



iPo

# HTML - CSS - JS



Plantilla de los proyectos

- index.html
- style.css
- script.js



World Wide Web Consortium  
[W3C: Página Inicio](#)



MDN: [Mozilla Developer Network](#)

- Punto de partida: [MDN: Web](#)
- HTML: [MDN: HTML](#)
- CSS: [MDN: CSS](#)
- Guía JS: [MDN: Guía JS](#)
  - Web APIs: [MDN: Web APIs](#)
  - DOM: [MDN: DOM](#)
  - CSSOM: [MDN: CSSOM](#)
  - Eventos: [MDN: Eventos](#)



CSS-Tricks:  
<https://css-tricks.com/>



**web.dev** <https://web.dev/>

**CAN I USE:** <https://caniuse.com/>



CodePen:  
<https://codepen.io>

# Herramientas



## Visual Studio Code (VSCode)

Extensiones (recomendadas):



Emmet (<https://emmet.io/>)  
(Preinstalada en VSCode)



Live Server ([Ritwick Dey](#))



Prettier (<https://prettier.io/>)

- Motores Web
  - Blink / (Chrome, Edge, Brave)
  - Gecko / Firefox
  - Webkit / Safari
- Herramientas de desarrollo
  - Chrome DevTools
  - Firefox Developer Tools





Meta contenido

`html head body meta title link script ...`

Continentes

Textual

`h1 h2 h3 h4 h5 h6 p ul ol li ...`

Multimedia

`img canvas audio video map ...`

Interacción

`a button input select option label ...`

Decoradores

`span i ...`

*Pseudoelementos CSS*

`::after ::before`

Contenedores

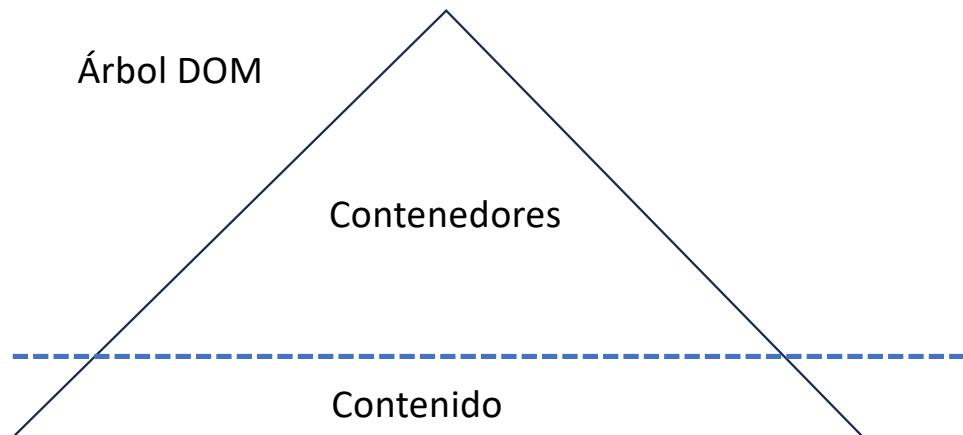
`div` (header footer nav main article section aside)  
`form`

`<div></div>` solo para crear textos, elementos.

