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 19:30 a 20:30**
    - Dudas, sugerencias y quejas

- 12 sesiones (10 ordinarias + 1 repaso + 1 examen)
  - Viernes 16:30 a 19:30 - Desde el 16/09/2022 al 12/09/2023
  - Sesión de repaso - Viernes 22/12/2022
  - Examen 1ra convocatoria - Jueves 12/01/2023
  - Examen 2da convocatoria - Miércoles 14/06/2023

Evaluación

- 3 tareas obligatorias (1ra conv. - 40% / 2da conv. - 25%)
  - 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]-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
  - Calificación de **4.5 o superior** para hacer media
    - Las tareas suspendidas deberán recuperarse en segunda convocatoria
    - Tarea no recuperada en 2da convocatoria - Suspenso de la asignatura
  - Publicación de calificaciones y soluciones en *Classroom*
    - Fecha de entrega oficial + 10 días

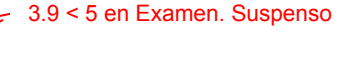
FORMACIÓN PROFESIONAL MONTECASTELO

Evaluación

- Examen (1ra conv. - 60% / 2da conv. - 75%)
  - Prueba teórico-práctica sobre los contenidos de la asignatura al completo
  - Calificación de **4.5 o superior** para hacer media
  - 1ra convocatoria - Jueves 12/01/2023
  - 2da convocatoria - Miercoles 14/06/2023
- 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.
- La calificación media de las dos partes (tareas / examen) debe ser superior a 5 para aprobar la asignatura
- 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

Evaluación

- Ejemplos de evaluación
  - Roberto
    - Tareas
      - **T1** superada con un **5**
      - **T2** superada con un **8**
      - **T3** suspendida con un 4.49
        - Recupera con un 8
    - **Examen 1ra** convocatoria → **7**
    - **Nota final** = 0.4 * (5 + 8 + 8) / 3 + 0.6 * 7 = **7**
  - Alicia
    - Tareas
      - **T1** aprueba con **8**
      - **T2** aprueba con **8**
      - **T3** superada con **6.8**
    - **Examen 1ra** convocatoria suspendido con un 3
      - **Examen 2da** convocatoria → **7**
    - **Nota final** = 0.25 * (8 + 8 + 6.8) / 3 + 0.75 * (3 + 7) / 2 = **5.65 ~ 6**
  - Laura
    - Tareas
      - **T1** aprueba con **8**
      - **T2** aprueba con **8**
      - **T3** superada con **6.8**
    - **Examen 1ra** convocatoria suspendido con un 3
      - **Examen 2da** convocatoria → **4.9**
    - **Nota final** = 0.25 * (8 + 8 + 6.8) / 3 + 0.75 * (3 + 4.9) / 2



Planificación de sesiones, tareas y exámenes

FORMACIÓN PROFESIONAL MONTECASTELO

CALENDARIO CICLO SUPERIOR 2º CICLO SUPERIOR DESARROLLO APLICACIONES WEB CURSO 22-23

septiembre 2022

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

octubre 2022

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

noviembre 2022

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

diciembre 2022

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

enero 2023

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

Sesión Teorica
Tarea
Examen

Horario Clases	
Lunes	17:00 a 20:00
Jueves y Viernes	17:00 a 20:00

** Los horarios de los exámenes de Junio serán comunicados por los profesores con la suficiente antelación.*

Planificación de sesiones, tareas y exámenes

- Tarea 1:
  - Fecha de publicación: jueves 6 de Octubre de 2022
  - Fecha de entrega: 23:59 del jueves 27 de Octubre de 2022
  - Fecha final de prórroga: 23:59 del domingo 30 de Octubre de 2022
  - Contenido: Tema 1 (Introducción)
- Tarea 2:
  - Fecha de publicación: jueves 3 de Noviembre de 2022
  - Fecha de entrega: 23:59 del jueves 24 de Noviembre de 2022
  - Fecha final de prórroga: 23:59 del domingo 27 de Noviembre de 2022
  - Contenido: Tema 2 (CSS básico) y Tema 3 (CSS Avanzado)
- Tarea 3:
  - Fecha de publicación: jueves 1 de Diciembre de 2022
  - Fecha de entrega: 23:59 del jueves 15 de Diciembre de 2022
  - Fecha final de prórroga: 23:59 del domingo 18 de Diciembre de 2022
  - Contenido: Tema 3 (Principios de diseño y personalidades web) y Tema 4 (Bootstrap)

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-22
- 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. Los 3 modos de construir diseños (*Layouts*)

3. **CSS Avanzado (~ 3 sesiones)**
   - *3.1.* Propiedad *Float*
   - 3.2. Módulo *Flexbox*
   - 3.3. Uso del Grid CSS
   - 3.4. Animaciones y transiciones
   - *3.5.* *Media queries*
   - 3.6. µSeminario - ¿Cómo funciona CSS?

