

# HTML + CSS

Algunas propiedades y el CSS Object Model

# Algunas propiedades de css

CSS significa Cascade Style Sheets, son una serie de propiedades que se le dan a los objetos del DOM y nos ayudan a darle estilos y algunas animaciones a los elementos, en esta clase vamos a ver algunas de esas, hablaremos de **especificidad** y les explicare con un ejemplo un poco de cómo podemos usar los archivos de css.

# Cómo estructuramos nuestro css

Para seleccionar elementos del HTML en nuestro CSS usamos el selector (la palabra que se ve azul en el ScreenShot), podemos seleccionar por tags, clases usando un punto antes de la clase o por IDs usando un numeral.

Después abrimos llaves y dentro de estas iran las propiedades que queremos usar en ese o esos elementos, en este caso a los tags section le estamos dando un alto y ancho de 100% de la pantalla.

```
section {  
    width: 100vw;  
    height: 100vh;  
}
```

# Unidades de medida en css

En css tenemos varias alternativas para darle medidas a los elementos, algunas se usan para valores absolutos como **PX**, **REM** y **EM** y otros para valores que dependen del contenedor o la pantalla como **%**, **VW** o **VH**

```
section {  
    width: 100vw;  
    height: 100vh;  
    font-size: 1.5rem;  
    margin-top: 2em;  
    padding: 1%;  
}
```

# Unidades de medida en css

- **PX** lo utilizamos para dar valores en píxeles, estos valores no cambiarán en la página (30px siempre van a ser 30px).
- **REM** es el valor del font-size del elemento root, es decir 1rem es el valor que tiene la propiedad font-size del html, en la mayoría de los navegadores esto equivale a 16px default.
- **EM** es el valor del font-size en el elemento donde es usado, es útil cuando estamos dando valores a espacios o márgenes de un elemento y queremos que sean basados en el tamaño de letra.
- **VW** es un valor computado que resulta de fraccionar el ancho de la pantalla en 100 partes iguales, es decir que 1vw es el 1% del ancho de la pantalla.
- **VH** es un valor computado que resulta de fraccionar el alto de la pantalla en 100 partes iguales, es decir que 1vw es el 1% del alto de la pantalla.

# Especificidad en css

La especificidad en css es la un valor que es usado para que el navegador decida qué propiedades debe aplicar a los elementos de la página, es importante tener siempre presente esto cuando creamos los estilos para así obtener el resultado deseado y evitar conflictos de estilos.

La especificidad es calculada de la siguiente forma:

- Selectores de tipo o por tag y pseudo-elementos.
- Selectores de clase, atributos y pseudo-clases.
- Selectores de ID

**Nota:** los estilos escritos en el tag dentro del html sobrescriben los de las páginas de estilos.

# Ejemplo de especificidad

Hola Clase, este es un ejemplo de especificidad

The screenshot shows a web browser's developer tools interface. The left pane displays the HTML structure, and the right pane shows the computed styles for the selected element, `h1#index-title.index-title`.

