@media only screen and not max-width:800px and min-width:400px {  
    ...  
}
```

Niega la query **completa**, no solo una de las condiciones

Media queries

- Usando
 - MQ
- Crea el siguiente diseño **responsive**
 - Móvil: **width < 720px**
 - Tablet: **720 px < width < 1020px**
 - Escritorio: **width > 1020px**



Transiciones

- Permiten “animar” cambios en propiedades CSS de forma que se produzcan de forma paulatina a lo largo de un intervalo de tiempo
 - **transition-property**: Lista separada por comas de propiedades CSS a las que deberían aplicarse las transiciones.
 - **transition-duration**: Duración de las transiciones especificada en segundos o milisegundos. Puede indicarse una duración global para todas las propiedades o una lista separada por comas con la duración para cada propiedad de la transición (se aplica en orden).
 - **transition-timing-function**: Especifica la “forma” en la que se calculan los valores intermedios de las propiedades que transicionan.
 - **transition-delay**: Define el tiempo de espera entre el momento en que se cambia una propiedad y el inicio de la transición.

Transiciones

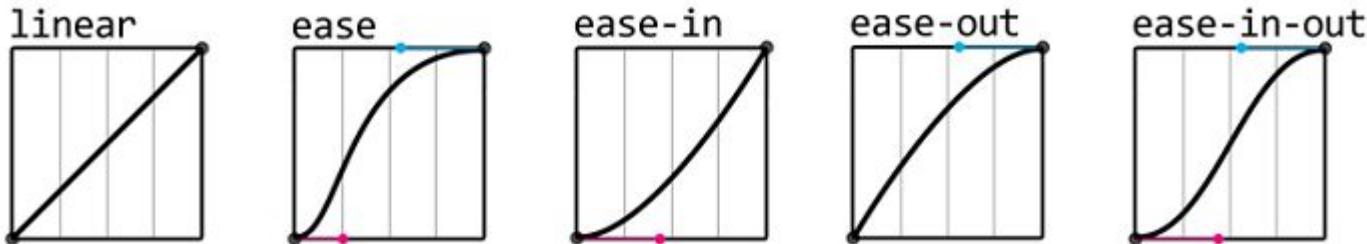
Hay una larga lista de propiedades admitidas por **transition-property** y por tanto transicionables. Las más usadas son las siguientes:

- **background-[color | image | position | size]**: P.ej. para animar cambios de imagen de fondo
- **border-[color | radius | width | spacing]**: Transición de propiedades de los bordes
- **opacity** y **visibility**: Se pueden usar para crear un “fade” o fundido, en el que un elemento aparece o desaparece poco a poco
- **width** y **height**: Cambios en los tamaños de los elementos
- **top, right, bottom** y **left**: La transición de estas propiedades de posicionamiento permite crear animaciones de movimiento de los elementos
- **color, font-[size | weight]**: Transiciones en propiedades del texto. Habitual en combinación con pseudoelementos como **:hover**, **:focus** ...

Transiciones

La propiedad **transition-timing-function** toma habitualmente los siguientes valores predefinidos:

- **linear**: El valor de la propiedad cambia proporcionalmente al paso del tiempo
- **ease**: La propiedad cambia lentamente al principio y al final de la transición
- **ease-in**: La transición es más lenta al principio
- **ease-out**: La transición es más lenta al final
- **ease-in-out**: Cambia despacio al inicio y al final de la transición. El resto del tiempo cambia linealmente.



También es posible personalizar esta propiedad detallando los pesos de los nodos de una curva de bézier o una función por pasos, pero se trata de una característica avanzada y poco frecuente que queda fuera del alcance del curso

Transiciones

```
* {
  padding: 0;
  margin: 0;
}

.container {
  height: 75vh;

  display: flex;
  justify-content: center;
  align-items: center;
}

p {
  font-family: monospace;
}

.transitionable {
  font-size: 50px;
  color: lightskyblue;

  transition: font-size 1.5s ease-in,
              color 2.5s ease-out,
              font-weight 3s ease-in-out;
}

.transitionable:hover {
  font-size: 60px;
  color: orangered;
}
```

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Transiciones</title>
</head>
<body>
  <div class="container">
    <p class="transitionable"> Transición al hacer
<em>hover</em> </p>
  </div>
</body>
</html>
```

Transicion al hacer *hover*

Transicion al hacer *hover*

Transicion al hacer *hover*

Animaciones

- Al igual que las transiciones, aplican una transformación a los valores de una o varias propiedades en un intervalo de tiempo dado.
- En las **transiciones** las propiedades cambian entre un valor o estado inicial y otro final.
- En las **animaciones** se pueden definir tantos estados intermedios como se desee. Para ello se utiliza la At-Rule **@keyframes**

```
@keyframes ejemplo-1 {  
    from {color: red;}  
    to {color: yellow;}  
}  
  
@keyframes ejemplo-2 {  
    0%   {color: red;}  
    25%  {color: yellow;}  
    50%  {color: blue;}  
    100% {color: green;}  
}
```

Animaciones

- Propiedades CSS:
 - **animation-name:** Nombre de la animación especificado en `@keyframes`
 - **animation-duration:** Duración de la animación en segundos o milisegundos. Como en transiciones
 - **animation-delay:** Tiempo de retardo inicial de la animación en segs o mseg. Como en transiciones
 - **animation-iteration-count:** Número de veces que se repite la animación. Admite el valor `infinite` para reproducirla en bucle
 - **animation-direction:** Sentido en el que se produce la animación
 - `normal` | `reverse` | `alternate` | `alternate-reverse`
 - **animation-timing-function:** Como cambian los valores intermedios de las propiedades animadas. Como en transiciones

Animaciones

- Propiedades CSS:
 - **animation-fill-mode:** Estilo que aplica al elemento animado cuando la animación no esta funcionando (antes de comenzar, después de terminar o en ambos casos)
 - forwards | backwards | both *revisar
 - **animation-play-state:** Si la animación está funcionando o pausada
 - running | paused
 - **animation:** Atajo con todas las propiedades anteriores
 - duration | easing-function | delay | iteration-count | direction | fill-mode | play-state | name

Animaciones

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <title>Animaciones</title>
</head>
<body>
  <div class=container>
    <div class="square"></div>
    <div class="roof"></div>
  </div>
</body>
</html>
```

```
* {
  margin: 0;
  padding: 0;
}

.container {
  padding: 75px;
  width: 50vw;
}

.square {
  width: 250px;
  height: 250px;

  background-color: lightskyblue;
  border: 1px solid grey;

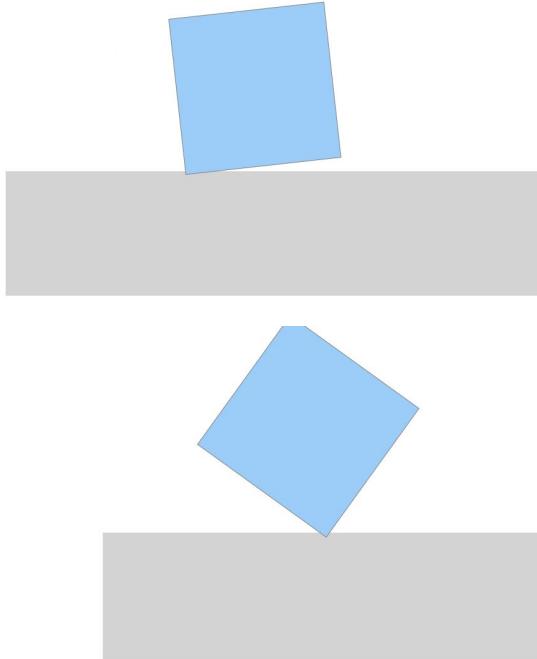
  animation-name: move;
  animation-duration: 5s;
  animation-iteration-count: infinite;
  animation-timing-function: ease-in;
  animation-fill-mode: none;
}

.roof {
  border-top: 5px;
  background-color: lightgrey;

  height: 200px;
}
```

La mitad del viewport

Cambios bruscos al final de cada transición

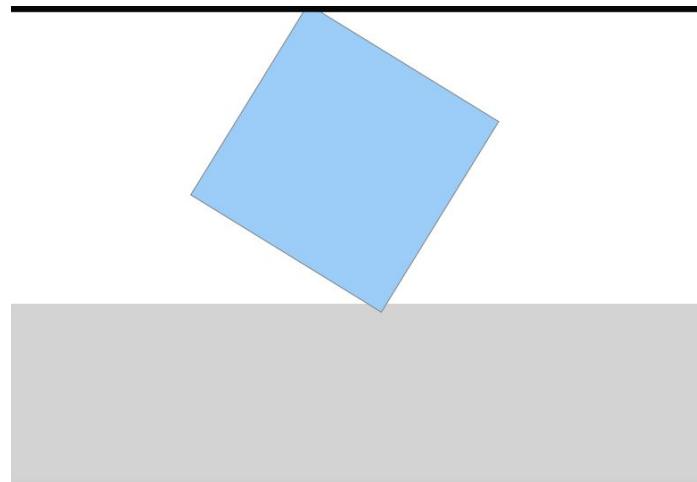


Animaciones