4. **Principios de diseño y personalidades web (~ 2 sesiones)**
   - 4.1. Tipografía
   - 4.2. Colores
   - 4.3. Imágenes e ilustraciones
   - 4.4. Iconos
   - 4.5. Sombreados
   - *4.6.* *Border-radius*
   - 4.7. Espaciado
   - 4.8. Jerarquía Visual
   - 4.9. Experiencia de usuario (*UX*)
   - 4.10. Elementos y componentes
   - 4.11. Patrones de diseño

5. **Introducción a *Bootstrap* (~ 1 sesión)**
   - 5.1. ¿Qué es Bootstrap(5)?
   - 5.2. *Breakpoints*
   - *5.3.* *Containers*
   - *5.4.* *BS Grid*
   - 5.5. Formularios
   - 5.6. Componentes
     - 5.6.1. Botones
     - *5.6.2.* *Dropdowns*
     - *5.6.3.* *Cards*
     - 5.6.4. *Navbar*, *navs* y *tabs*

6. **Introducción a SASS (~ 1 sesión)**
   - 6.1. ¿Qué es SASS?
   - 6.2. Variables y anidamiento
   - 6.3. *Mixins*, *Extends* y *Functions*

Tema 1: Introducción

Asignatura: Diseño de Interfaces Web

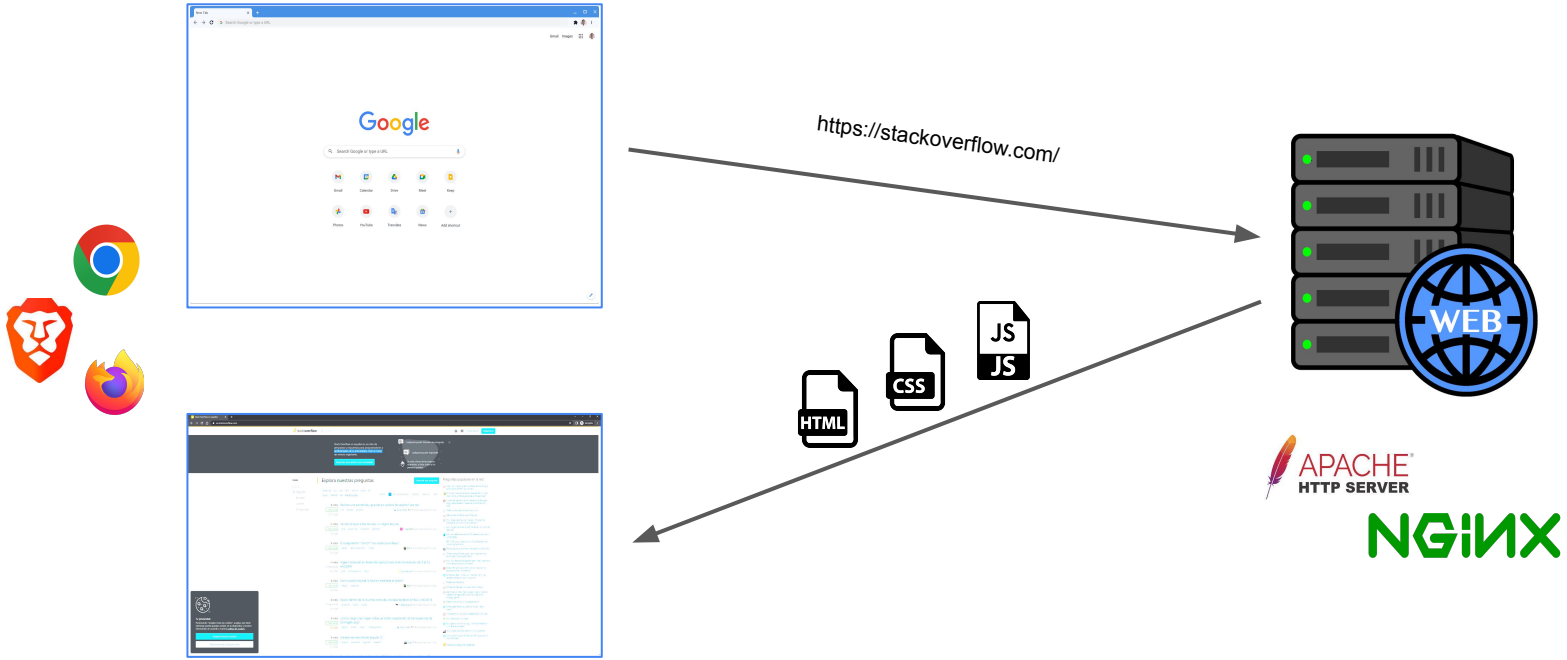
Contenidos

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



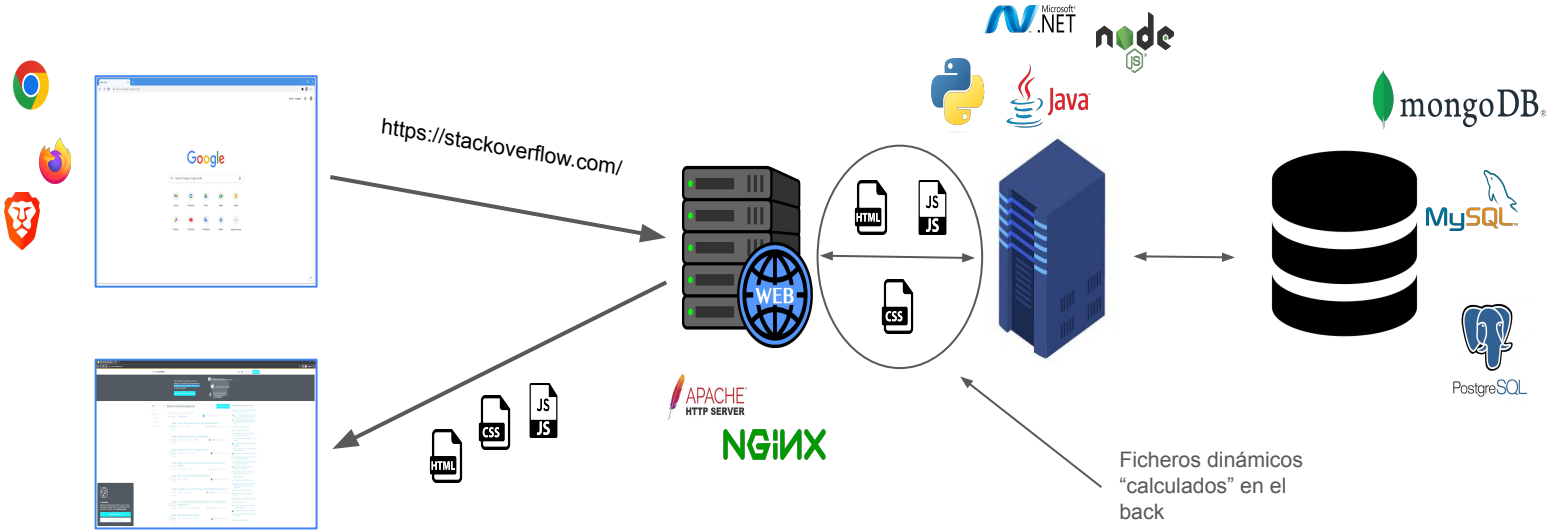
Desarrollo front-end vs back-end

