

HTML & CSS



Método evaluación: Apto / No Apto

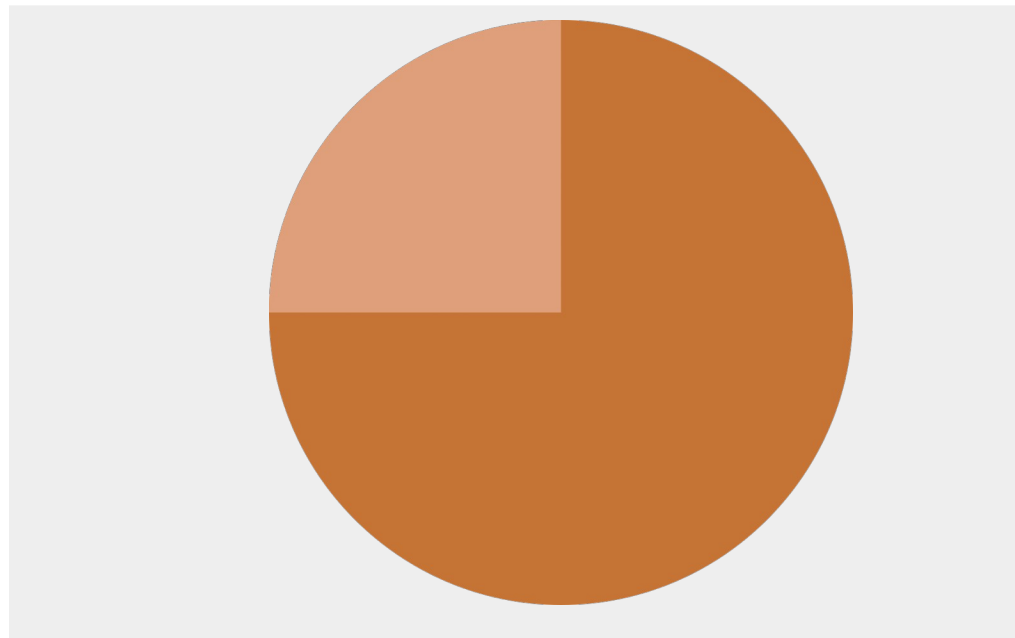
75% Asistencia obligatoria
Entregas obligatorias
Test escritos

Recursos

Moodle: Documentos clase, documentos extendidos, cheatsheets

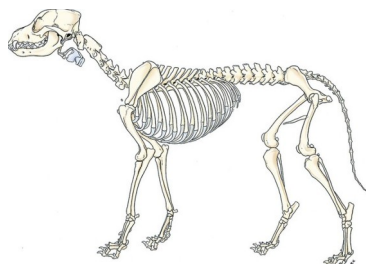
[MDN_ en ingles \(recomendado\)](#), [MDN_ en castellano](#)
[stack overflow](#) (resultados de google)

IA -> Último recurso, de momento...



Conceptos iniciales:

HTML



Estructura

CSS



Estilo

JS



Funcionalidad

Qué es HTML y quién define el estándar

Acrónimo: HyperText Markup Language.

Versión actual:



Organismos responsables:

World Wide Web Consortium (W3C) w3.org/w3.org — publica recomendaciones oficiales.

WHATWG whatwg.org — mantiene el Living Standard que evoluciona continuamente.

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#).)

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#), etc.

Primera web: <http://info.cern.ch/hypertext/WWW/TheProject.html>

Preparación entorno de desarrollo:

Instalar VS Code: <https://code.visualstudio.com/>

Crear estructura de carpetas:
html>clase1>index.html, css y scripts

Instalar plugins:

- ✓ Prettier (activar formatear al guardar)
- ✓ ESLint
- ✓ Live server (Ritwick Dey)
- ✓ GitLens (itKraken)
- ✓ Error lens (Alexander)

Revisamos DevTools (F12)



HTML: A romper cosas

Pairing!

