

Agencia de
Aprendizaje
a lo largo
de la vida

Desarrollo Fullstack



Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 03

HTML Revoluciones

- ▶ Multimedia
- ▶ Formularios

Clase 04

Introducción a CSS

- ▶ ¿Qué es CSS?
- ▶ Vinculación
- ▶ Selectores
- ▶ Cascada
- ▶ Herencia
- ▶ Especificidad

Clase 05

CSS Inicial

- ▶ Fuentes y Tipografías
- ▶ Colores
- ▶ Iconos

CSS

Nos vestimos para la ocasión



¿Qué es?

CSS es utilizado para **diseñar y dar estilo** a las páginas web, por ejemplo, alterando la fuente, color, tamaño y espaciado del contenido, dividirlo en múltiples columnas o agregar animaciones y otras características **decorativas**.

Hojas de Estilo en Cascada (del inglés “*Cascading Style Sheets*”), es decir, CSS hace referencia al comportamiento que tiene este lenguaje cuando nuestros estilos entran en conflicto.

Sintaxis de CSS



```
1 selector {  
2     propiedad: "valor";  
3 }
```

selector

Elemento de nuestro HTML que deseamos aplicar estilos.

propiedad

Nombre de la propiedad CSS que queremos modificar.

valor

Valor que le asignamos a esa propiedad.

Sintaxis de CSS



```
1 <p>¡CSS es lo máximo!</p>
```

¡CSS es lo máximo!



```
1 p {  
2     color: #2b2b2b;  
3     font-size: 52px;  
4 }
```

Vinculamos CSS a nuestro proyecto.

Interno

Colocamos nuestro código CSS dentro de la etiqueta **<style></style>** en el head de nuestro archivo HTML.

```
<head>
  <style>

  </style>
</head>
```

Externo

Usamos la etiqueta **<link />** donde el **attr rel** indica la relación con el recurso y el **attr href** la ruta a nuestro archivo css.

```
<link rel="stylesheet" href="style.css">
```

En línea

Se coloca en el **attr style** del elemento/etiqueta HTML a modificar.

```
<p style="color: pink">¡CSS es lo máximo!</p>
```


Selectores CSS

Universal

Selecciona todos los elementos de HTML.

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

Etiqueta

Se utiliza para seleccionar una etiqueta específica.

```
p {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

clase

Selecciona todos los elementos html que contengan ese atributo *class*.

```
.cards {  
  width: 300px  
}
```

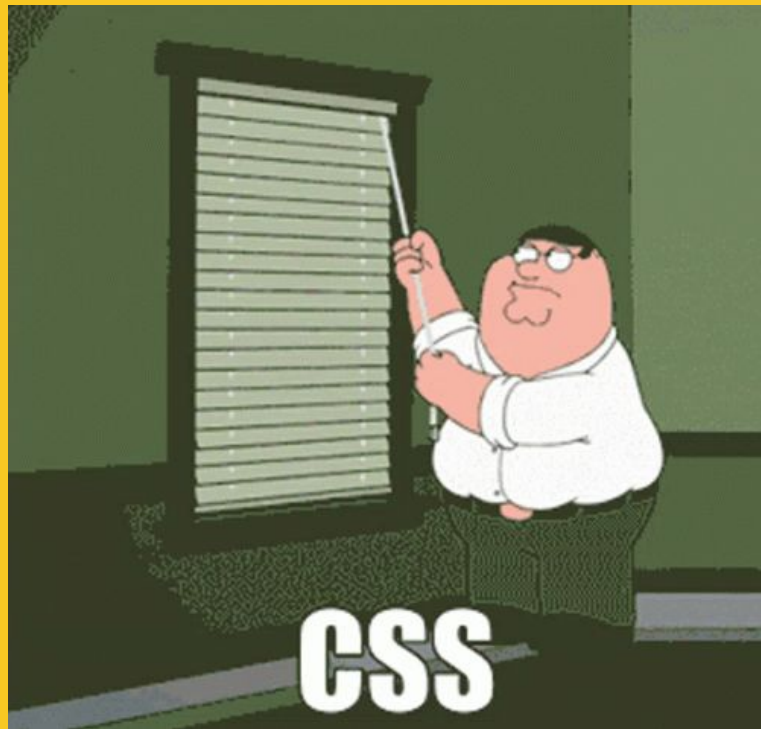
id

Selecciona todos los elementos html que contengan ese atributo *id*.