- 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



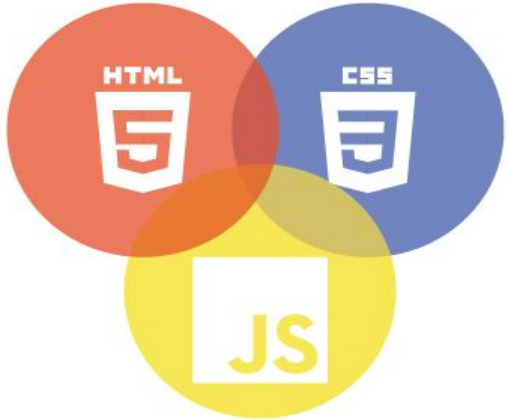
Desarrollo front-end vs back-end

- 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

CONTENIDO



PRESENTACIÓN

EFECTOS DINÁMICOS Y APLICACIONES WEB

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

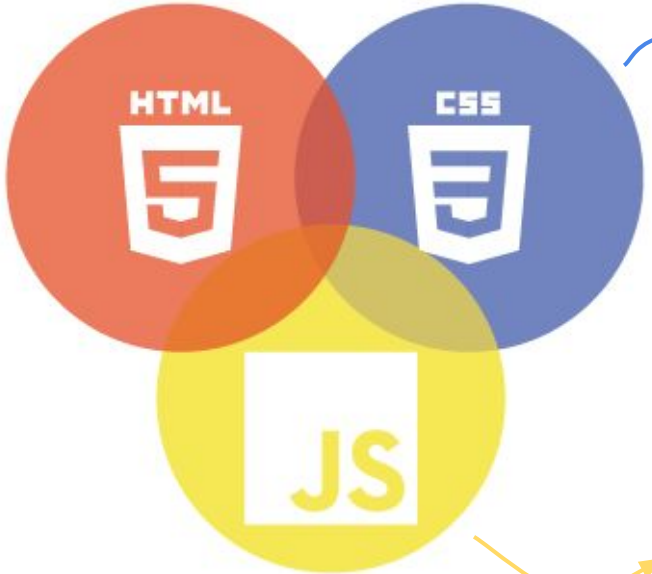
Los 3 lenguajes del front-end

NOMBRE

significa “párrafo”

CONTENIDO

PRESENTACIÓN



ADJETIVO

p {color: blue}

significa “el color del párrafo es azul”

VERBO

p.hide()

significa “esconde el párrafo”

EFECTOS DINÁMICOS Y APLICACIONES WEB

Los 3 lenguajes del front-end

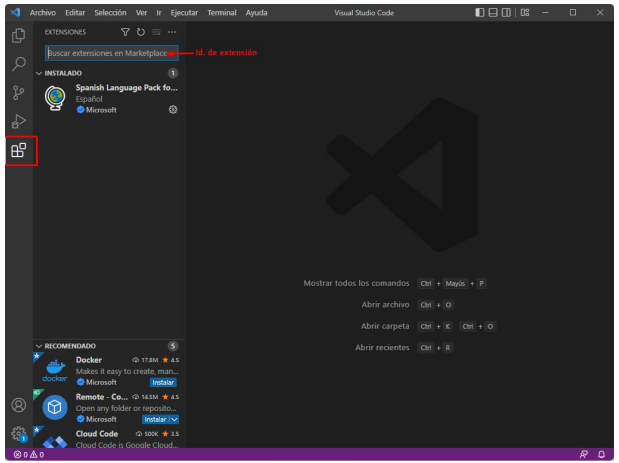
- Para construir un sitio o aplicación web se utiliza una combinación de HTML, CSS y Javascript
- El desarrollo front-end moderno se realiza utilizando “frameworks”, que son paquetes de software que proporcionan módulos de código reutilizable, tecnologías de front-end estandarizadas y bloques o componentes 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 un 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