- Uno escribe, el otro es "**copilot**"
- En 10 min cambiamos, en el **mismo portatil**
- Empezamos escribiendo ! **e intro** en el archivo index.html, en VS Code
- Hacemos click en **Go Live** (abajo derecha en VS Code)
- Abrimos **DevTools SIEMPRE!!!** (F12)



HTML: Estructura de un documento

- `<!DOCTYPE html>` – indica al navegador que el documento sigue la especificación HTML 5.
- `<html>` – elemento raíz que engloba todo el documento
- `<head>` – contiene metadatos, enlaces a recursos externos (hojas de estilo, scripts, fuentes) y configuraciones de la página
- `<body>` – alberga todo lo que se muestra al usuario (texto, imágenes, formularios, etc.)

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Título de la página</title>
  <!-- Metadatos, enlaces a CSS/JS, favicons, etc. -->
</head>
<body>
  <!-- Contenido visible para el usuario -->
</body>
</html>
```

Sintaxis y estructura de los elementos HTML

- Etiquetas (tags): <etiqueta> de apertura y </etiqueta> de cierre.
- Etiquetas auto-cerrables (self-closing): ,
, <meta ... />, etc.
- Atributos: nombre="valor" – añaden información a la etiqueta (p. ej., href, src, alt).
- Contenido: texto o nodos hijos que aparecen entre la apertura y el cierre.
- Comentarios: <!-- Comentario --> – útiles para documentación interna y pruebas. En VS Code, Command + /
- Indentación y formato: mantener una jerarquía visual clara facilita la lectura y el mantenimiento; Prettier y ESLint ayudan a automatizarlo.
- Tolerancia del navegador: los navegadores intentan renderizar la página pese a errores de marcado, pero los fallos pueden producir resultados inesperados.

Etiqueta apertura <p>

Contenido “ ¡ Quiero viajar !

Etiqueta cierre </p>

Elemento <p> ¡ Quiero viajar ! </p>

```
<p> ¡ Quiero viajar ! </p>
```


HTML: Metadatos en head

La sección `<head>` contiene información vital para el navegador y los motores de búsqueda, pero que no se muestra directamente en la página

```
<head>
  <!-- Define la codificación de caracteres. UTF-8 es el estándar. -->
  <meta charset="UTF-8">

  <!-- Asegura que la página se vea bien en todos los dispositivos (responsive). -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- El título que aparece en la pestaña del navegador. -->
  <title>Mi Primera Página Web</title>

  <!-- Se pueden incluir estilos CSS directamente aquí (aunque no es lo ideal). -->
  <style>
    body {
      font-family: sans-serif;
    }
  </style>
</head>
```

Copiar

HTML: Etiquetas en <body>

Estas son algunas de las etiquetas más comunes que usarás para construir el contenido visible de tus páginas.

- <h1> a <h6>: Encabezados, siendo <h1> el más importante y <h6> el menos.
- <p>: Párrafo de texto.
- <a>: Ancla o enlace (hipervínculo).
- : Imagen.
- <div>: Contenedor genérico en bloque, usado para agrupar otros elementos y aplicarles estilos.
- : Contenedor genérico en línea, usado para agrupar texto o elementos dentro de un bloque.
- <button>: Botón interactivo.
-
: Salto de línea.

```
<!-- Párrafo -->
<p>Este es un párrafo de ejemplo.</p>

<!-- Enlace -->
<p>Visita <a href="https://developer.mozilla.org/es/" target="_blank">MDN</a>
para más información.</p>

<!-- Imagen -->


<!-- Contenedor de bloque -->
<div style="border:1px solid #4a90e2; padding:0.5rem;">
  <p>Contenido dentro de un div.</p>
</div>

<!-- Contenedor en línea -->
<p>Texto con <span style="background:#ffffae6;">resaltado usando span</span>.
</p>

<!-- Botón -->
<button type="button" onclick="alert('Botón pulsado!')">
  Pulsar
</button>