```
@keyframes move {
  0% {
    transform: translate(0px, 0px) rotateZ(0deg);
  }
  10% {
    transform: translate(71.21px, -53px) rotateZ(45deg);
  }
  20% {
    transform: translate(calc(2*71.21px), 0px) rotateZ(90deg);
  }
  30% {
    transform: translate(calc(3*71.21px), -53px) rotateZ(135deg);
  }
  40% {
    transform: translate(calc(4*71.21px), 0px) rotateZ(180deg);
  }
  50% {
    transform: translate(calc(5*71.21px), -53px) rotateZ(225deg);
  }
  60% {
    transform: translate(calc(6*71.21px), 0px) rotateZ(270deg);
  }
  70% {
    transform: translate(calc(7*71.21px), -53px) rotateZ(315deg);
  }
  80% {
    transform: translate(calc(8*71.21px), 0px) rotateZ(360deg);
  }
}
```

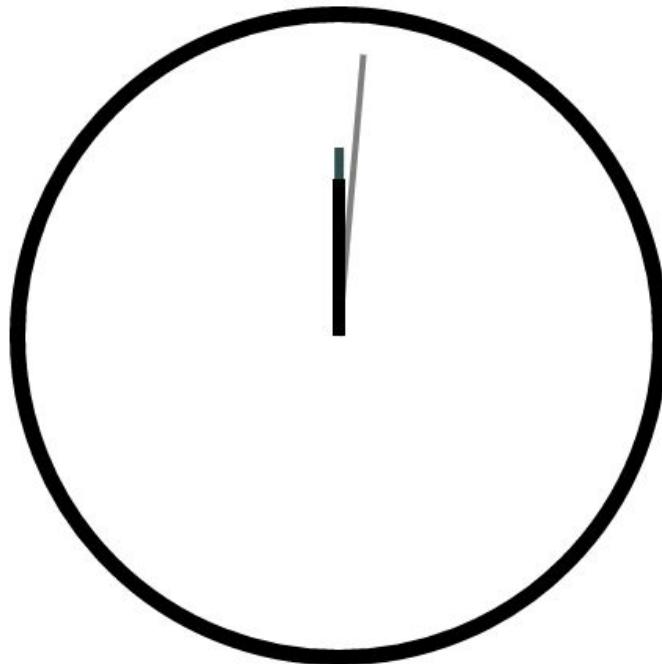
Desplazamiento de
71.21 px a la derecha
y 53 px hacia arriba

Rotación de 45
grados respecto de
su centro en sentido
horario



Animaciones

- Usando
 - Animaciones y transiciones
- Crear el siguiente reloj con animación de todas las agujas



Dimensiones CSS

Unidad	Descripción
cm	centímetros
mm	milímetros
in	pulgadas (1in = 96px = 2.54cm)
px	pixeles (1px = 1/96 de 1in)
pt	puntos (1pt = 1/72 de 1in)
pc	picas (1 pc = 12pt)

Dimensiones CSS

Unidad	Descripción
em	Relativo al tamaño de fuente del elemento
ch	Relativo al ancho del cero “0”
rem	Relativo al tamaño de fuente del elemento raíz
vw	Relativo al 1% de la anchura del viewport
vh	Relativo al 1% de la altura del viewport
vmin	Relativo al 1% de dimensión más pequeña del viewport
vmax	Relativo al 1% de dimensión más grande del viewport
%	Relativo al ancho del elemento padre

μSeminario - Píxeles CSS

- Hasta CSS2 las unidades absolutas (in, cm, mm, pt, pc) estaban ligadas a sus medidas físicas. Es decir, 1 in (pulgada) en CSS medía siempre 1 pulgada en la pantalla, si se medía con una regla. El píxel era una unidad relativa cuyo tamaño venía definido por el concepto de **píxel de referencia**.
- Desde CSS2.1 (2011), el px es una unidad absoluta con una relación fija con las demás unidades absolutas, en concreto $96\text{px} = 1 \text{ in}$.
- Hay dos formas de definir esta relación fija.
 - a. Todas las unidades absolutas se corresponden con sus tamaños físicos
 - b. El tamaño del *px* se define en base al ***px de referencia***, y a partir del px se derivan los tamaños físicos de todas las demás unidades absolutas derivadas
- La primera solo aplica a formatos impresos o dispositivos con resoluciones muy elevadas, de forma que no es de nuestro interés
- La segunda sin embargo aplica a los dispositivos que son objetivo del desarrollo web



μSeminario - Píxeles CSS

- Las unidades absolutas, como in y cm, no son necesariamente iguales a sus tamaños reales si se miden con una regla en la pantalla. Es decir, una línea que se define como de 1 pulgada de largo no tiene porqué medir 1 pulgada en la pantalla, puede ser más corta o más larga.
- El tamaño físico de estas unidades absolutas está determinado por el tamaño físico de px, y el tamaño físico de px depende del **pixel de referencia**.
- Un píxel del dispositivo físico tiene una longitud física A, y el píxel de referencia para este dispositivo específico tiene una longitud física B.
 - La longitud de un px se establece en el múltiplo de A que mejor se aproxima a B.

μSeminario - Píxeles CSS

- Un ejemplo:
 - El tamaño del píxel de un dispositivo físico cualquiera es de 0,12 mm y el píxel de referencia para este tipo de dispositivo es de 0,26 mm (p, ej.)
 - El múltiplo de 0,12mm que mejor se aproxima a 0,26 mm es $2 * 0,12 \text{ mm} = 0,24 \text{ mm}$
 - Por lo tanto, la longitud física de un px es de 0,24 mm
 - La longitud física de 1 pulgada (que equivale a 96 px) en esta pantalla será $96 * 0,24 \text{ mm} = 23,04 \text{ mm}$, es decir, un poco menos que la longitud real de una pulgada de 25,4 mm
 - El **device pixel ratio** (a veces llamado **CSS pixel ratio**) equivale a 2



μSeminario - Píxeles CSS

- Otro ejemplo:
 - El Huawei P10 tiene una pantalla de 432 PPI, es decir, un tamaño de píxel físico de 0,0588 mm.
 - Supongamos que el tamaño de píxel de referencia para este dispositivo es de 0,15 mm
 - El múltiplo de 0,0588 mm más cercano a 0,15 mm es $3 * 0,0588 \text{ mm} = 0,1764 \text{ mm}$.
 - Por lo tanto, el tamaño de px en el Huawei P10 es de 0,1764 mm.
 - Una línea de 1 pulgada en este dispositivo mide, por tanto, $96 * 0,1764 \text{ mm} = 16,93 \text{ mm}$.
 - Lo importante es saber que el **device pixel ratio** del Huawei P10 es 3.

μSeminario - Píxeles CSS

- Los pasos para calcular el tamaño del pixel físico de cualquier dispositivo son los siguientes
 - Buscar la densidad de píxeles (PPI) del dispositivo en cuestión
 - Calcular el tamaño físico del pixel del dispositivo
 - Dividir 1 / PPI -> Esto es el tamaño físico de un pixel del dispositivo en pulgadas
 - Calcula el tamaño físico de un px (Pixel CSS)
 - Multiplica el tamaño del pixel de dispositivo por el **device pixel ratio (DPR)**

μSeminario - Píxeles CSS

- Device Pixel Ratio (DPR)
 - El **DPR** como vimos arriba, es el ratio entre píxeles físicos (o de dispositivo) y píxeles lógicos (píxeles CSS)
 - Es el número de píxeles físicos que equivalen a un pixel CSS
 - Lo definen los fabricantes para cada dispositivo, aunque no está del todo claro si aplicaciones como los navegadores intervienen de alguna manera
 - El objetivo del DPR es “mentir” sobre la resolución del dispositivo: p. ej. si la resolución del Huawei P10 es de 1080x1920 px y su DPR es 3, entonces para los navegadores su resolución será en apariencia de 360 x 640 px
 - El DPR no tiene porque ser un número entero
- El propósito del **DPR** es simplemente evitar que necesitemos lupas para ver el contenido en dispositivos con densidades de píxeles (**PPI**) muy elevadas

Referencias

- [Flexbox Froggy](#) - Juego CSS para practicar con Flexbox
- [Grid Garden](#) - Juego CSS para practicar con Grid
- [Coding Fantasy](#) - Sitio Web con 3 juegos para practicar Flexbox, Grid y JS
- [CSS Grid Generator](#) - Permite generar el código de un Grid con algunos clicks
- [Ejercicios CSS Grid](#) - Muchos ejercicios de CSS Grid (con soluciones)
- [Josh Comeau](#) - Artículo completo sobre animaciones y transiciones
- [Easing Functions](#) - Compilación de easing functions avanzadas
- [Tom Roth](#) - ¿Cómo funcionan los píxeles CSS?
- [CSS Pixels](#) - Otro artículo sobre cómo funcionan los píxeles CSS

Tema 4: Bootstrap

Asignatura: Diseño de Interfaces Web



Contenidos

1. [¿Qué es Bootstrap\(5\)?](#)
2. [Breakpoints](#)
3. [Containers](#)
4. [BS Grid](#)
5. [Utilidades](#)
6. [Componentes](#)
 - 6.1. [Cards](#)
 - 6.2. [Dropdowns](#)
 - 6.3. [Navbars](#)
 - 6.4. [Carousel](#)



¿Qué es Bootstrap(5)?

- “Framework CSS que permite construir con facilidad sitios web responsivos y orientados a dispositivos móviles”
- Es una librería de hojas de estilo CSS compiladas desde código SASS. Algunas características requieren también alguna dependencia en JavaScript.
- “...responsivos y orientados a dispositivos móviles”: el framework impone un flujo de diseño que se inicia con los dispositivos móviles
- La [web oficial](#) contiene documentación detallada y con muchos ejemplos

¿Qué es Bootstrap(5)?

- Para empezar a utilizarlo tenemos las siguientes opciones:
 - CSS / JS precompilado y minimizado desde esta [URL](#)
 - Descomprimimos el ZIP y *linkamos* el fichero **bootstrap.css** en nuestro documento HTML como haríamos con cualquier hoja de estilos
 - *Source files* desde la siguiente [URL](#)
 - Se trata del código SASS de origen
 - Tendremos que compilarlo a CSS para poder utilizarlo
 - Descargar desde un CDN incluyendo en el documento HTML las líneas indicadas en esta [URL](#)

Breakpoints

- Conjunto de infijos para clases CSS que dotan de responsividad al diseño en función del ancho del dispositivo

Breakpoint	Infijo	Dimensiones
Extra small	None	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px



Breakpoints

- No existe *breakpoint* para dispositivos extra pequeños porque es el que aplica por defecto
- Están construidos a partir de *media queries* de tipo *min-width* y *max-width*