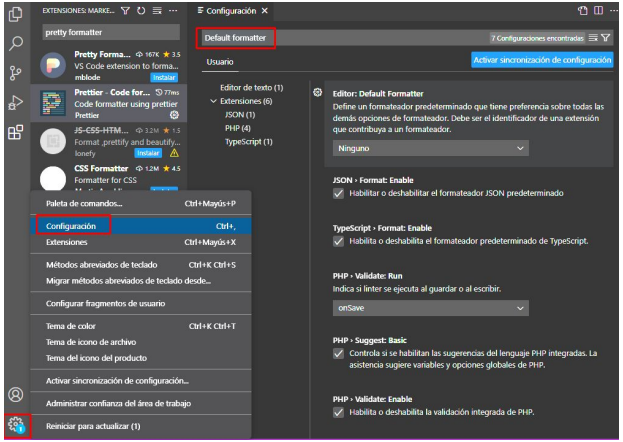
Configurando el editor de código

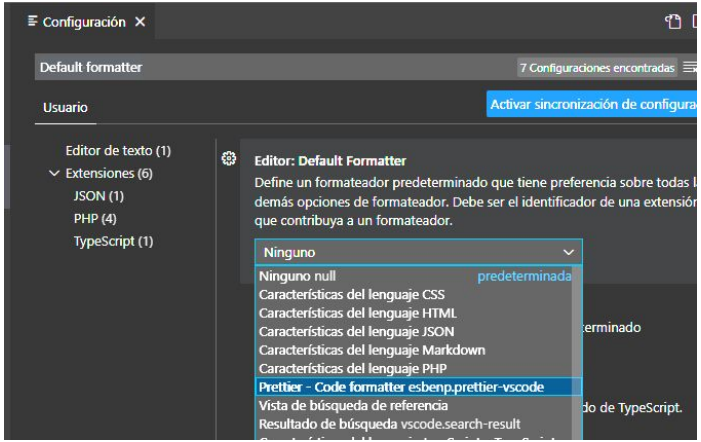
- Descargar e instalar VSCode desde su web oficial. Dispone de versiones para los SO principales.
  - Dispone de una versión de navegador (con limitaciones respecto de la versión de escritorio), indicada para desarrollos rápidos o desde los repositorios de Azure
- Ejecutar VSCode e instalar la siguiente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



Configurando el editor de código

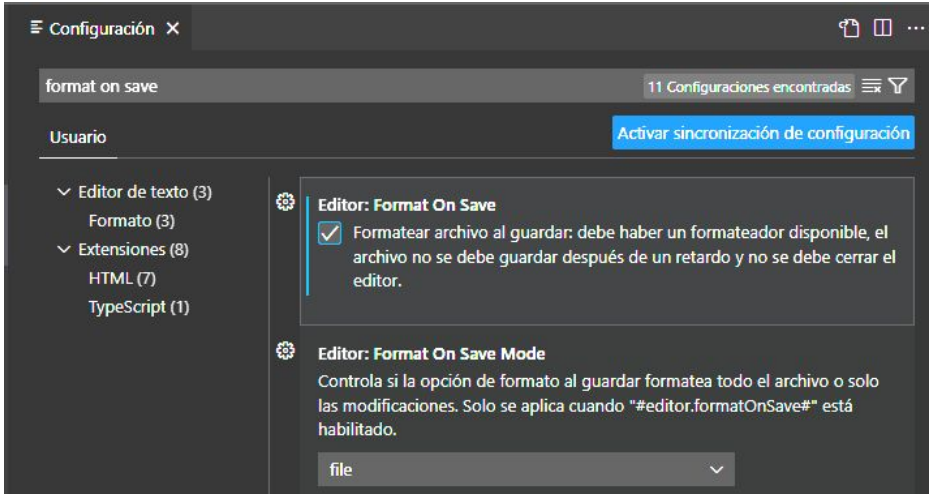
- ***Prettier Code Formatter*** - Formateo automático del código. Mantiene los proyectos limpios, uniformes y coherentes, a pesar del formato “opinionado”
  - Id. de extension - ***esbenp.prettier-vscode***
- Configuración
  - Establecer *Prettier* como formateador por defecto (*default formatter*)