```
#banner {  
  margin-top: 20px;  
}
```

Práctica en vivo

Creamos un nuevo archivo llamado ***index.css***, lo vinculamos con nuestro HTML y le aplicamos color y tamaño de fuente a una etiqueta **párrafo** con un **lorem ipsum**.



Herencia

Es un **comportamiento** de CSS el cual permite que algunas **propiedades asignadas** a un elemento **padre** sean adquiridas o heredadas por **sus** elementos **hijos**.

Cabe destacar que esto aplica solo a **algunas** propiedades y suelen ser bastante intuitivas, como por ejemplo el **color de fuente** o la **familia tipográfica**.

```
<h1>Nueva sección del sitio</h1>
<article>
  <p>
    Lorem ipsum dolor sit amet consectetur elit.
    Magni laudantium cupiditate odio rem alias
    numquam obcaecati ea, excepturi tempora animi
    provident? Rem eligendi numquam odio velit a,
    consectetur ab blanditiis?
  </p>
</article>
```

```
body {
  font-family: 'Merriweather Sans', sans-serif;
}

article {
  color: ■ crimson;
}
```

Nueva sección del sitio

Lorem ipsum dolor sit amet consectetur elit. Magni laudantium cupiditate odio rem alias numquam obcaecati ea, excepturi tempora animi provident? Rem eligendi numquam odio velit a, consectetur ab blanditiis?

El elemento `<p></p>` *hereda* el **color** de texto de `<article></article>` y la *tipografía* de `<body></body>` mientras que el `<h1></h1>` sólo hereda la **fuerza** y no el **color**.

Es **importante** mencionar que una vez **asignada la misma propiedad heredada al elemento hijo** (aunque tenga distinto valor), éste corta la **herencia de su elemento padre**.

Precedencia

Es muy **importante** en CSS ya que de esta depende la **jerarquía** con la que se aplicarán los estilos a nuestros elementos HTML, por ende, decidirá qué **estilo prevalece** sobre los demás cuando para un elemento apliquen 2 o más estilos.

Junto con el, **también** entra en juego el término de **estilos en cascada** que como vimos anteriormente, está presente en el nombre de CSS.

Orden de precedencia

etiqueta < clase < id < estilo en línea < !important

¿Cuál de todos estos será el que prevalece? 🤔

```
p {  
  font-size: 16px;  
}  
  
.parrafos-main {  
  font-size: 18px;  
}
```

```
#parrafo-principal {  
  font-size: 20px;  
}  
  
.parrafo-diferenciado {  
  font-size: 14px!important;  
}
```

```
<p style="font-size: 30px;">  
  ¡CSS es lo máximo!  
</p>
```

Estilos en Cascada

Además de la **precedencia** existe la jerarquía por **cascada** que es la responsable de decidir qué estilos **prevalecen** cuando dos o más selectores de un mismo elemento tienen la **misma especificidad**.

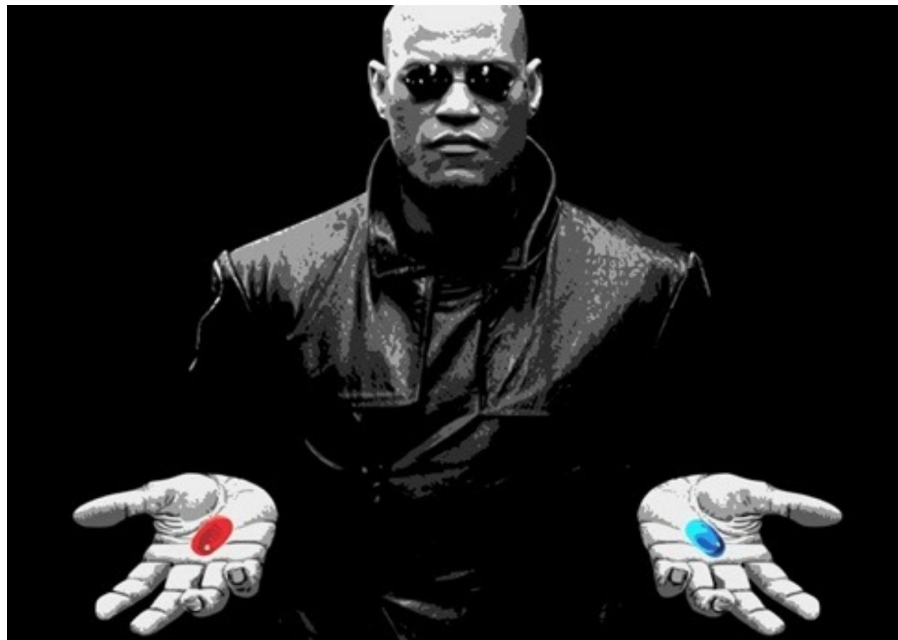
En estos casos, siempre se **tomará el último** valor vigente.