`<a>`

`<button>` para casos precisos

Árbol DOM



Nodos del tronco



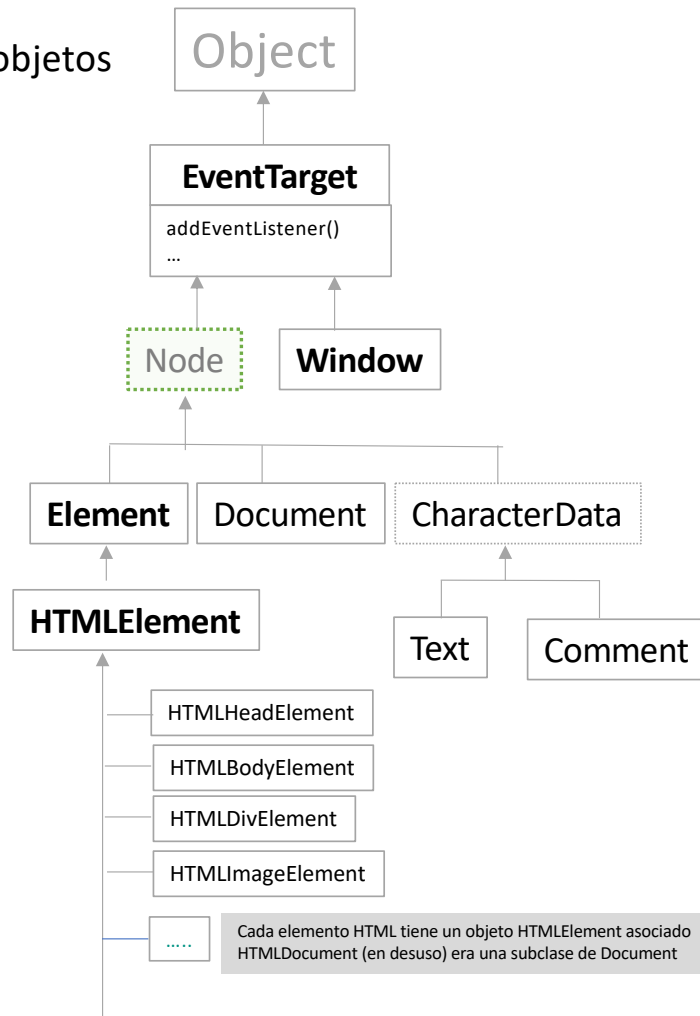
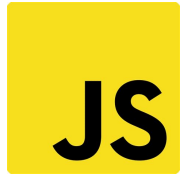
Hojas

Pilares (Pisos)



Ladrillos (Paredes)

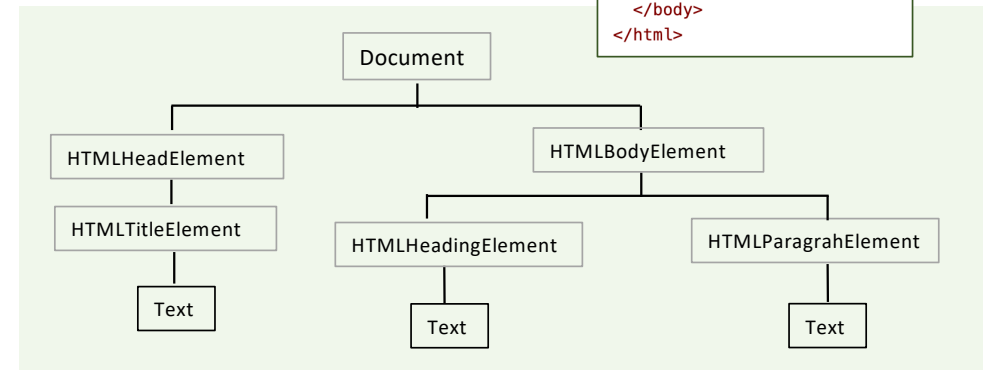
## Jerarquía de objetos



## Web APIs

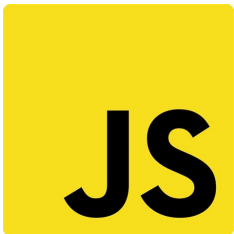
- HTML DOM API
- UI Events API
- Drag and Drop API
- Web Audio API
- Canvas API
- SVG API
- Web Components

## DOM



## Buenas prácticas

- Incluir JS mediante el elemento `<script>` en la cabecera `<header>` de la página
  - Con atributo `type="module"` o `defer` según se vaya a hacer uso o no de importación de scripts
- Definir interacciones mediante `EventListener` (no mediante atributos HTML `onXXX`)
- Enlazar JS con elementos HTML mediante atributos `id`. Para seleccionar conjuntos: `class` o `data-*`
- Separar el aspecto de la interacción: `.classList` vs `.style`
- Usar con `let` o `const` en las declaraciones (no `var`)



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Documento</title>
</head>
<body>
  <p class="mensaje"></p>      usar <id>
  <p class="mensaje"></p>
  <p>Contador: <span id="contador">0</span></p>
  <button onclick="incrementarContador()">Incrementar</button>      usar
                                                                    EventListener
  <script>
    var contador = 0;
    function incrementarContador() {
      contador++;
      document.getElementById('contador').textContent = contador;
    }
  </script>
  <script src="script.js"></script>      no meter los scripts al
                                                                    final
</body>
</html>
```