<!-- Salto de línea -->
<p>Línea 1<br>Línea 2 después del br</p>
</body>
```

HTML: Buenas prácticas anidamiento

Anidamiento etiquetas “inline”

```
<p> Lo <strong>más importante</strong> es sonreír. </p>
```

Anidamiento etiquetas “block”

```
<div>
  <div>
    <div>
      <div>
        <!-- contenido -->
      </div>
    </div>
  </div>
</div>
```

HTML: Atributos

Los atributos nos permiten añadir información y configuración específica a los elementos HTML. Son fundamentales para CSS y JavaScript.

```

```

```
<a href="https://www.pamplona.es/" title="Página web del ayto"> Haz click en este enlace! </a>
```

Atributos de identificación:

```
<p class="fondo_verde" id="primer_parrafo"> Primer párrafo. </p>
```

```
<p class="fondo_verde"> Segundo párrafo.</p>
```

HTML: DOM

- Document Object Model
- El **navegador** crea su **DOM** de la página según su interpretación* de las etiquetas
- Arbol de **elementos**

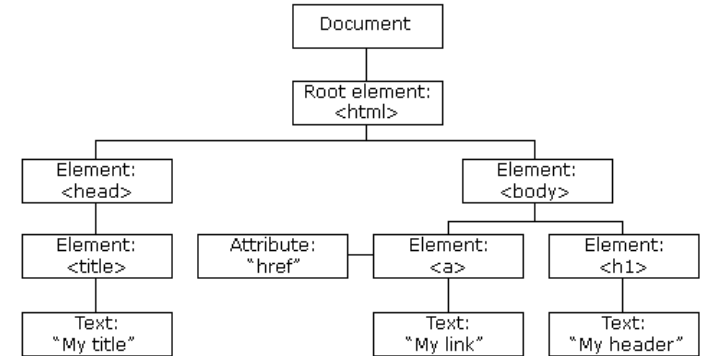
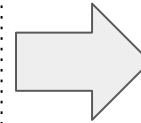
```
<html>

<head>
  <title> My title </title>
</head>

<body>
  <a href="www.mylink.link"> My link </a>
  <h1> My header </h1>
</body>

<!-- mi comentario -->

</html>
```



HTML: Cargar Scripts de JavaScript

Para añadir interactividad a nuestra página, usamos la etiqueta `<script>`. Lo más común es enlazar a un archivo externo.

¿Dónde ponerla?

La mejor práctica es colocar la etiqueta `<script>` dentro de `<head>` **incluyendo el atributo "defer"**.

Otra opción es colocar la etiqueta `<script>` justo antes del cierre de la etiqueta `</body>`. Puede ser útil para soportar navegadores antiguos que no soportan "defer".

¿Por qué? Esto asegura que el navegador haya cargado y construido todo el contenido HTML de la página (el DOM) antes de que el script de JavaScript intente acceder a él o manipularlo. Si el script se carga antes, podría intentar modificar un elemento que todavía no existe, causando errores.

```
<script src="mi-script.js"></script>
```

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo con script defer</title>

  <!-- Hoja de estilos opcional -->
  <link rel="stylesheet" href="styles.css">

  <!-- Script cargado de forma diferida -->
  <script src="app.js" defer></script>
</head>

<body>
  <h1>Demo de script con defer</h1>
  <p>El contenido de la página se muestra antes de que se ejecute el archivo JavaScript.</p>

  <!-- Otros elementos visibles -->
  <button id="miBtn">Haz clic</button>
</body>
</html>
```

HTML: Introducción DOM

Cuando un navegador carga un documento HTML, no lo lee como simple texto. Lo convierte en una estructura de datos en memoria llamada DOM (Document Object Model).

El DOM representa la página como un árbol de nodos, donde cada etiqueta, atributo y texto es un nodo.

Esta estructura en árbol permite JavaScript acceder, modificar, añadir o eliminar cualquier elemento de la página de forma dinámica.