```
// Dispositivos xs (ancho inferior a 576px)
// No se define MQ al ser el comportamiento por
defecto

// Dispositivos s (a partir de 576px)
@media (min-width: 576px) { ... }

// Dispositivos m (tablets, a partir de 768px)
@media (min-width: 768px) { ... }

// Dispositivos l (monitores, a partir de
992px)
@media (min-width: 992px) { ... }

// Dispositivos xl (monitores grandes, a partir
de 1200px)
@media (min-width: 1200px) { ... }

// Dispositivos xxl (monitores aun más grandes,
a partir de 1400px)
@media (min-width: 1400px) { ... }
```

Containers

- Junto con los *breakpoints* son el bloque básico de construcción de BS
- Nos permiten agrupar, disponer y centrar el contenido del sitio web
- BS ofrece 3 tipos distintos de *containers*:
 - **.container**: el ancho del contenedor cambia con cada breakpoint
 - **.container-{breakpoint}**: el ancho del contenedor es del 100% hasta el breakpoint especificado, después cambia para cada tamaño de dispositivo
 - **.container-fluid**: el ancho del contenedor es siempre el 100% del ancho de su padre (da igual el ancho del dispositivo)

Containers

	Extra small $<576px$	Small $\geq 576px$	Medium $\geq 768px$	Large $\geq 992px$	X-Large $\geq 1200px$	XX-Large $\geq 1400px$
.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%



BS Grid

- El *Grid* de BS es el equivalente al *Grid CSS* (aunque se basa en *Flexbox*) y permite construir diseños responsivos mediante un sistema basado en *containers*, columnas, filas y *breakpoints*
 - El *container* sirve para centrar y alinear horizontalmente el contenido del *Grid*
 - Las filas actúan como *wrappers* de las columnas
 - ... que son siempre **12**
 - Las columnas están separadas por un *gap* o *gutter* en términos de BS
 - El tamaño de las columnas se controla usando los infijos de clase definidos para los 6 *breakpoints* visto previamente

BS Grid

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Prefijo de clase	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
Número de columnas				12		
Ancho del <i>Gutter</i>			1.5rem (.75rem a cada lado)			
<i>Gutters</i> personalizados				Si		
Anidado				Si		
Columnas ordenables				Si		



BS Grid

- rows
 - elemento fila, contenedor del elemento columna
 - `class="row"`
- columns
 - elemento columna, 12 por fila
 - `class="col-{breakpoint}-{1...12|auto}"`
- gutters
 - equivalente a `gap` en el Grid CSS
 - `class="g{x|y}-{breakpoint}-{0...5}"` *4 por defecto, ver siguiente diapositiva

Utilidades

- Tamaños *relativos al spacer* por defecto = **1rem** (**\$spacer** en el mapa SASS)
 - **0** - 0
 - **1** - spacer * 0.25
 - **2** - spacer * 0.5
 - **3** - spacer
 - **4** - spacer * 1.5
 - **5** - spacer * 3
 - **auto** - auto

Utilidades

- *background-color*
 - Añadir fondos de colores predeterminados
 - `class="bg-{primary|secondary|success|danger|warning|info|light|dark|white}"`
 - Añadir un gradiente
 - `class="bg-{primary|secondary|success|danger|warning|info|light|dark|white}.bg-gradient"`
 - Opacidad del borde
 - `class="bg-opacity-{10|25|50|75}"`
 - 10, 25...75 %

Utilidades

.bg-primary
.bg-secondary
.bg-success
.bg-danger
.bg-warning
.bg-info
.bg-light
.bg-dark
.bg-body
.bg-white
.bg-transparent

.bg-primary.bg-gradient
.bg-secondary.bg-gradient
.bg-success.bg-gradient
.bg-danger.bg-gradient
.bg-warning.bg-gradient
.bg-info.bg-gradient
.bg-light.bg-gradient
.bg-dark.bg-gradient

Utilidades

- *color*
 - Color del texto
 - `class="text-{primary|secondary|success|danger|warning|info|light|dark|white|body|muted|white-50|dark-50}"`
 - Opacidad
 - `class="text-opacity-{10|25|50|75}"`
- *visibilidad*
 - `class="visible|invisible"`

Utilidades

- *border*
 - Añadir / eliminar bordes
 - `class="border-{top|end|bottom|start}"`
 - Color del borde
 - `class="border-{primary|secondary|success|danger|warning|info|light|dark|white}"`
 - Opacidad del borde
 - `class="border-opacity-{10|25|50|75}"`
 - *Border radius*
 - `class="rounded-{top|end|bottom|start|circle|pill}-{0...5}"`

Utilidades

- dimensiones
 - Relativo al parente
 - `width: class="w-{10|25|50|75|auto}"`
 - `height: class="h-{10|25|50|75|auto}"`
 - `max-width: class="mw-{10|25|50|75|auto}"`
 - `max-height: class="mh-{10|25|50|75|auto}"`
 - Relativo al *viewport*
 - `class="{vw|vh}-{auto|25|50|75|auto}"`
 - `class="min-{vw|vh}-{auto|25|50|75|auto}"`

Utilidades

- *margin y padding*

- `margin: class="m{t|b|s|e|x|y}-{n}{0...5|auto}"`
- `padding: class="p{t|b|s|e|x|y}-{n}{0...5|auto}"`
 - **t** - top, arriba
 - **b** - bottom, abajo
 - **s** - start, izquierda
 - **e** - end, derecha
 - **x** - horizontales
 - **y** - verticales
 - **n** - valores negativos

Utilidades

- *justify*
 - `align: class="text-{left|center|right|justify}"`
 - `transform: class="text-{lowercase|uppercase|capitalize}"`
 - `weight: class="text-weight-{bold|normal|light}"`
 - `italic: class="text-italic"`
- *align* (sólo para inline, inline-block, inline-table y celdas de tablas)
 - `class="text-{baseline|top|middle|bottom|text-top|text-bottom}"`

Utilidades

- *display*

- `class="d-{breakpoint}-{none|inline|inline-block|block|table|table-cell|table-row|flex|inline}"`
- Útil para esconder elementos:

Tamaño de pantalla	Clase
Escondido en todas	<code>.d-none</code>
Escondido solo en <code>xs</code>	<code>.d-none .d-sm-block</code>
Escondido solo en <code>sm</code>	<code>.d-sm-none .d-md-block</code>



Utilidades

- *position*
 - `class="position-{static|relative|absolute|fixed|sticky}"`
- *float*
 - `class="float-{breakpoint}-{left|right|none}"`
 - `clearfix: class="clearfix"`

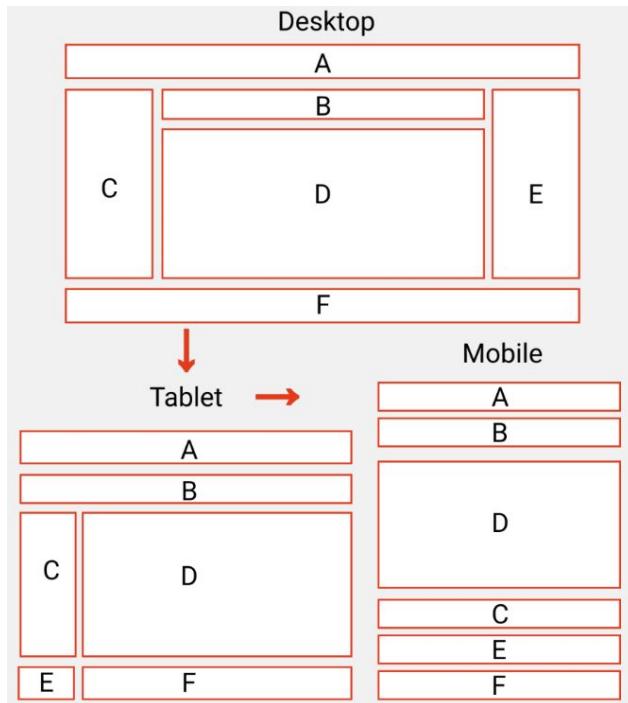
Utilidades

- **flex**

- direction: class="flex-{breakpoint}-{row|column|row-reverse|column-reverse}"
- justify-content: class="justify-content-{breakpoint}-{start|end|center|between|around}"
- align-items: class="align-items-{breakpoint}-{start|end|center|baseline|stretch}"
- align-self: class="align-self-{breakpoint}-{start|end|center|baseline|stretch}"
- wrap: class="flex-{breakpoint}-{wrap|nowrap|wrap-reverse}"
- order: class="order-{breakpoint}-{0...12}"
- align-content: class="align-content-{breakpoint}-{start|end|center|around|stretch}"

Ejemplo

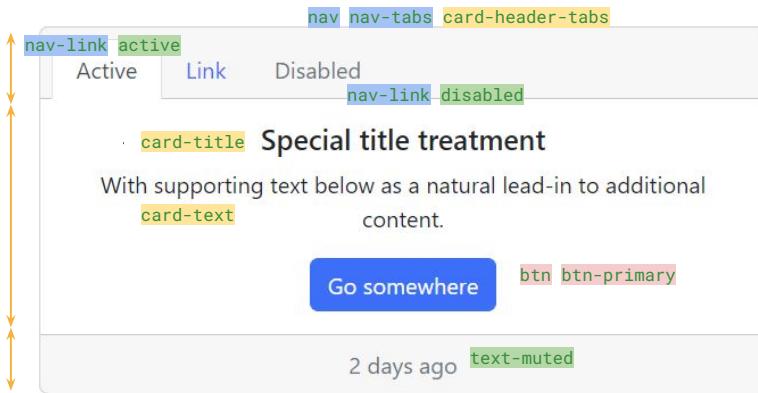
- Usando
 - BS
- Crea el siguiente diseño *responsive*
 - Móvil: **width < md**
 - Tablet: **md < width < l**
 - Escritorio: **l > 1020px**



Componentes - Card