**HTML Structure:**

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    ...
    <h1 class="index-title" id="index-title">Hola Clase, este
    es un ejemplo de especificidad</h1> == $0
    <!-- Code injected by live-server -->
    <script type="text/javascript">...</script>
  </body>
</html>
```

**Computed Styles:**

Property	Value	Source
<b>display</b>	block	
<b>font-size</b>	18px	
1.8rem	#index-title	style.css:11
14px	.index-title	style.css:7
24px	h1	style.css:4
2em	h1	user agent stylesheet
10px	html	style.css:1
<b>font-weight</b>	700	
height	22px	
<b>margin-block-end</b>	12.06px	
<b>margin-block-start</b>	12.06px	
<b>margin-inline-end</b>	0px	
<b>margin-inline-start</b>	0px	
<b>padding-bottom</b>	18px	
<b>padding-left</b>	18px	
<b>padding-right</b>	18px	
<b>padding-top</b>	18px	
1em	.index-title	style.css:7
width	421px	

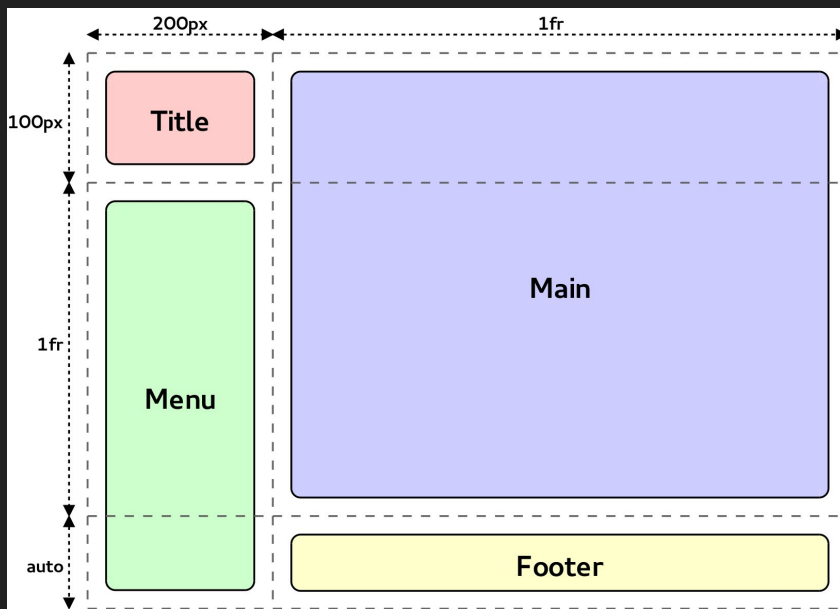
**Rendered Fonts:**

Times — Local file (46 glyphs)

# Display values

Esta propiedad nos permite cambiar la forma en la que este elemento se comporta en el layout, podemos elegir si queremos que sea inline, un bloque y además podemos usarla para definir el layout que van a tener los elementos hijos de este.

La propiedad display en css nos permite darle los siguientes valores: **flex**, **grid**, **block**, **inline**.

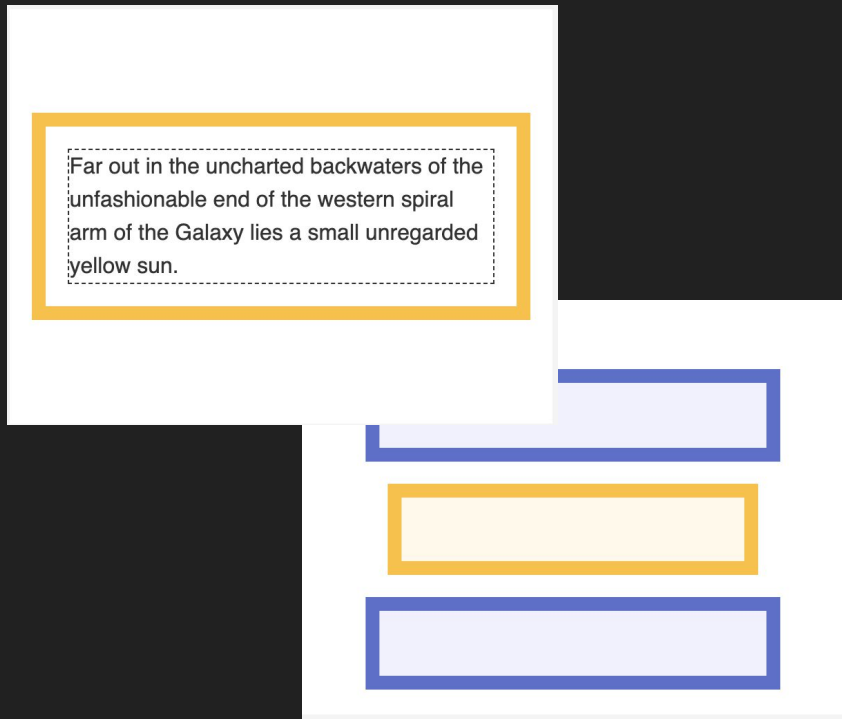




# Padding vs Margin

- **Padding** es el espacio interior, es decir el espacio entre el contenido y los bordes de nuestro elemento.
- **Margin** es el espacio exterior, es decir el espacio que hay rodeando al elemento.

En estas dos propiedades podemos especificar un valor para cada lado usando -top, -right, -bottom, -left o darle el mismo valor a todos los lados usando solamente margin o padding.



# Que es el CSS Object Model?

Es el equivalente al DOM pero para CSS, es decir un conjunto de elementos que nos permiten modificar el CSS desde Javascript.

En Javascript usariamos algo como

`element.style.propertyName =`

`“propertyValue”` para darle un valor que queremos al css, esto es muy útil cuando queremos hacer modificaciones de estilo al interactuar con el usuario.

```
> let links = document.getElementsByClassName('navigation-link');
< undefined
> console.log(links)
  ▶ HTMLCollection(2) [li.navigation-link, li.navigation-link] VM2059:1
< undefined
> links[0].style.border = "3px solid green";
< '3px solid green'
>
```

Usando CSSOM también es posible ver como un arreglo de objetos las stylesheets que le están dando los estilos a nuestra página o elemento seleccionado, en este caso estamos viendo el documento.

```
> document.styleSheets
< ▼ StyleSheetList {0: CSSStyleSheet, length: 1} ⓘ
  ▼ 0: CSSStyleSheet
    ▼ cssRules: CSSRuleList
      ▼ 0: CSSStyleRule
        cssText: "* { border: 1px solid red; margin: 0px; padding: 0px; }"
        parentRule: null
        ▶ parentStyleSheet: CSSStyleSheet {ownerRule: null, cssRules: CSSRul...
        selectorText: "*"
        ▶ style: CSSStyleDeclaration {0: 'border-top-width', 1: 'border-righ...
        ▶ styleMap: StylePropertyMap {size: 25}
          type: 1
        ▶ [[Prototype]]: CSSStyleRule
        ▶ 1: CSSStyleRule {selectorText: 'section', style: CSSStyleDeclaration...
        ▶ 2: CSSStyleRule {selectorText: '.introSection-img', style: CSSStyleD...
          length: 3
        ▶ [[Prototype]]: CSSRuleList
        disabled: false
        href: "http://127.0.0.1:5500/03-Laboratorio-Intro-web/ejercicio/estilo...
        ▶ media: MediaList {length: 0, mediaText: ''}
        ▶ ownerNode: link
        ownerRule: null
        parentStyleSheet: null
        ▶ rules: CSSRuleList {0: CSSStyleRule, 1: CSSStyleRule, 2: CSSStyleRule,...
          title: null
          type: "text/css"
        ▶ [[Prototype]]: CSSStyleSheet
          length: 1
        ▶ [[Prototype]]: StyleSheetList
```