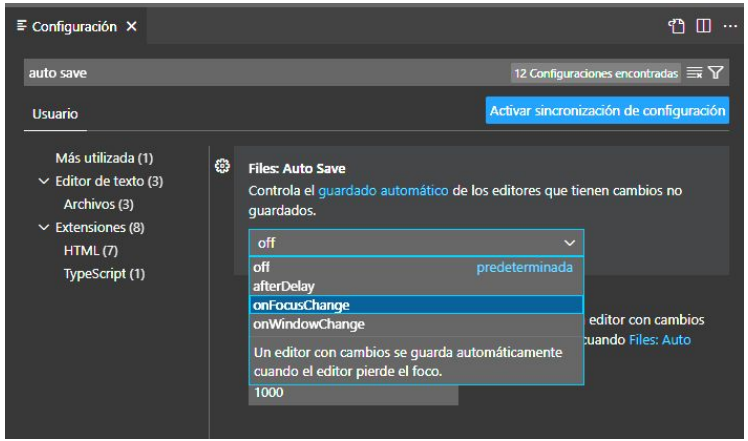
Configurando el editor de código

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



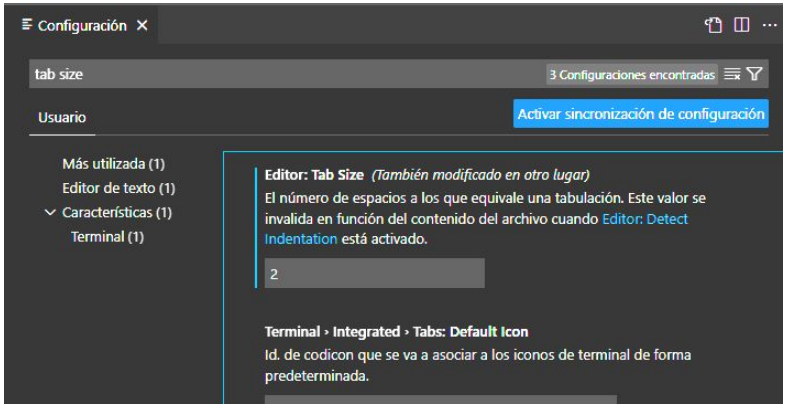
Configurando el editor de código

- 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*)



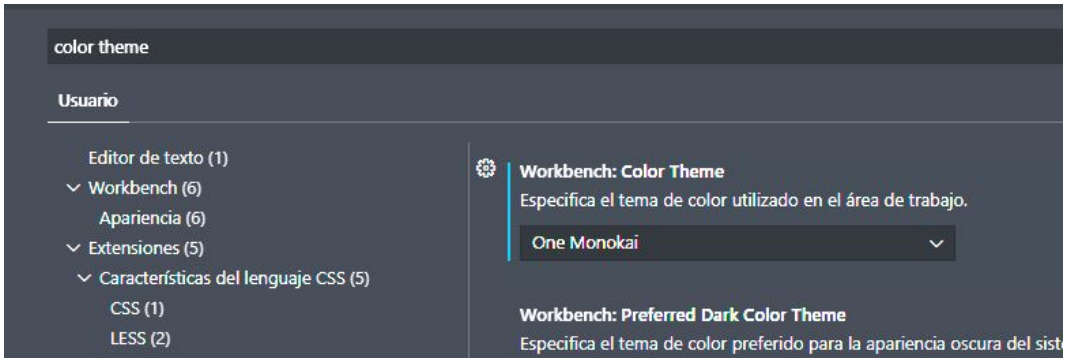
Configurando el editor de código

- 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 extension - ***azemoh.one-monokai***
    - Configuración (*color theme*)



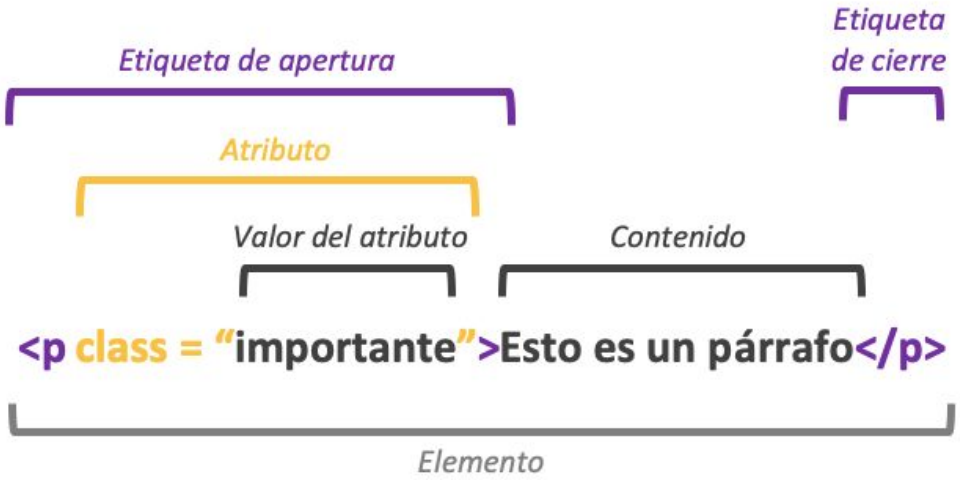
Configurando el editor de código

- 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 Server -*** Servidor web embebido en el editor
    - Id. de extension - ***ritwickdey.LiveServer***
- Activar el cierre automático de tags
  - Configuración > ***auto closing tags*** > Habilitar el *checkbox*

Un breve repaso a HTML(5)

- **H**yper**T**ext **M**arkup **L**anguage
- 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://gitfront.io/r/tcamba/2MCVy6P2LQu8/diw-22.git
  - … 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 Flexbox CSS Grid

The Basic Language of the Web: HTML

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

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 dev language
- To hand-craft beautiful websites instead of 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 Jonas Schmedtmann
- The Unknown Powers of CSS
  By Jim Dillon
- Why JavaScript is Awesome
  By Matilda