```
<div class="card text-center">
  <div class="card-header">
    <ul class="nav nav-tabs card-header-tabs">
      <li class="nav-item">
        <a class="nav-link active" href="#">Active</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Disabled</a>
      </li>
    </ul>
  </div>
  <div class="card-body">
    <h5 class="card-title">Special title treatment</h5>
    <p class="card-text">With supporting text below as a natural lead-in to additional content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
  <div class="card-footer text-muted">
    2 days ago
  </div>
</div>
```

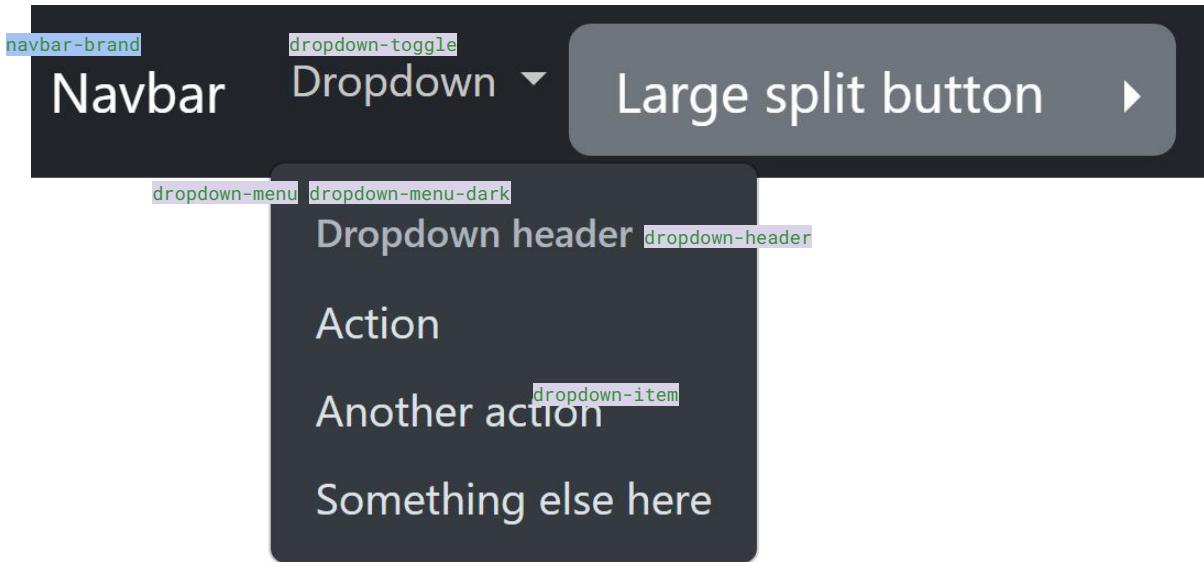
card-header
card-body
card-footer



Componentes - Dropdown

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDarkDropdown"
aria-controls="navbarNavDarkDropdown" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDarkDropdown">
      <ul class="navbar-nav">
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul class="dropdown-menu dropdown-menu-dark">
            <li><h6 class="dropdown-header">Dropdown header</h6></li>
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

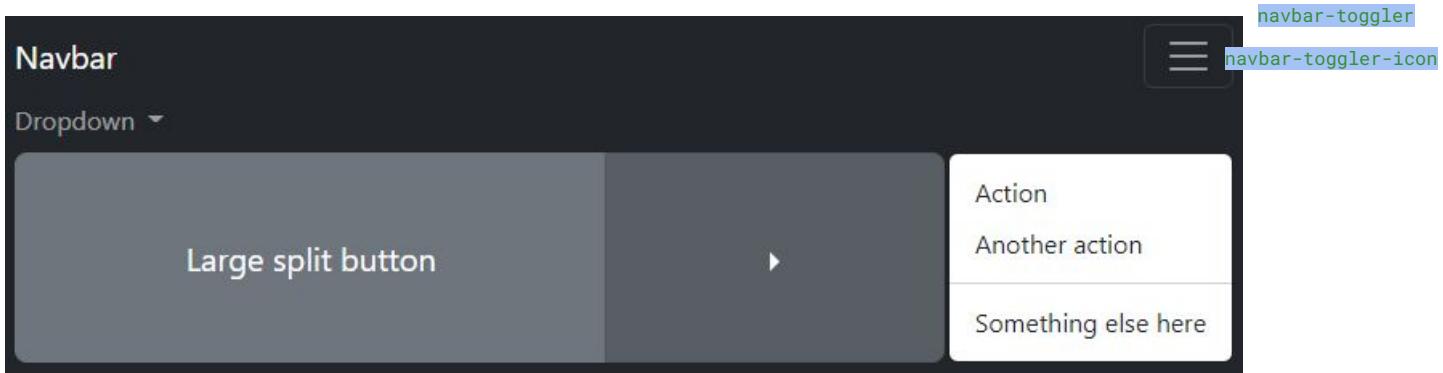
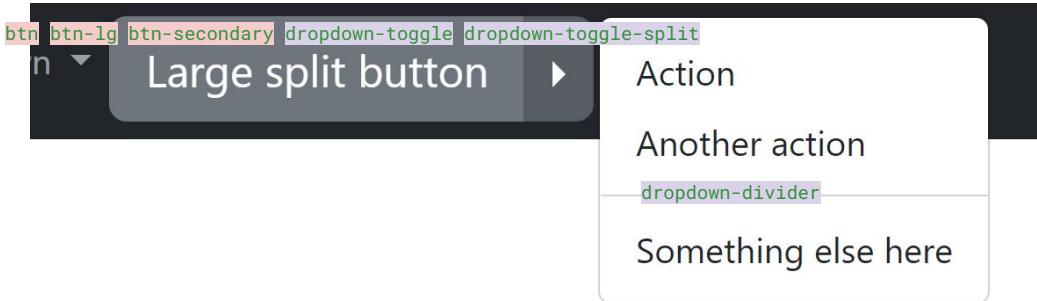
Componentes - Dropdown



Componentes - Dropdown

```
...
<li class="btn-group dropdown-center dropend">
    <button class="btn btn-secondary btn-lg" type="button">
        Small split button
    </button>
    <button type="button" class="btn btn-lg btn-secondary dropdown-toggle dropdown-toggle-split" data-bs-toggle="dropdown"
aria-expanded="false">
        <span class="visually-hidden">Toggle Dropdown</span>
    </button>
    <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="#">Action</a></li>
        <li><a class="dropdown-item" href="#">Another action</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item" href="#">Something else here</a></li>
    </ul>
</li>
</ul>
</div>
</div>
</nav>
```

Componentes - Dropdown



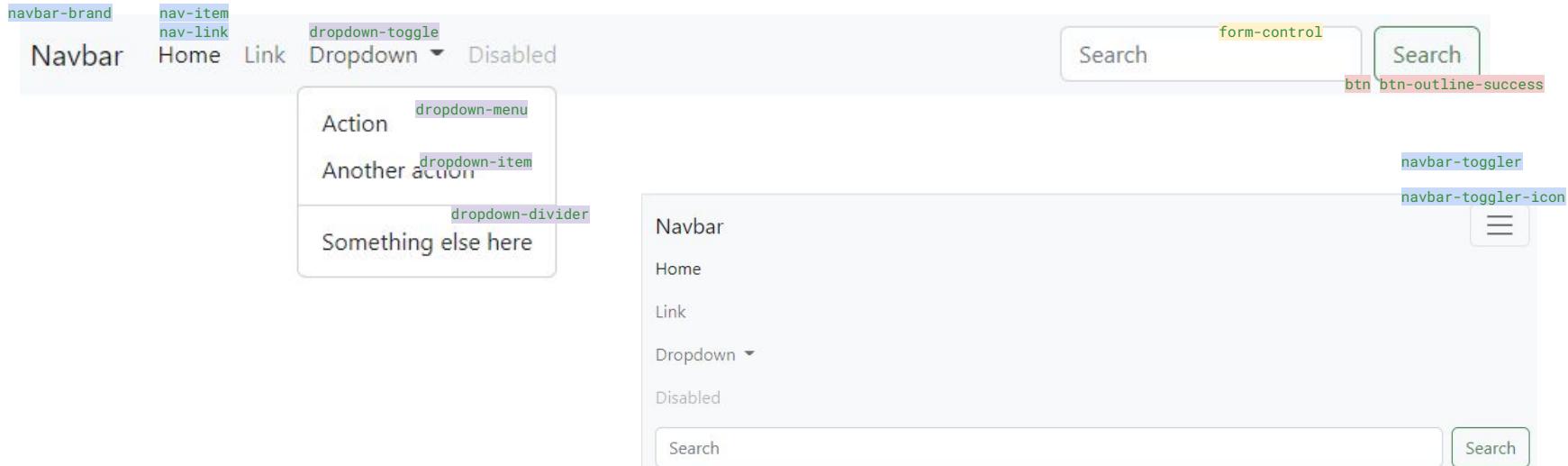
Componentes - Navbars

