

A graphic on the left side of the slide consists of four stacked, 3D rectangular blocks. From top to bottom, the colors are pink, orange, yellow, and blue. The orange block is slightly offset to the right, creating a sense of depth. The text 'Agencia de Aprendizaje a lo largo de la vida' is written across these blocks in white, with the orange block acting as a large arrow pointing to the right.

Agencia de
Aprendizaje
a lo largo
de la vida

FULL STACK PYTHON

Clase 5

CSS 1

Introducción a CSS



Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 04

HTML 4 - Formularios y subida al servidor

- Formularios.
- Etiquetas semánticas.
- Subida a un hosting gratuito.
- Cómo pensar un proyecto web.

Clase 05

CSS 1 - Introducción a CSS

- Bases del CSS y atributo class.
- CSS externo, interno y en línea.
- Selectores básicos (id, clase, etiqueta, universal).
- Especificidad, Herencia, Cascada y Orden de las reglas en CSS.

Clase 06

CSS 2 - Medidas, colores, fondos, fuentes e íconos

- Unidades de medida.
- Colores CSS.
- Fondos en CSS.
- Fuentes y tipografías.
- Estilos para textos y listas.
- Íconos

CSS

Es un lenguaje de diseño que nos permite darle **estilos** a los componentes de un documento en función de una jerarquía. Se ocupa de la estética, el aspecto.

CSS es una sigla que proviene de **Cascading StyleSheets** (Hojas de Estilo en Cascada, en español). La palabra cascada hace referencia a una propiedad muy importante de CSS, y es la forma en que se comporta cuando entran en conflicto dos o más reglas de estilo.

Cuando diseñamos un sitio web profesional, con un equipo de trabajo, mantener los estilos separados de la estructura y contenido (HTML) facilita la división de tareas entre los desarrolladores.

¿Cómo incorporamos CSS?

CSS en Línea: Dentro del atributo **style=""** incorporamos los estilos que se van a aplicar solo en esa misma etiqueta. Opción no recomendable.

```
<p style="color: ■ white; background-color: ■ violet">Esto es un párrafo</p>
```

Esto es un párrafo

Utilizando el atributo **style** dentro de la etiqueta le proporcionamos estilo al párrafo. Se pueden utilizar a la vez varias parejas de: **propiedad: valor**.

¿Cómo incorporamos CSS?

CSS Interno: Incluimos la etiqueta **<style>** dentro del **<head>** en nuestro documento. Opción menos recomendable:

```
<!-- Estilo interno -->  
<style>  
  h1{  
    color: white;  
    background-color: red;  
  }  
</style>
```



Título H1

En el ejemplo anterior todas las etiquetas **<h1>** tendrán color de fuente blanco y fondo de color rojo.

¿Cómo incorporamos CSS?

CSS Externo: En el **head** del documento HTML tenemos que incluir una referencia al archivo .css dentro del elemento `<link>`. Es la forma más recomendada.

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

La referencia al archivo externo debe incluir **la ruta completa, el nombre del archivo y su extensión** si se encuentra en alguna subcarpeta dentro del proyecto.

En caso de que el archivo de estilos se encuentre en la misma carpeta que el documento HTML, únicamente se debe incluir el **nombre del archivo y su extensión**. Recordemos que es aconsejable mantener estos archivos (CSS y HTML) en carpetas separadas.

Por ejemplo: `href="css/estilos.css"` o `href="estilos.css"`

Estructura

Selector (1): Indica el elemento al cual vamos a aplicar una regla de estilo.

Propiedad (2) y Valor (3): Especifica qué característica voy a afectar de un elemento y qué valor tomará.

Bloque de declaración (4): Indica el estilo que le daremos al selector.

```
(1) h1 { (2) (3)
(4) { color: blue;
font-size: 24px;
}
```