Copyright © 2027 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 imagenes -->
    <img
      src="assets/images/laura-jones.jpg" <!-- Ruta al fichero con la imagen. NO USAR RUTAS ABSOLUTAS -->
      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 parrafos →
  <!-- <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)

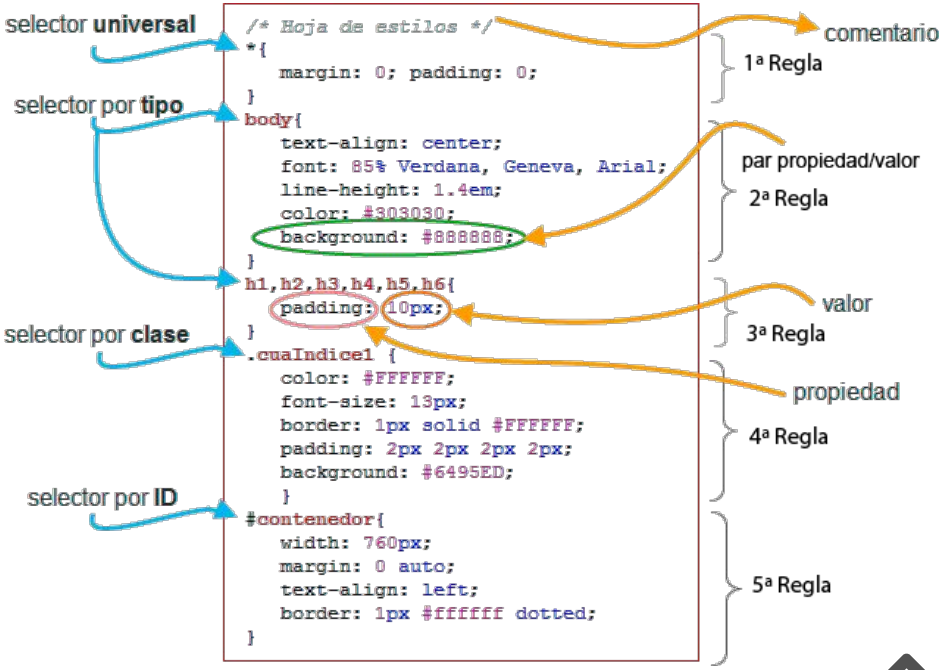
```
<p>
  All modern websites and web applications are built using three
  <em>fundamental</em> <!-- <em> Aplica énfasis (cursiva o itálica) a su contenido. Preferible a <i> por razones semánticas
-->
  technologies: HTML, CSS and JavaScript. These are the languages of the
  web.
</p>

...

<!-- 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>
      <img
        src="img/related-1.jpg"
        alt="Person programming"
        width="75"
        width="75"
      />
      <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 &copy; 2027 by The Code Magazine.</footer> <!-- &copy es una entidad HTML que renderiza al símbolo © →
```

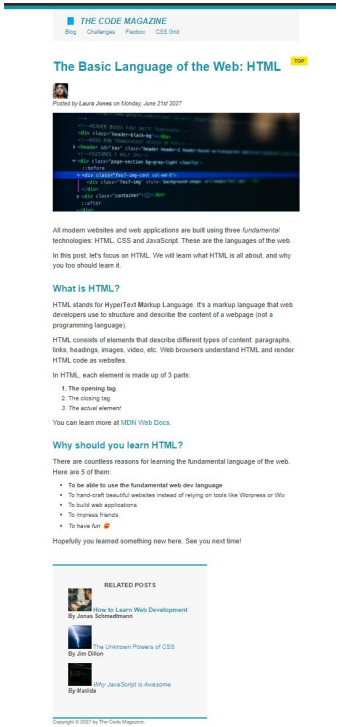
Una pequeña introducción a CSS(3)

- **C**ascading **S**tyle **S**heets
- **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, espaciado, 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://gitfront.io/r/tcamba/2MCVy6P2LQu8/diw-22.git
  - … 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



Una pequeña introducción a CSS(3)

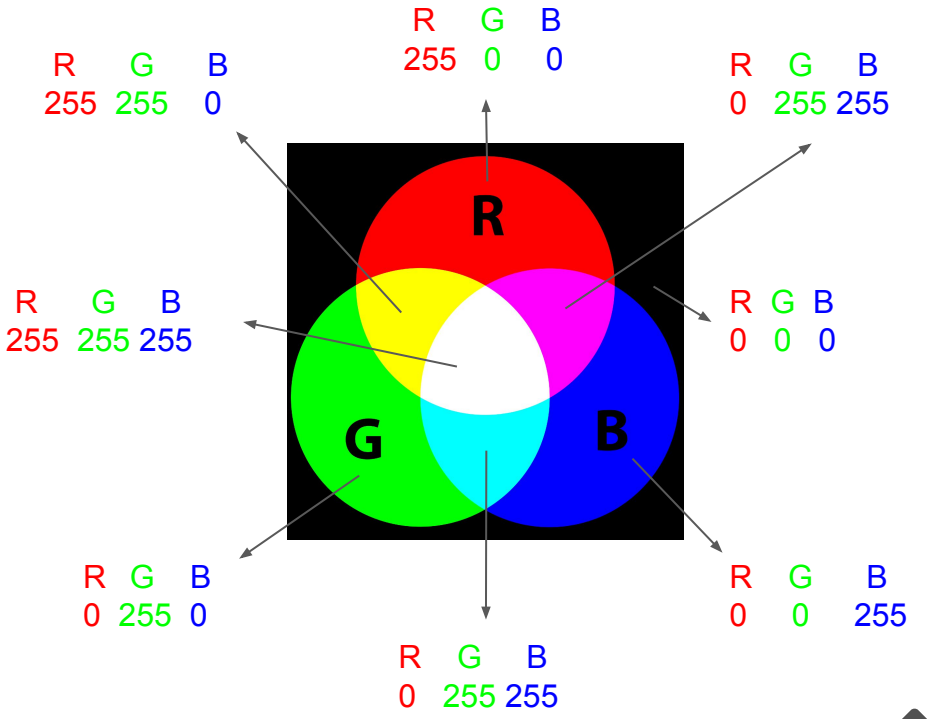
```
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <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>


    ...
```