```

<nav class="navbar navbar-expand-lg bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" href="#">Disabled</a>
        </li>
      </ul>
      <form class="d-flex" role="search">
        <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
        <button class="btn btn-outline-success" type="submit">Search</button>
      </form>
    </div>
  </div>
</nav>

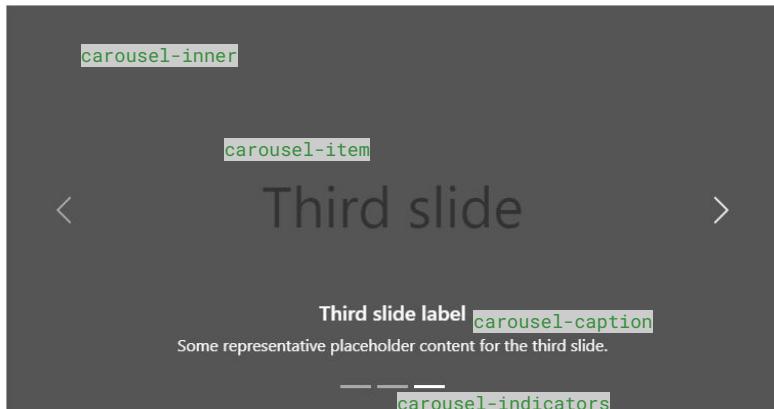
```

Componentes - Navbars



Componentes - Carousel

```
<div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="false">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-label="Slide 2"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>First slide label</h5>
        <p>Some representative placeholder content for the first slide.</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>Second slide label</h5>
        <p>Some representative placeholder content for the second slide.</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>Third slide label</h5>
        <p>Some representative placeholder content for the third slide.</p>
      </div>
    </div>
  </div>
</div>
```



Componentes - Carousel

```
<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="prev">  
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>  
  <span class="visually-hidden">Previous</span>  
</button>  
<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="next">  
  <span class="carousel-control-next-icon" aria-hidden="true"></span>  
  <span class="visually-hidden">Next</span>  
</button>  
</div>
```



Referencias

- Documentación oficial de Bootstrap
 - <https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Tema 4: Introducción a Sass

Asignatura: Diseño de Interfaces Web



Contenidos

1. ¿Qué es SASS?
2. Anidamiento
3. Variables e interpolación
4. At-Rules
5. Mixins e @include
6. Flow Control

¿Qué es Sass?

“Sass es un lenguaje de hojas de estilo que es compilado a CSS”

- No es procesable de forma directa por los navegadores y debe ser compilado a CSS con sus propias herramientas
- Extiende el lenguaje CSS con variables, sentencias de control de flujo como *if-else* y bucles *for*, funciones y más...
- Facilita la organización del código CSS en módulos y su reutilización dentro y entre proyectos
- Ofrece una API en JS y otra en Dart

¿Qué es Sass?

“Sass es un lenguaje de hojas de estilo que es compilado a CSS”

- No es procesable de forma directa por los navegadores y debe ser compilado a CSS con sus propias herramientas
- Extiende el lenguaje CSS con variables, sentencias de control de flujo como *if-else* y bucles *for*, funciones y más...
- Facilita la organización del código CSS en módulos y su reutilización dentro y entre proyectos
- Ofrece una API en JS y otra en Dart

¿Qué es Sass?

1. Instalación

a. Descargar el binario para el OS / ARCH correspondiente desde la URL

- <https://github.com/sass/dart-sass/releases/tag/1.56.1>
- Añadir el fichero binario al **PATH** del sistema

b. Usando Node.js

- `npm install -g sass`

c. Extensión para Visual Code (**RECOMENDADA**)

- <https://marketplace.visualstudio.com/items?itemName=glenn2223.live-sass>
- **OJO**, no confundir con la extensión de Ritwick Dey (con más descargas) que está discontinuada

¿Qué es Sass?

- Sass admite dos sintaxis distintas, con su propio formato
 - Formato **SCSS (style.scss)**: Extiende a CSS de modo que **todo código CSS válido es SCSS válido** (no a la inversa)

```
@mixin button-base() {  
  @include typography(button);  
  @include ripple-surface;  
  @include ripple-radius-bounded;  
  
  display: inline-flex;  
  position: relative;  
  height: $button-height;  
  border: none;  
  vertical-align: middle;  
  
  &:hover { cursor: pointer; }  
  
  &:disabled {  
    color: $mdc-button-disabled-ink-color;  
    cursor: default;  
    pointer-events: none;  
  }  
}
```

¿Qué es Sass?

- Sass admite dos sintaxis distintas, con su propio formato
 - Formato **SASS (style.sass)**: Es la sintaxis original. Sustituye las llaves de apertura y cierre {} por indentaciones

```
@mixin button-base()
  @include typography(button)
  @include ripple-surface
  @include ripple-radius-bounded

  display: inline-flex
  position: relative
  height: $button-height
  border: none
  vertical-align: middle

  &:hover
    cursor: pointer

  &:disabled
    color: $mdc-button-disabled-ink-color
    cursor: default
    pointer-events: none
```

¿Qué es Sass?

1. Un fichero SCSS se compone de

a. Sentencias

- i. Declaradas obligatoriamente al principio del documento SCSS
 1. Declaración de módulos con `@use`
 2. `@import` (como en CSS pero en proceso de ser *depreciado*)
 3. Mixins con `@mixin`, una agrupación de estilos parametrizables
 4. Funciones con `@function`
- ii. Sentencias CSS (menos dentro de bloques `@function`)
 1. Estilos, p.ej. `h1 { /* ... */ }`
 2. *At-Rules*, como `@media` o `@font-face`
 3. Mixins que utilizan `@include`

¿Qué es Sass?

- i. Sentencias universales
 - 1. Variables como, `$var: value`
 - 2. *Flow control*, como `@if` o `@each`
 - 3. `@error`, `@warn` o `@debug`
- b. Expresiones, que es cualquier cosa que produce un valor
 - i. Literales, que representan valores estáticos: Números, *strings*, colores, booleanos, `null`, listas y *maps*
 - ii. Operaciones entre literales: `==`, `!=`, `+`, `-`, `*`, `/`, `<`, `>`, `<=`, `>=`
 - iii. Variables como `$var`, llamadas a funciones como `nth($list, 1)`, funciones especiales como `calc(1px + 100%)`, el selector padre `&` y el valor `!important`



¿Qué es Sass?

- Los comentarios en SASS se escriben de la siguiente manera

```
// Este comentario no se incluirá en el CSS

/* Este sí se incluirá, excepto en modo comprimido */

/* Pueden contener interpolaciones:
 * 1 + 1 = #{1 + 1} */

/*! Este comentario se incluirá en el CSS incluso en modo comprimido */

p /* Se pueden escribir comentarios multilínea en cualquier sitio
   * Los espacios en blanco se permite. */ .sans {
  font: Helvetica, // Igual con los comentarios de una sola línea
  sans-serif;
}
```

Anidamiento

- Las jerarquías de selectores pueden anidarse en bloques, evitando repeticiones

// SASS

```
.foo {  
  background: orange;  
  
  p {  
    color: blue;  
    font-size: 14px;  
    font-weight: bold;  
  }  
}
```



/* CSS */

```
.foo {  
  background: orange;  
}  
  
.foo p {  
  color: blue;  
  font-size: 14px;  
  font-weight: bold;  
}
```

Anidamiento

- Las propiedades CSS también pueden anidarse *Corregir

// SASS

```
.foo {  
    background: orange;  
  
    font: {  
        family: arial;  
        weight: bold;  
        size: 2em;  
    }  
}
```



/* CSS */

```
.foo {  
    font-family: arial;  
    font-weight: bold;  
    font-size: 2em;  
}
```

Anidamiento

- El literal & permite seleccionar el elemento padre

// SASS

```
a {  
    color: blue;  
    text-decoration: none;  
    &:hover {  
        text-decoration: underline;  
    }  
}
```



```
/* CSS */  
  
a {  
    color: blue;  
    text-decoration: none;  
}  
  
a:hover {  
    text-decoration: underline;  
}
```

Anidamiento

- El literal & permite seleccionar el elemento padre

```
// SASS
```

```
.foo {  
    background: orange;  
    &.block {  
        border: 1px solid;  
    }  
}
```