Título con H1

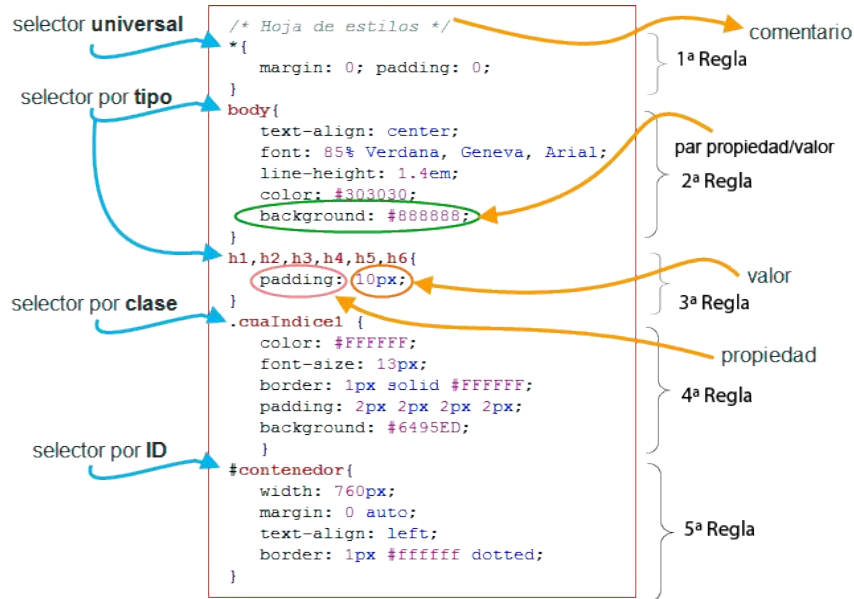
Lorem ipsum dolor sit amet cc

Los estilos se **heredan** de una etiqueta a otra. Si tenemos declarado en el **<body>** unos estilos, en muchos casos, estas declaraciones también afectarán a etiquetas que estén dentro del **body**.

Selectores

Indican el elemento al que se debe aplicar el estilo. La declaración indica "qué hay que hacer" y el selector indica "a quién hay que aplicarlo". Hay cuatro selectores básicos:

- **selector universal:** Selecciona todos los elementos de HTML.
- **selector de etiqueta o tipo:** Se utiliza para seleccionar una etiqueta específica.
- **selector de clase:** Se utiliza agregando el atributo *class* a los elementos que queramos aplicarles estilos.
- **selector de identificador (id):** Similar a *.class* pero solo se aplica a una etiqueta individual.



Selector universal (*)

Se indica con un * (asterisco) y aplica el estilo a todos los elementos contenidos en el documento HTML.

```
* {  
    margin: 10px; /*margin establece el margen para los cuatro lados*/  
    padding: 5px; /*padding establece el espacio de relleno*/  
    background-color: lightgreen; /*cambia el color de relleno*/  
    font-family: Verdana; /*cambia el tipo de letra*/  
}
```

Selector de etiqueta o tipo (<tag>)

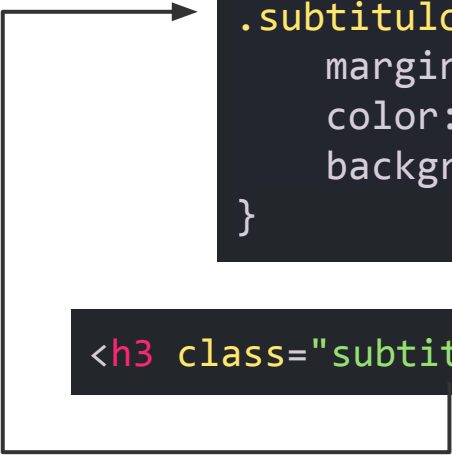
Este selector afecta a una etiqueta específica. Esto nos permite ser más precisos a la hora de aplicar estilos a elementos particulares. Por ejemplo, a **todas** las etiquetas <h1> o <p> del documento.

```
h1 {  
  color: lightblue;  
  background-color: blue;  
}
```

```
p {  
  color: black;  
  font-style: italic;  
  font-size: 130%;  
}
```

Selector de clase (.selector)

Se aplica con un punto (.) seguido del nombre que le asignemos al selector. Dentro del documento HTML se lo referencia dentro de la etiqueta usando el atributo `class`, con la siguiente estructura: `class="nombredelselector"`:




```
.subtitulos {  
  margin-left: 50px;  
  color: yellowgreen;  
  background-color: olivedrab;  
}
```

```
<h3 class="subtitulos">Selectores de clase</h3>
```

Selector de id (#selector)

Se coloca con un numeral (#) en CSS y en el documento HTML se hace referencia al selector con id="nombredelselector", dentro de la etiqueta a la cual se aplica:



```
#texto {  
  color: white;  
  background-color: violet;  
  text-align: center;  
  margin-left: 100px;  
}
```

```
<div id="texto"> Selector de id aplicado a una etiqueta div </div>
```

Span y Div (contenedores de información)

span (abarcar): Es un elemento en línea. Sirve para aplicar estilo al texto o agrupar elementos uno a continuación del otro. Sus etiquetas son **** y **** (ambas obligatorias). Crea una caja que puede contener texto y/u otras etiquetas que se adapten al ancho del contenedor. [+info](#)

div (división): Es un elemento en bloque. Sirve para crear secciones o agrupar contenidos. Sus etiquetas son **<div>** y **</div>** (ambas obligatorias). Crea una caja, que puede contener texto y/u otras etiquetas, que se separa de la caja anterior con un salto a la línea siguiente. [+info](#)