```
// script.js

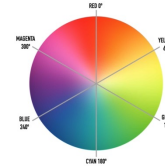
var mensajes = document.querySelectorAll('.mensaje');

mensajes.forEach((element, index) => {
  element.textContent = "Hola texto " + (index + 1);
  element.style.color = 'blue';
});
```



## COLOR

color  
background-color



Sistemas de color: `rgb()`, `hsl()`, ...

## TIPOGRAFÍA

font-{family, size, weight}  
text-{decoration, align, transform}  
line-height



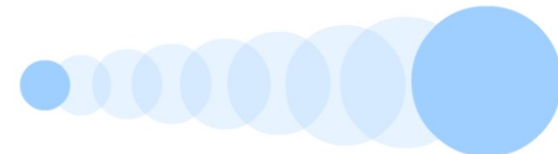
## ESPACIO

DIMENSIÓN	width height
CAJA	padding margin border box-sizing
COMPOSICIÓN	display : <code>block</code>   <code>inline</code>   <code>flex</code>   <code>grid</code>
POSICIONAMIENTO	position, top, right, left, bottom, z-index

Unidades: `px`, `em`, `rem`, `%`, `vw`, `vh`, ...

## MOVIMIENTO

transition transform animation



# Buenas Prácticas



- Usar hojas de estilos enlazadas con `<link>` en `<header>` (evitar elemento `<style>`)
- Concentrar en la regla `:root` el diseño cromático y tipográfico de la página
- Usar preferentemente **clases** y/o **notación BEM** para los selectores de las reglas
- Usar el sistema **hsl** para nombrar colores. Evitar nombres (red, blue) o **rgb**.
- Minimalismo: evitar propiedades innecesarias
- Componer la página (nodos intermedios) con **flexbox/grid**.
  - Limitar el flujo normal (**block+inline**) a las hojas del árbol DOM
- Usar unidades apropiadas según el tipo de contenido:
  - Textual: **em**, **rem**
  - Multimedia: **px**, **%**, **clamp(min, preferente, max)**,
  - Evitar **px** en otros casos, salvo para medidas secundarias y pequeñas
- Ser cautos a la hora de crear espacio al seleccionar entre **margin** y **padding**
- Evitar valores poco intuitivos (ej.: márgenes negativos)

Regla **reset** básica para unificar diseños entre navegadores

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
  /* outline: 1px solid red */  
}
```

La propiedad **outline** es similar a **border** pero dibuja el recuadro encima de la caja (**no** ocupar espacio).  
Es útil para depurar la composición pues muestra todas las cajas de la página



## Estrategia de composición

- **Dirección:** Ascendente (primero estiliza las hojas y continúa hacia arriba hasta el nodo raíz)
- **Flexibilidad:** Emplear en los nodos intermedios unidades proporcionales **%**, **vw**, **min()**, **max()**, **clamp(min, preferente, max)**
- **Minimalismo:** emplear el menor número posible de restricciones en las dimensiones (anchura, altura)
- **Orgánico:** identificar los componentes de la composición y asignar los espacios de forma coherente





```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <style>
    #titulo {
      color: blue;
      font-size: 24px;
    }
  </style>
  <h1 id="titulo">Hola Mundo</h1>
  <p>Este es un párrafo de ejemplo.</p>
  <div id="contenedor">
    <p class="texto">Uno</p>
    <p class="texto">Dos</p>
  </div>
</body>
</html>
```

```
/* style.css */

#contenedor {
  width: 80%;
  height: auto;
  border: 1px solid black;
  padding: 10px;
  background-color: lightgray;
}

.texto {
  --azul-osculo: #00008b;
  font-family: 'Courier New', monospace;
  color: var(--azul-osculo);
  font-size: 16px;
  line-height: 1.5;
  margin-bottom: 10px;
}
```

# CSS Notación BEM

Notación BEM (<http://getbem.com/>)

Los selectores son clases (**class**)

Los nombres de las clases usan notación *kebab-case* para los bloques, elementos y modificadores

NOTA: No se admiten subelementos (jerarquía plana)

**bloque\_\_elemento\_\_subelemento**

block-name\_\_elemento-name—modifier-name

Un **bloque** es una entidad significativa por sí sola

Un **elemento** es una parte de un bloque cuyo sentido es incompleto por sí solo

Un **modificador** es una característica (aspecto o comportamiento) de un bloque o elemento.

```
<nav class="menu">
  <ul>
    <li class="menu__item"><a href="#" class="menu__enlace menu__enlace--seleccionado"></a>Item 1</li>
    <li class="menu__item"><a href="#" class="menu__enlace"></a>Item 2</li>
    <li class="menu__item"><a href="#" class="menu__enlace"></a>Item 3</li>
  </ul>
</nav>
```



iPo