```
/* CSS */  
  
.foo {  
    background: orange;  
}  
  
.foo.block {  
    border: 1px solid;  
}
```

Variables e interpolación

- Las variables Sass permiten una vez declaradas reutilizar valores de propiedades como colores (RGBa, HSL...), tamaños, etc...
- Su sintaxis es la siguiente:

`$variable-name: value`

// SASS

```
$border-color: #ccc;  
  
.box {  
    border: 1px solid $border-color;  
}
```



/* CSS */

```
.box {  
    border: 1px solid #ccc;  
}
```

Variables e interpolación

- Se pueden utilizar expresiones matemáticas de forma directa

```
// SASS                                /* CSS */  
  
$percentage: 100%;  
  
html {  
    font-size: (12 / 16) * $percentage;  
}  
  
—————>  
  
html {  
    font-size: 75%;  
}
```

VARIABLES E INTERPOLACIÓN

- La interpolación es una característica que permite utilizar expresiones en casi cualquier lugar de una hoja de estilos Sass
 - Selectores
 - Nombres de propiedades
 - At-Rules CSS
 - @extends
 - @import CSS
 - Strings con comillas o sin ellas
 - Nombres de funciones CSS
 - Comentarios



Variables e interpolación

- La interpolación es una característica que permite utilizar expresiones en casi cualquier lugar de una hoja de estilos Sass. Su sintaxis es

`#{ expresión }`

// SASS

```
@mixin corner-icon($name, $top-or-bottom, $left-or-right) {  
  .icon-#{$name} {  
    background-image: url("/icons/#{$name}.svg");  
    position: absolute;  
    #{$top-or-bottom}: 0;  
    #{$left-or-right}: 0;  
  }  
}  
  
@include corner-icon("mail", top, left);
```

/* CSS */

```
.icon-mail {  
  background-image: url("/icons/mail.svg");  
  position: absolute;  
  top: 0;  
  left: 0;  
}
```



At-Rules

- Constituyen el grueso de las funcionalidades extra que ofrece Sass. Son las siguientes
 - `@use`
 - `@forward`
 - `@import`
 - `@mixin` e `@include`
 - `@function`
 - `@extend`
 - `@at-root`
 - `@error, @warn` y `@debug`
 - `@if, @each, @for` y `@while`

At-Rules

- **@use** que permite cargar o importar mixins, funciones y variables desde otras hojas de estilo Sass y también combinar CSS con distinto origen

```
// style.scss
@use 'foundation/code';
@use 'foundation/lists';
```

módulos

=

```
// fichero foundation/_code.scss
code {
  padding: .25em;
  line-height: 0;
}
```

+

```
// fichero foundation/_lists.scss
ul, ol {
  text-align: left;
}
& & {
  padding: {
    bottom: 0;
    left: 0;
  }
}
```



At-Rules *CORREGIR

- Cada módulo define un **namespace** con todas sus *variables*, *mixins*, *funciones*
- Los miembros de un módulo se acceden con la sintaxis *namespace.member*
- Los miembros de un módulo sólo son accesibles en la hoja de estilos que lo carga con **@use**.
- Lo anterior se resuelve utilizando **@forward**, que carga los miembros del módulo como si fueran propios

```
// style.scss
@use "src/corners";  
  
.button {
  @include corners.rounded;
  padding: 5px + corners.$radius;
}
```

carga el módulo

```
// src/_corners.scss
$radius: 3px;  
  
@mixin rounded {
  border-radius: $radius;
}
```

At-Rules *CORREGIR

- Se puede cargar el módulo con un namespace personalizado
- ... o cargar sus miembros sin namespace
- Se pueden declarar miembros privados añadiendo el prefijo - (signo menos) delante del nombre

```
// style.scss
@use "src/corners" as *;

.button {
  @include rounded;
  padding: 5px + $radius;
}
```

```
// style.scss
@use "src/corners" as *;

.button {
  @include rounded;
  padding: 5px + $radius;
}
```

At-Rules

- Sass permite definir funciones como las que ofrecen otros lenguajes de programación usando la at-rule **@function**
- El valor de retorno de la función se declara usando **@return expresión**

```
@function pow($base, $exponent) {  
    $result: 1;  
  
    @for $_ from 1 through $exponent {  
        $result: $result * $base;  
    }  
  
    @return $result;  
}  
  
.sidebar {  
    float: left;  
    margin-left: pow(4, 3) * 1px;  
}
```



At-Rules

- Admite argumentos **opcionales** y **arbitrarios**
- Cuando se llama a una función, sus argumentos pueden pasarse por orden de definición o en cualquier orden pero especificando su nombre
- Sass proporciona varias funciones *built-in* de utilidad

```
@function invert($color, $amount: 100%) {
  $inverse: change-color($color, $hue: hue($color) + 180);

  @return mix($inverse, $color, $amount);
}

$primary-color: #036;

.header {
  background-color: invert($primary-color, 80%);
}
```

```
@function sum($numbers...) {
  $sum: 0;

  @each $number in $numbers {
    $sum: $sum + $number;
  }

  @return $sum;
}

.micro {
  width: sum(50px, 30px, 100px);
}
```

At-Rules

- **@error, @warn y @debug** seguidas de una expresión permiten emitir en tiempo de compilación mensajes de aviso o simplemente informativos
- Es importante destacar lo de “*en tiempo de compilación*”, es decir, los mensajes se generarán y serán visibles para el usuario durante el proceso de compilación a CSS

```
@mixin inset-divider-offset($offset, $padding) {  
  $divider-offset: (2 * $padding) + $offset;  
  @debug "divider offset: #{\$divider-offset}";  
  
  margin-left: $divider-offset;  
  width: calc(100% - #{$divider-offset});  
}
```



```
test.scss:3 Debug: divider offset:  
132px
```

At-Rules

A partir del material disponible en el repositorio GH de la asignatura

- <https://github.com/tcamba-ciclos-montecastelo/diw-22/tree/master/t5-sass/ej-1>

realiza la web que se observa en el siguiente video

- https://drive.google.com/file/d/19-u9AUuo6S6pqAGk-d3XX_aUi8dGqS/view?usp=share_link

utilizando

- **Sass**

At-Rules

- Instrucciones

- Declarar (y utilizar) una variable **\$fontstack** y almacenar la fuente importada de Google 'Gotu' en esta variable
- Declarar (y utilizar) una variable **\$fontsize** con valor 16px
- Declarar una variable **\$primary** con valor *lightskyblue*. Utiliza una función Sass para oscurecer este color y guarda el resultado en la variable **\$primary_dark**
 - Más tarde, puedes cambiar este color por otro color "claro", por ejemplo, rosa claro (0:27 en el vídeo de la solución) o aguamarina (0:37 en el vídeo de la solución) y toda tu página (incluida la barra de navegación) se "recoloreará"
- Añade un reset universal, aplica una altura de línea estándar y añade algunos paddings donde sea necesario
- El **#contenedor** debe estar centrado horizontalmente y tiene un ancho máximo de 800px
- El color del texto del encabezado principal **h1** es **\$primary-dark**



At-Rules

- Instrucciones
 - Utiliza **flexbox** para transformar la lista desordenada del elemento **nav** en una barra de navegación completa
 - Utilizar la anidación (para que la otra lista del documento no se vea afectada)
 - El color de fondo de los elementos es igual a **\$primary**, mientras que el contorno (da mejores resultados que el borde) y los textos de los enlaces se muestran en **\$primary-dark**
 - Al pasar por encima de un elemento del menú, el enlace se pone en negrita y los colores (color y color de fondo) cambian en una transición de 1s
 - Para pantallas pequeñas, los elementos del menú se colocan uno debajo del otro. A partir de un ancho de pantalla de 650px, los elementos del menú se colocan uno al lado del otro
 - Ajustar el margen izquierdo de la lista desordenada en el elemento principal (para que las viñetas estén alineadas horizontalmente con el resto del documento)

Mixins e @include

- Permiten definir agrupaciones o “paquetes” de estilos parametrizables fáciles de reutilizar
- Se declaran utilizando la **at-rule @mixin**
- Se “importan” con la at-rule **@include**

```
@mixin reset-list {
  margin: 0;
  padding: 0;
  list-style: none;
}

@mixin horizontal-list {
  @include reset-list;

  li {
    display: inline-block;
    margin: {
      left: -2px;
      right: 2em;
    }
  }
}

nav ul {
  @include horizontal-list;
}
```



```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}

nav ul li {
  display: inline-block;
  margin-left: -2px;
  margin-right: 2em;
}
```

Mixins e @include

- Del mismo modo que en @function, @mixin admite argumentos opcionales

```
@mixin replace-text($image, $x: 50%, $y: 50%) {  
  text-indent: -99999em;  
  overflow: hidden;  
  text-align: left;  
  
  background: {  
    image: $image;  
    repeat: no-repeat;  
    position: $x $y;  
  }  
}  
  
.mail-icon {  
  @include replace-text(url("/images/mail.svg"), 0);  
}
```



```
.mail-icon {  
  text-indent: -99999em;  
  overflow: hidden;  
  text-align: left;  
  background-image: url("/images/mail.svg");  
  background-repeat: no-repeat;  
  background-position: 0 50%;  
}
```

Mixins e @include *CORREGIR

- ... y también arbitrarios

```
@mixin square($size, $radius: 0) {  
    width: $size;  
    height: $size;  
  