Span y Div (contenedores de información)

```
<span style="color:red">Un texto en span </span>  
<span style="color:blue">Otro texto en span</span>  
<div style="color:darkgreen">Un texto con div</div>  
<div style="background-color:lightblue">Otro texto con div</div>
```

Un texto en span Otro texto en span
Un texto con div
Otro texto con div

Nota: Con *display: inline* el *span* no atiende a la propiedades de alto (height) y ancho (width) ya que se adaptará al tamaño del contenido. Cambiando a *display: inline-block* se permiten estas propiedades, porque se comporta como un *div*. Se puede hacer que un *span* se comporte como un *div* si en CSS agrego *display: block*. [+info](#)

Atributos globales

Estos atributos pueden utilizarse con todos los elementos HTML:

- **style="estilo CSS":** Especifica un estilo CSS conforme al elemento. [+info](#)
- **class="texto":** Especifica uno o más nombres de clases para un elemento (haciendo referencia a una clase en una hoja de estilo). [+info](#)
- **id="texto":** Especifica un id único por cada página. [+info](#)
- **title="texto":** Especifica información extra sobre un elemento (Tooltip Text). [+info](#)
- **hidden:** Evita que el elemento y sus descendientes se muestran en el navegador. Cualquier control de formulario o de script dentro de la sección hidden será ejecutado, aunque no se muestra al usuario. [+info](#)

Atributos globales

- **tabindex="número"**: Especifica la posición del elemento en el orden de tabulación del documento. Se usa para tabular a través de los links de la página (o campos de un formulario). [+info](#)
- **translate="yes|no"**: Indica si el texto del contenido del elemento y los valores del atributo deben ser traducidos o no. La opción por defecto es yes. [+info](#)
- **lang="es"**: Especifica el idioma del contenido del elemento. [+info](#)
- **spellcheck="true|false"**: Especifica si se debe corregir o no la gramática y la ortografía del elemento. [+info](#)
- **draggable="true|false"**: Indica si el elemento es arrastrable; se puede mover haciendo click sin soltar, moviéndolo a una nueva posición en la ventana. [+info](#)

[otros atributos globales](#)

Especificidad

La **especificidad** es el mecanismo mediante el cual los navegadores deciden qué valores de una propiedad CSS son más relevantes para un elemento. Esto influye en la manera en que los estilos son aplicados. La especificidad está basada en las reglas de coincidencia que están compuestas por diferentes tipos de selectores CSS.

La especificidad sólo se aplica cuando el mismo elemento es objetivo de múltiples declaraciones. Según las reglas de CSS, en caso de que un elemento sea objeto de una declaración directa, está siempre tendrá preferencia sobre las reglas heredadas de su ancestro.

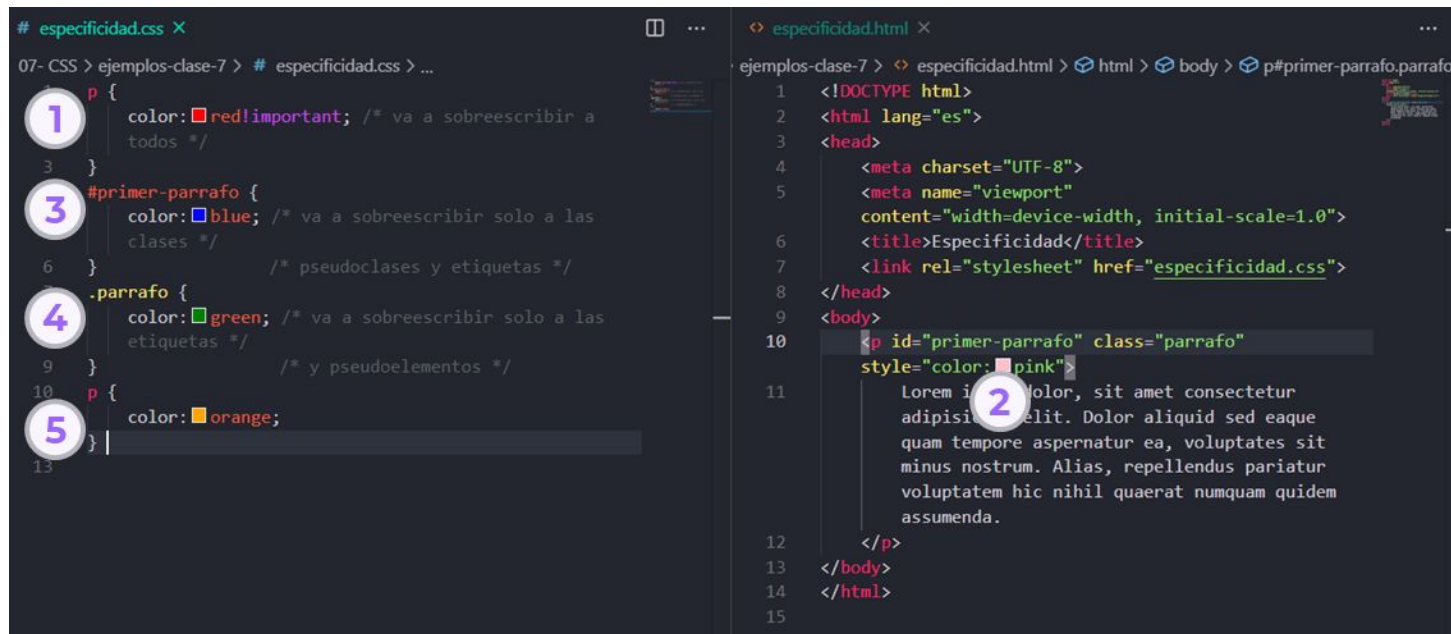
La especificidad hace referencia a la relevancia que tiene un estilo sobre un elemento de la página al cual le están afectando varios estilos de CSS al mismo tiempo. Es decir, hace referencia al grado de importancia de un estilo sobre otro.