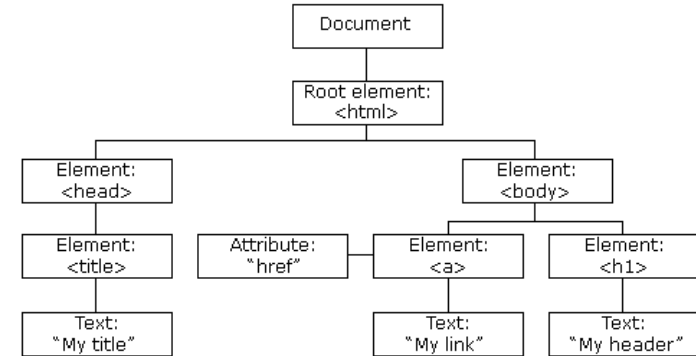
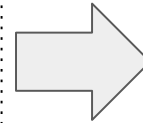
```
<html>

<head>
  <title> My title </title>
</head>

<body>
  <a href="www.mylink.link"> My link </a>
  <h1> My header </h1>
</body>

<!-- mi comentario -->

</html>
```



HTML: El Objeto `document`

JavaScript nos proporciona un objeto global especial para interactuar con el DOM: [el objeto document](#). Este objeto es el punto de entrada a toda la estructura de la página.

Funciona como cualquier otro objeto de JavaScript, permitiéndonos acceder a sus propiedades y métodos mediante la notación de punto.

```
// Acceder a la propiedad 'title' del objeto 'document'  
console.log(document.title);  
  
// Usar el método 'getElementById' para encontrar un elemento  
const miElemento = document.getElementById('mi-id');
```


HTML: El Objeto `document`

JavaScript nos proporciona un objeto global especial para interactuar con el DOM: [el objeto document](#). Este objeto es el punto de entrada a toda la estructura de la página.

Funciona como cualquier otro objeto de JavaScript, permitiéndonos acceder a sus propiedades y métodos mediante la notación de punto.

```
// Acceder a la propiedad 'title' del objeto 'document'  
console.log(document.title);  
  
// Usar el método 'getElementById' para encontrar un elemento  
const miElemento = document.getElementById('mi-id');
```

HTML: Modificar el HTML desde Javascript

Utilizando los metodos y las propiedades del objeto document podemos seleccionar los nodos que nos interesen del DOM, asignarlos a un variable, modificarlos o crear nodos nuevos, y volver a inyectarlos en el HTML.

```
<h1 id="saludo">Hola</h1>
<button id="miBoton">Cambiar Saludo</button>
```

```
<script>
  // 1. Seleccionar los elementos del DOM
  const titulo = document.getElementById('saludo');
  const boton = document.getElementById('miBoton');

  // 2. Escuchar el evento 'click' en el botón
  boton.addEventListener('click', () => {
    // 3. Manipular el contenido del h1
    titulo.textContent = '¡Adiós, Mundo!';
  });
</script>
```

HTML: *Esquema de Referencias Básicas para HTML*



1. Mozilla Developer Network (MDN)

- **Descripción:** MDN Web Docs es un recurso mantenido por Mozilla que ofrece documentación extensa y ejemplos sobre HTML, CSS, JavaScript, y otras tecnologías web. Es muy popular entre desarrolladores por su claridad y amplitud.
- **Enlace General:** [MDN Web Docs](https://developer.mozilla.org/)
- **Elementos & Atributos HTML:**
 - [Elementos HTML en MDN](#)
 - [Atributos HTML en MDN](#)

2. Stack Overflow

Descripción: Stack Overflow es una comunidad de desarrolladores donde se pueden encontrar respuestas a preguntas específicas sobre HTML, CSS, y JavaScript. Es útil para resolver problemas y aprender de la experiencia de otros.

Enlace: [Stack Overflow](https://stackoverflow.com/)