    @if $radius != 0 {  
        border-radius: $radius;  
    }  
}  
  
.avatar {  
    @include square(100px, $radius: 4px);  
}
```



```
.avatar {  
    width: 100px;  
    height: 100px;  
    border-radius: 4px;  
}
```

At-Rules

A partir del material disponible en

- <https://github.com/tcambacicos-montecastelo/diw-22/tree/master/t5-sass/ej-2>

Escribe el fichero **style.scss** que produce el siguiente resultado

Dolor sit amet

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

A culpa eaque eligendi est harum hic maxime nulla pariatur quam quia, quibusdam quo, rerum sint temporibus, vel veniam voluptas? Adipisci amet atque, dicta eius eum exercitationem, fugit minus necessitatibus nisi nulla totam vero, voluptas volutate. Aperiam assumenda beatae blanditiis commodi cum debitis deleniti eaque eius excepturi explicabo.

Ad aliquam amet animi, aperiam, architecto aspernatur atque corporis deserunt doloremque eaque eos est excepturi exercitationem fugiat harum hic ipsum itaque libero nam nemo numquam omnis perspicatis quaerat qui quibusdam quidem reiciendis rem similiqute soluta sunt tempora totam ullam unde! Ab deleniti ea nobis quas. Assumenda cum cumque ducimus eveniet fuga, id, illum, in minus.

Dolor sit amet

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

Dolor sit amet

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

Small (≤ 768px) screen:

Medium (> 768px) screen or larger:

Large (≥ 1200px) screen:

At-Rules

- Instrucciones
 - Declarar 2 variables **\$small** (con valor 576px) y **\$medium** (con valor 768px)
 - Escribe un mixin con 2 argumentos, **\$color** y **\$size** (con valor por defecto 14px), que establezcan
 - el color (del texto) a **\$color**
 - el color de fondo al color invertido de **\$color**, pero con un ajuste de **\$lightness** del 20%
 - el tamaño de la fuente, el margen y el relleno deben ser iguales a **\$size**
 - la altura de línea igual a 1,5 (o **\$size***1,5)
 - un borde sólido de 5px y color **\$color**
 - Añadir un reset universal y algunos paddings donde sea necesario
 - Usar Verdana (con un tipo de letra de respaldo adecuado)
 - Utilizar el mixin para dar a **#container** un diseño basado en *darkred* (y el tamaño por defecto **\$size** de 14px)
 - En pantallas pequeñas, el diseño debe estar basado en *darkblue* y 16px
 - En pantallas medianas (o más grandes), el diseño debe estar basado en *darkgreen* y 20px



Flow Control

- @if y @else

```

@use "sass:math";

@mixin triangle($size, $color, $direction) {
  height: 0;
  width: 0;

  border-color: transparent;
  border-style: solid;
  border-width: math.div($size, 2);

  @if $direction == up {
    border-bottom-color: $color;
  } @else if $direction == right {
    border-left-color: $color;
  } @else if $direction == down {
    border-top-color: $color;
  } @else if $direction == left {
    border-right-color: $color;
  } @else {
    @error "Unknown direction #{$direction}.";
  }
}

.next {
  @include triangle(5px, black, right);
}

```



Flow Control

- @each

```
$sizes: 40px, 50px, 80px;

@each $size in $sizes {
  .icon-#${$size} {
    font-size: $size;
    height: $size;
    width: $size;
  }
}
```



```
.icon-40px {
  font-size: 40px;
  height: 40px;
  width: 40px;
}

.icon-50px {
  font-size: 50px;
  height: 50px;
  width: 50px;
}

.icon-80px {
  font-size: 80px;
  height: 80px;
  width: 80px;
}
```

Flow Control

- @each

```
$icons: ("eye": "\f112", "start": "\f12e", "stop": "\f12f");

@each $name, $glyph in $icons {
  .icon-#{$name}:before {
    display: inline-block;
    font-family: "Icon Font";
    content: $glyph;
  }
}
```



CSS OUTPUT

```
@charset "UTF-8";
.icon-eye:before {
  display: inline-block;
  font-family: "Icon Font";
  content: ""; CORREGIR
}

.icon-start:before {
  display: inline-block;
  font-family: "Icon Font";
  content: "";
}

.icon-stop:before {
  display: inline-block;
  font-family: "Icon Font";
  content: "";
}
```

Flow Control

- @for y @while

```
$base-color: #036;

@for $i from 1 through 3 {
  ul:nth-child(3n + ${i}) {
    background-color: lighten($base-color, $i * 5%);
  }
}
```



```
ul:nth-child(3n + 1) {
  background-color: #004080;
}

ul:nth-child(3n + 2) {
  background-color: #004d99;
}

ul:nth-child(3n + 3) {
  background-color: #0059b3;
}
```

```
@use "sass:math";

@function scale-below($value, $base, $ratio: 1.618) {
  @while $value > $base {
    $value: math.div($value, $ratio);
  }
  @return $value;
}