Especificidad

La siguiente tabla muestra los niveles de especificidad, desde los más específicos a los más generales:

Estilo	Ejemplo	Nivel (peso)
!important	cualquier-selector { color: #FF0000!important; }	1 , 0, 0, 0, 0
inline	<p style="color:#FF0000;">Lorem Ipsum</p>	0, 1 , 0, 0, 0
ID	#parrafo { color: #FF0000; }	0, 0, 1 , 0, 0
Clases, atributos y pseudoclases	.parrafo { color: #FF0000; }	0, 0, 0, 1 , 0
Etiquetas y pseudoelementos	p { color: #FF0000; }	0, 0, 0, 0, 1

Orden de las reglas en CSS



```
# especificidad.css X
07- CSS > ejemplos-clase-7 > # especificidad.css > ...
1 p {
  color: red !important; /* va a sobrescribir a todos */
3 }
3 #primer-parrafo {
  color: blue; /* va a sobrescribir solo a las clases */
6 } /* pseudoclases y etiquetas */
4 .parrafo {
  color: green; /* va a sobrescribir solo a las etiquetas */
9 } /* y pseudoelementos */
10 p {
5 color: orange;
13 }

especificidad.html X
ejemplos-clase-7 > especificidad.html > html > body > p#primer-parrafo
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport"
  content="width=device-width, initial-scale=1.0">
6   <title>Especificidad</title>
7   <link rel="stylesheet" href="especificidad.css">
8 </head>
9 <body>
10   <p id="primer-parrafo" class="parrafo"
  style="color: pink">
11     Lorem ipsum dolor, sit amet consectetur
  adipiscing elit. Dolor aliquid sed eaque
  quam tempore aspernatur ea, voluptates sit
  minus nostrum. Alias, repellendus pariatur
  voluptatem hic nihil quaerat numquam quidem
  assumenda.
12   </p>
13 </body>
14 </html>
15
```

IMPORTANTE: Cuando varias declaraciones tienen igual especificidad, se aplicará al elemento la última declaración encontrada en el CSS.

Ejemplo explicado:

1. Resultado inicial: Lorem ipsum dolor, sit amet consectetur adipisicing elit. Dolor aliquid s

Si comentamos esta línea: color: red!important;

2. Se aplica el estilo en línea, que es el siguiente en el nivel de jerarquía:

```
<p id="primer-parrafo" class="parrafo" style="color:pink"> Lorem ipsum dolor,
```

3. Si quitamos el estilo en línea, se aplica el estilo del ID (#primer-parrafo):

```
<p id="primer-parrafo" class="parrafo"> Lorem ipsum dolor, sit amet
```

Ejemplo explicado:

4. Si le quitamos el estilo del ID, se aplica el estilo de clase, que es el siguiente en jerarquía:

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Dolor aliquid

5. Y si, finalmente, quitamos el estilo de clase se aplicará el estilo de la etiqueta, que es el último en jerarquía:

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Dolo

Más información:

<https://developer.mozilla.org/es/docs/Web/CSS/Specificity>

https://www.w3schools.com/css/css_specificity.asp

Tarea para el Proyecto:

- Utilizar CSS para aplicar formatos a los párrafos, títulos y demás elementos de las páginas HTML de nuestro proyecto.
- Utilizar CSS en el formulario.

No te olvides de dar el presente

Recordá:

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**
- **Realizar los Ejercicios obligatorios.**

Todo en el Aula Virtual.

Muchas gracias por tu atención.

Nos vemos pronto