Una pequeña introducción a CSS(3)

```css
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 espaciado 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)

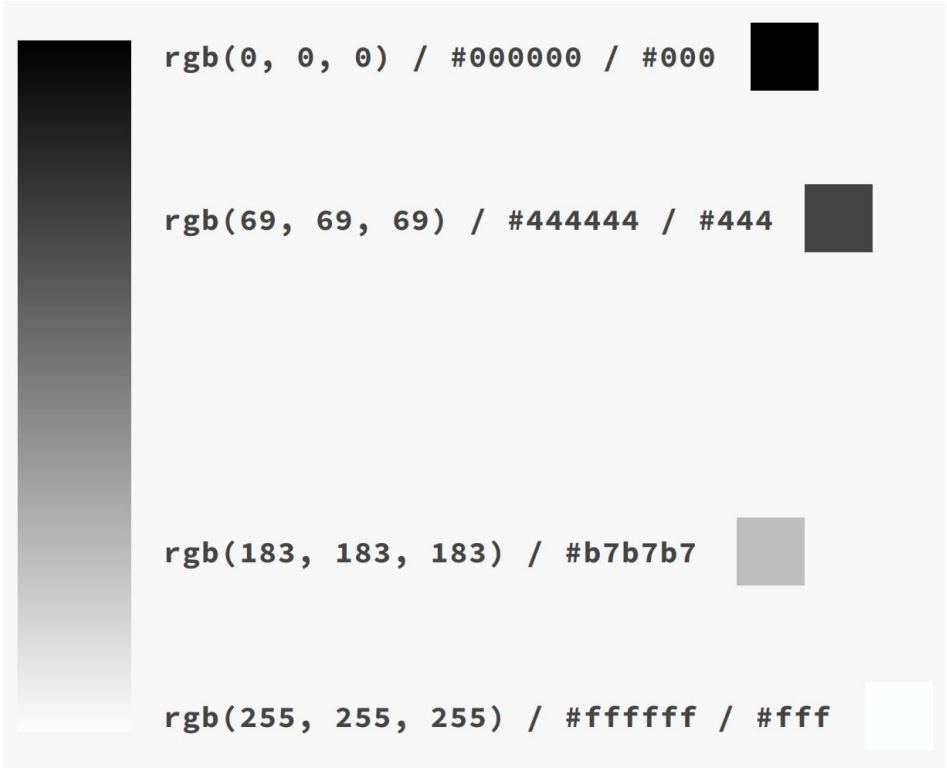
- Notación Hexadecimal
  - En lugar de usar una escala de 0 a 255, vamos desde **0 a FF** (hexadecimal)
    - **#00ffff** 
  - Cuando todos los colores son pares idénticos se acorta a
    - **#0ff**

#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

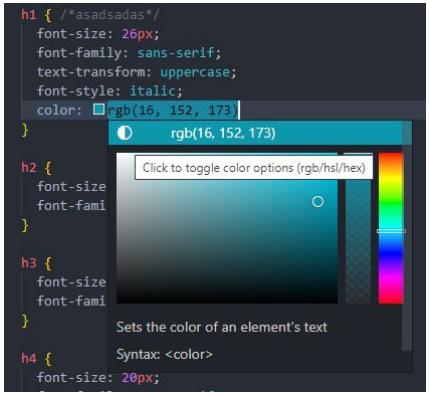


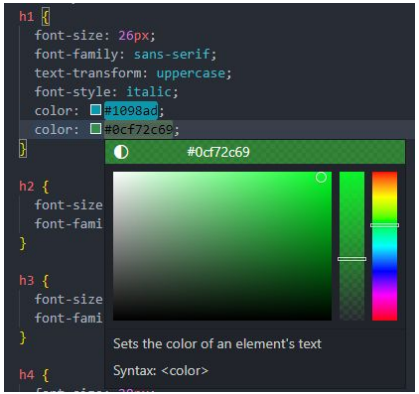
Colores

- 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;
}
```