$normal-font-size: 16px;
sup {
  font-size: scale-below(20px, 16px);
}
```



```
sup {
  font-size: 12.36094px;
}
```

At-Rules

A partir del material disponible en el repositorio GH de la asignatura

- <https://github.com/tcamba-ciclos-montecastelo/diw-22/tree/master/t5-sass/ej-3>

realiza la web que se observa en el siguiente video

- [https://drive.google.com/file/d/1DDGqoTI_U1OJqkCUgm5W1HtmvmxAbu3o/vie
w?usp=share_link](https://drive.google.com/file/d/1DDGqoTI_U1OJqkCUgm5W1HtmvmxAbu3o/view?usp=share_link)

utilizando

- **Sass**

At-Rules

- Instrucciones
 - Escribir 3 nuevos **partials**: `_variables.scss`, `_mixins.scss` y `_reboot.scss`
 - En tu archivo principal `style.scss`, importa la fuente de Google 'Anton' y los 3 **partials** en el orden correcto
 - `_variables.scss`
 - Declara 3 variables `$fontstack-titles` ('Anton') , `$fontstack-text` (Verdana) y `$small` (600px)
 - `_mixins.scss`
 - Copiar el diseño del mixin (del ejercicio 2) en este archivo
 - `_reboot.scss`
 - Añade un reinicio universal
 - body:
 - `font: $fontstack-text`
 - diseño basado en `white` y `1rem` para pantallas extra pequeñas (`< 600px`)
 - diseño basado en `black` y `1.25rem` para pantallas pequeñas o grandes (`= 600px`)
 - encabezado principal:
 - fuente: `$fontstack-titles` con tamaño `1.75rem`
 - algo de `padding` en la parte inferior
 - párrafos (en el `article`)
 - algo de `padding` en la parte inferior
 - enlaces en todos los estados:
 - elementos de bloque con un ancho de `130px`
 - fuente: `$fontstack-titles`
 - diseño basado en negro y `1rem` (¡eliminar el margen a la izquierda!)
 - enlaces al hacer `hover`:
 - diseño basado en `dimgray` y `1rem`

At-Rules

A partir del material disponible en el repositorio GH de la asignatura

- <https://github.com/tcamba-ciclos-montecastelo/diw-22/tree/master/t5-sass/ej-4>

realiza la web que se observa en el siguiente video

- <https://drive.google.com/file/d/17U7ZaBiLkFU5w14DhLBrGJEdgT9o-Oaq/view?usp=sharing>

utilizando

- **Sass**

At-Rules

- Instrucciones
 - Hacer 2 nuevos parciales: `_mixins.scss` y `_reboot.scss`
 - En el archivo principal `style.scss`, importa estos 2 parciales (en el orden correcto)
 - `_mixins.scss`
 - Escribe un mixin con 6 parámetros para los marcos de las fotos
 - `$thickness`: grosor del borde
 - `$style`: estilo del borde (default = `solid`)
 - `$color-border`: color del borde (default = `#000`)
 - `$color-bg`: color de fondo (default = `#eee`)
 - `$padding`: espacio en el interior del elemento (default = `10px`)
 - `$radio`: redondeo aplicado al elemento (default = `15px`)
 - y, al hacer `hover`: un `box-shadow` en negro opaco de 0.75, 5px a la derecha y al fondo, con un desenfoque de 10px



At-Rules

- Instrucciones
 - **_reboot.scss**
 - Añade un reinicio universal y algo de relleno debajo del encabezado principal
 - **cuerpo:**
 - fuente *Geneva* con una fuente de reserva adecuada
 - texto negro sobre fondo gris (#ccc)
 - **artículo:**
 - ancho máximo de 800px, centrado
 - fondo blanco (#fff)
 - contenido centrado
 - **sección**
 - Utiliza flexbox para mostrar las imágenes una al lado de la otra (con el mismo espacio entre ellas), y en varias líneas cuando sea necesario
 - **img** (en la sección)
 - anchura y altura iguales a 225px
 - algún margen en la parte inferior
 - Añadir un poco de diseño utilizando el marco mixto:
 - **1^a imagen:** grosor del borde 3px, sin especificar otros parámetros
 - **2^a imagen:** grosor del borde 3px, estilo de borde punteado, sin otros parámetros especificados
 - **3^a imagen:** grosor del borde 3px, estilo de borde sólido, color del borde verde medio, sin otros parámetros especificados
 - **4^a imagen:** grosor del borde 6px, estilo de borde sólido, color del borde #b8181b, fondo #e7d5c2, espacio de 20px en el interior y redondeo de 40px
 - **5^a imagen:** grosor del borde 6px, estilo de borde discontinuo, color del borde rgb(109,85,106), fondo **peru**, espacio de 20px en el interior y redondeo del 50%.
 - **6^a imagen:** grosor del borde 3px, fondo **#aac5d9**, espacio de 20px en el interior y redondeo del 50%, sin otros parámetros especificados

At-Rules

A partir del material disponible en el repositorio GH de la asignatura

- <https://github.com/tcamba-ciclos-montecastelo/diw-22/tree/master/t5-sass/ej-5>

realiza el mini-framework CSS como se indica en las instrucciones utilizando

- **Sass**

At-Rules

- Instrucciones
 - **.box-primary, .box-danger, ...**
 - Añadir un nuevo parcial **helpers/_box.scss**
 - Utiliza una regla **@each** para definir las clases **.box-primary, .box-danger, ...** que convierten un elemento en una "caja" con colores basados en los colores definidos en **\$color-map**
 - El color del texto es igual al color especificado
 - El color de fondo es igual al color especificado, pero con un valor alfa de 0,15
 - Añadir un borde sólido de 1px en el color especificado, pero con una luminosidad ajustada de -10%.
 - Añadir un **padding** y **margin-bottom** de 1rem
 - Añade un **border-radius** de 0.25rem
 - Después, intenta mover el "código de la caja" a un mixin color-box en **mixins/_colorbox.scss**
 - Importe este nuevo mixin parcial **mixins/_colorbox.scss** en el lugar correcto en **style.scss**
 - El mixin **color-box** debe tener (al menos) un parámetro para el **\$color** (pero también puedes añadir otros parámetros)
 - **.bg-primary-50, .bg-danger-90, ...**
 - Añadir un nuevo parcial **helpers/_background.scss**
 - Usa una iteración anidada (**@for** dentro de **@each** o viceversa) para generar las clases **.bg-primary-X, .bg-danger-X, ...** con X igual a 0, 10, 20, ..., 100
 - El color de fondo debe establecerse en el color especificado, con un valor alfa de X/100
 - Clases para el relleno (**.p-3, .pt-5, ...**) y el margen (**.mx-2, ml-0, ...**)
 - Añadir un nuevo parcial **helpers/_spacing.scss**
 - Utiliza una regla **@for** para definir las clases de relleno **.p-X, .px-X, .py-X, .pt-X, .pr-X, .pb-X y .pl-X** para añadir relleno (global, horizontal, vertical, superior, derecho, inferior e izquierdo)
 - El valor de X oscila entre 0 y 5 (¡incluido!) dando como resultado un valor de relleno de X/3rem
 - Del mismo modo, genera también las correspondientes clases de margen (global, horizontal, vertical, superior, derecho, inferior e izquierdo) **.m-X, .mx-X, .my-X, .mt-X, .mr-X, .mb-X y .ml-X**

Índice de grabaciones

- Sesión 1 - Viernes 22 de Septiembre de 2023 (17:00 a 20:00)
 - [Inicio Sesión 1. Criterios de evaluación y planificación](#) - 0h0m0s
 - [Tema 1. Introducción](#) - 0h48m0s
 - [Descanso](#) - 1h31m0s
 - [Configuración del IDE](#) - 1h52m0s
 - [Repaso HTML](#) - 2h14m0s
 - [Introducción a CSS](#) - 2h30m0s
- Sesión 2 - Viernes 29 de Septiembre de 2023 (17:00 a 20:00)
 - [Inicio Sesión 2. Repaso sesión anterior](#) - 0h0m0s
- Sesión 3 - Jueves 05 de Octubre de 2023 (17:00 a 20:00)
 - [Inicio Sesión 3. Repaso sesión anterior](#) - 0h0m0s

Índice de grabaciones 22-23

- **Sesión 1** - Jueves 15 de Septiembre de 2022 (17:00 a 20:00)
 - [Inicio Sesión 1. Criterios de evaluación y planificación](#) - 0h0m0s
 - [Temario Asignatura](#) - 0h35m0s
 - [Desarrollo Web a alto nivel y repaso de HTML](#) - 1h10m0s
 - [Ejemplo HTML](#) - 1h47m0s
 - [CSS Básico](#) - 2h31m0s
 - [Fin Sesión 1](#) - 2h25m0s
- **Sesión 2** - Viernes 23 de Septiembre de 2022 (17:00 a 20:00)
 - [Inicio Sesión 2. Continuación CSS Básico](#) - 0h0m0s
 - [Colores](#) - 0h21m10s
 - [Ejemplo de estilos básicos](#) - 0h35m0s
 - [DevTools de Chrome](#) - 1h5m0s
 - [Inicio Tema 2. Selectores](#) - 1h39m0s
 - [Herencia](#) - 2h49m0s
 - [Modelo de caja](#) - 2h52m

Índice de grabaciones 22-23

- **Sesión 3** - Viernes 07 de Octubre de 2022 (17:00 a 20:00)
 - [Inicio Sesión 3. Continuación CSS Básico](#) - 0m0s
 - [Modelo de caja](#) - 10m0s
 - [Tipos de caja](#) - 38m56s
 - [Posicionamiento absoluto](#) - 1h04m17s
 - [Descanso](#) - 1h09m00s
 - [Continuación posicionamiento](#) - 1h27m00s
 - [Pseudo-elementos](#) - 1h42m00s
 - [Layouts](#) - 2h00m00s
 - [Float](#) - 2h28m00s
 - [Juego Selectores](#) - 2h42m00s
 - [Fin](#) - 2h46m00s
- **Sesión 4** - Viernes 14 de Noviembre de 2022 (17:00 a 20:00)
 - [Inicio Sesión 4. Continuación CSS Avanzado](#) - 0m0s
 - [Ejercicio p.absoluto y pseudoelementos](#) - 18m0s
 - [Float. Ejemplo Float](#) - 40m0s
 - [Descanso](#) - 1h19m
 - [Flexbox](#) - 1h37m
 - [Ejemplo Propiedades Flex](#) - 1h54m
 - [Fin](#) - 2h44m

Índice de grabaciones 22-23

- Sesión 5 - Viernes 27 de Octubre de 2022 (17:00 a 20:00)
 - [Inicio Sesión 5. Continuación CSS Avanzado](#) - 0m0s
 - [Flexbox. Ejemplo diap. 99](#) - 20m32s
 - [CSS Grid. Teoría](#) - 1h6m0s
 - [Descanso](#) - 1h26m20s
 - [CSS Grid. Ejemplo diap. 111](#) - 1h42m0s
 - [CSS Grid. Grid Areas](#) - 2h5m0s
 - [Media Queries. Teoría](#) - 2h16m0s
 - [Fin](#) - 2h47m0s
- Sesión 6 - Viernes 4 de Noviembre de 2022 (17:00 a 20:00)
 - [Repaso sesión 5](#) - 0h0m0s
 - [Ejemplo Grid MQ](#) - 0h27m0s
 - [Transiciones](#) - 1h05m0s
 - [Descanso](#) - 1h17m0s
 - [Cont. transiciones](#) - 1h31m49s
 - [Animaciones](#) - 1h42m00s
 - [Ejercicio propuesto Reloj](#) - 1h57m30s
 - [Unidades CSS](#) - 2h28m0s
 - [Pixel CSS](#) - 2h35m0s

Índice de grabaciones 22-23

- **Sesión 7** - Viernes 11 de Noviembre de 2022 (17:00 a 20:00)
 - [Repaso sesión 6](#)- 0h0m0s
 - [Ejemplo Reloj](#) - 0h17m0s
 - [Bootstrap](#) - 0h40m
 - [Descanso](#) - 1h17m40s
 - [Cont. BS](#)- 1h34m0s
 - [Ejemplo BS GRID](#) - 1h45m0s
 - [Utilidades BS](#) - 1h45m0s
 - [Fin](#) - 2h45m0s
- **Sesión 8** - Viernes 19 de Noviembre de 2022 (17:00 a 20:00)
 - [Repaso sesión 7](#) - 0h0m0s
 - [Actividad Propuesta](#) - 0h22m30s
 - [Solución Actividad Propuesta](#) - 0h58m0s
 - [Descanso](#) - 1h19m0s
 - [BS - Cards](#) - 1h36m0s
 - [BS - Dropdowns / Navs-Tabs / Navbars](#) - 2h03m0s
 - [BS - Carousel](#) - 2h35m0s
 - [Fin](#) - 2h48m0s



Índice de grabaciones 22-23

- **Sesión 9** - Viernes 25 de Noviembre de 2022 (17:00 a 20:00)
 - [Repaso sesión 8](#)- 0h0m0s
 - [Introducción Sass](#)- 0h10m0s
 - [Anidamiento](#)- 0h31m0s
 - [Variables e Interpolación](#)- 0h44m0s
 - [At Rules](#)- 0h52m0s
 - [Descanso](#)- 1h03m0s
 - [Continuación At Rules](#)- 1h18m0s
 - [Ejemplo 1](#)- 1h30m0s
 - [Mixins](#)- 2h15m0s
 - [Fin](#)- 2h15m0s
- **Sesión 10** - Jueves 2 de Noviembre de 2022 (17:00 a 20:00)
 - [Repaso sesión 9](#)- 0h0m0s
 - [Ejemplo 2](#)- 0h29m0s
 - [Flow control](#)- 0h54m0s
 - [Descanso](#)- 1h11m0s
 - [Ejercicio 3](#)- 1h30m0s
 - [Actividad/Ejercicio 4](#)- 2h13m44s
 - [Fin](#)- 3h0m0s