```
p {  
  font-size: 16px;  
  color:  red;  
}  
  
p {  
  color:  blue;  
}
```

Estilos en Cascada

¿Cuál es tu elección?

```
p {  
  font-size: 16px;  
  color: ■ red;  
}  
  
p {  
  color: ■ blue;  
}
```



Si elegiste la Azul...

¡Ingresaste a la Matrix! 🖐️



Veamos algunos ejemplos

Especificidad

La unión hace la fuerza

Como vimos la clase pasada, dependiendo del tipo de selector que utilicemos **conseguiremos** mayor o menor precedencia sobre un elemento, siendo el selector por **ID** el más poderoso de todos.

También existe la posibilidad de **combinar** distintos selectores para poder **reforzar** la selección o llegar a un elemento mucho más **fácil** teniendo en cuenta el árbol de etiquetas anteriores a él.

Cada **selector** posee una **especificidad** y esta nos va a indicar cuán **directa** es la selección del elemento.

```
<element class="menu">  
  ...  
    <element class="submenu">
```

Selector Specificity: (0, 2, 0)

Especificidad según selectores

Si bien es imposible representar todas las combinaciones, cuantos más selectores combinemos más especificidad tendrá nuestra selección.

Etiqueta + Etiqueta:

Seleccionamos todos los `<p></p>` que son hijos de una etiqueta `<article></article>` - E: (0,0,2)

```
<article>
...
  <p>
Selector Specificity: (0, 0, 2)
article p{
  color: ■ crimson;
}
```

Clase + Etiqueta:

Seleccionamos todos los `` que son hijos de `.menu` - E: (0,1,1)

```
<element class="menu">
...
  <li>
Selector Specificity: (0, 1, 1)
.menu li{
  padding: 10px
}
```

ID + Clase:

Seleccionamos todos los `.shirts` hijos de `#sales`.

```
<element id="sales">
...
  <element class="shirts">
Selector Specificity: (1, 1, 0)
#sales .shirts {
  color: □ #2b2b2b;
}
```

Selectores combinados

Esto nos abre una puerta a nuevas combinaciones de selectores.

.menu > li:

Seleccionamos todos los `` que son hijos directos de un elemento con la clase `.menu`

```
<ul class="menu">
  <li>Este si</li>
  <ul>
    <li>Este No</li>
  </ul>
</ul>
```

h2 + p:

Selecciona la etiqueta `<p></p>` que sea hermana directa y adyacente de una etiqueta `<h2></h2>`

```
<h2>Un título</h2>
<p>Este si</p>
<p>Este no</p>
```

h2 ~ p:

Selecciona todas las etiquetas `<p></p>` que sigan luego de una etiqueta `<h2></h2>`

```
<h2>Un título</h2>
<p>Este si</p>
<p>Este si</p>
<p>Este si</p>
```

Selectores combinados

.link.active:

Seleccionamos la etiqueta con la clase **.link** y la clase **.active**

```
<a href="" class="link active">  
  Enlace  
</a>
```

#banner .img:

Seleccionamos la etiqueta con la clase **.img** hija del elemento con id **#banner**

```
<section id="banner">  
  <picture class="img">  
  
  </picture>  
</section>
```

#ofertas.cards:

Seleccionamos la etiqueta con el id **.ofertas** y la clase **.cards**

```
<section id="ofertas" class="cards">  
  
</section>
```


No te olvides de dar el presente

Recordá:

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

Todo en el Aula Virtual.

Gracias