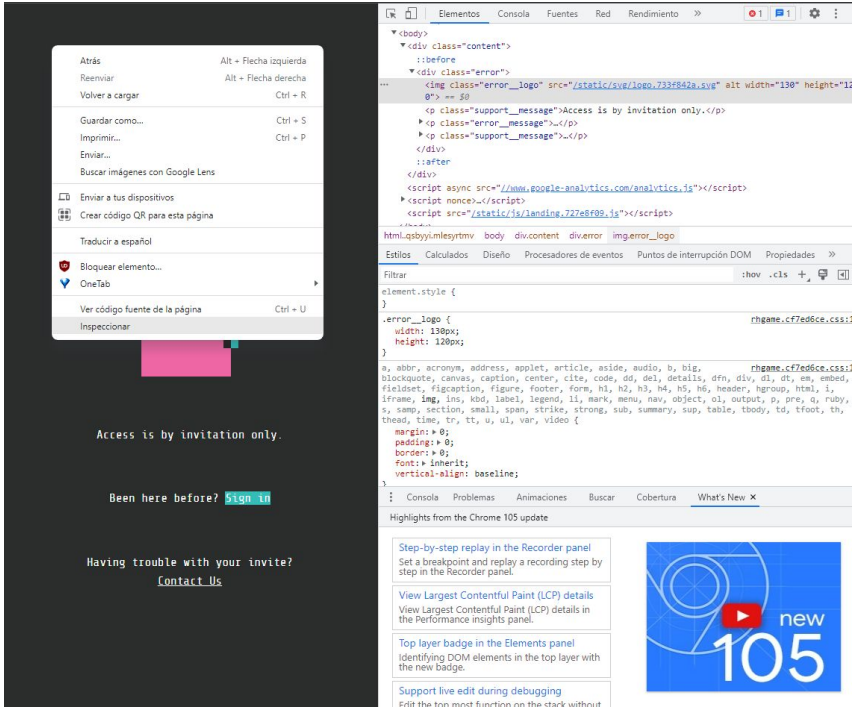
- El último en ser declarado





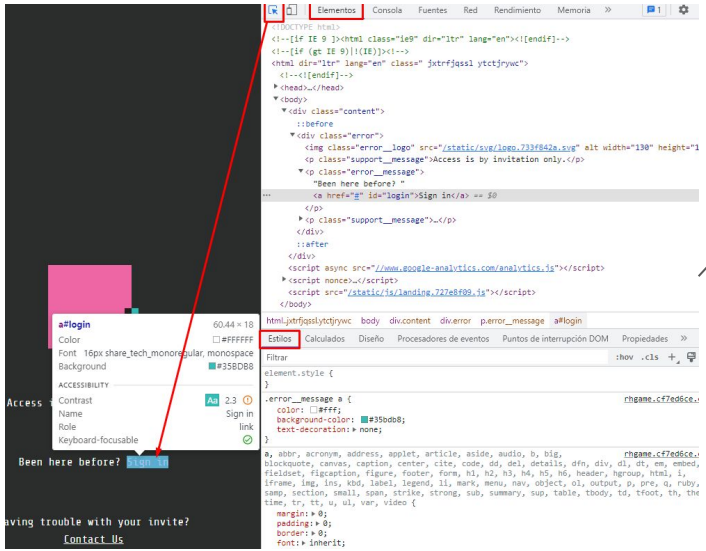
Herramientas de desarrollador

- 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



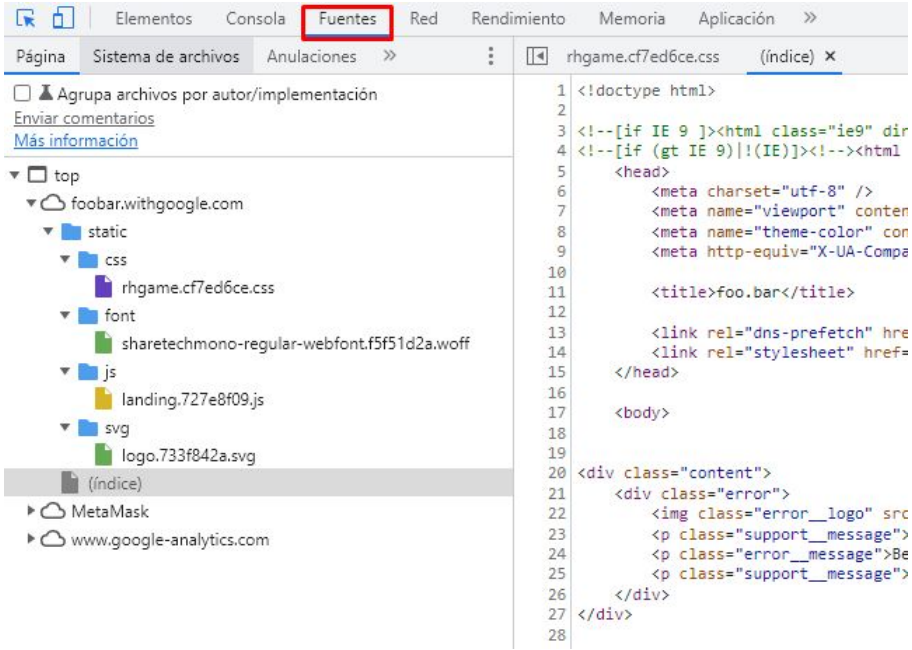
Herramientas de desarrollador

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



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…



Referencias

- Visual Studio Code
  - Emmet - *Cheatsheet* de Emmet. Emmet es una herramiento 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 todas 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
- Herramientas de desarrollador
  - Chrome DevTools Docs - Documentación oficial de las herramientas de desarrollador de Chrome

Referencias

- Guía de contenido de las grabaciones del tema