

Esta clase va a ser

- grabada

Certificados oficialmente por



CODERHOUSE

¿Dudas sobre el onboarding?

Míralo aquí

Clase 03. DESARROLLO WEB

Incluyendo CSS en nuestro proyecto

Certificados oficialmente por



CODERHOUSE

Glosario

Enlaces: también conocidos como links o anchors, se utilizan para relacionar partes del mismo documento. Por defecto, se visualizan azules y subrayados.

Listas: HTML permite agrupar elementos que tienen más significado de forma conjunta. Aunque cada palabra por separado tiene sentido, de forma conjunta constituyen el menú de navegación de la página, por lo que su significado conjunto es mayor que por separado. Esto se denomina listas.

Formularios: son etiquetas donde el usuario ingresará o seleccionará valores, que serán enviados a un archivo encargado de procesar la información.

Tablas: son un conjunto de celdas organizadas, dentro del cual es posible alojar distintos contenidos. Sirven para representar información tabulada, en filas y columnas.

Etiqueta IFRAME: es un elemento HTML que permite insertar o incrustar un documento HTML (como un video de Youtube) dentro de un documento HTML principal.

Objetivos de la clase

- **Comprender** la sintaxis de CSS
- **Incluir** CSS en nuestro Proyecto
- **Conocer** el uso de medidas, colores, fuentes y fondos en CSS

Temario

02

Primeros pasos con HTML

- ✓ Listas
- ✓ Formularios
- ✓ Enlaces

03

Incluyendo CSS a nuestro proyecto

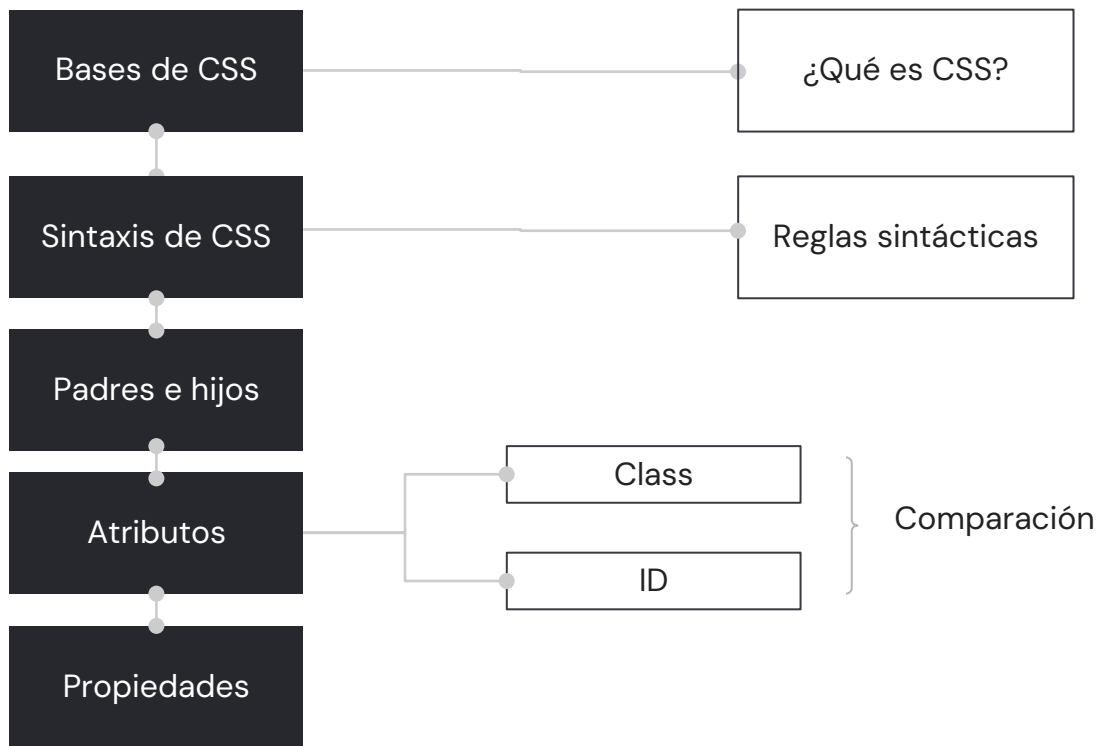
- ✓ [Bases de CSS](#)
- ✓ [Insertar CSS](#)
- ✓ [Primeras propiedades](#)

04

CSS+Box Modeling

- ✓ BEM
- ✓ Box Modeling
- ✓ Display

MAPA DE CONCEPTOS



Bases de CSS



PARA RECORDAR

Premisas

👉 CSS (Cascading Style Sheets) es un lenguaje web para aplicar formato visual (color, tamaño, separación y ubicación) al HTML. Es así que **puedes cambiar por completo el aspecto de cualquier etiqueta HTML.**

Bienvenidos a Diseño Web

Que la fuerza te acompañe!

- Inicio
- Blog
- Contactos

Sobre el profesor

El profesor explica, hace y ustedes practican

Sobre ustedes

Aprenderan a diseñar una página web



SOBRE EL PROFESOR

— ★ —
El profesor explica, hace y ustedes practican

SOBRE USTEDES

— ★ —
Aprenderan a diseñar una página web

Sintaxis de CSS

Sintaxis

```
selector {  
  propiedad1: valor;  
  propiedad2: valor;  
}
```

Ejemplo

```
h1 {  
  color: red;  
}
```

Reglas sintácticas

- ✓ Cada declaración CSS está formada por un par **propiedad: valor;**
- ✓ No se ve afectado por el espacio en blanco. Las propiedades se pueden escribir de corrido o una debajo de la otra.
- ✓ Siempre que la propiedad represente un número, el valor debe indicar en qué unidad se expresa.

Padres e hijos

Padres e hijos

Cuando tienes una etiqueta anidada dentro de otra, lo que haces es aplicar el concepto de padres e hijos.

En este caso, **section** es padre de **article** y, a su vez, **article** es padre del **h2** y del **p**.

```
<section>
  <article>
    <h2> Título </h2>
    <p> Lorem ipsum dolor sit amet,
    consectetur adipiscing elit, sed
    do eiusmod tempor incididunt ut
    labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis
    nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea
    commodo consequat.
  </p>
  <article>
</section>
```

Padres e hijos

Esto habilita a agregar atributos específicos a “hijos”, sin alterar los del “padre”. Un padre puede tener muchos hijos, y todos ellos heredan sus características, pudiendo tener también características particulares.

Selector **PADRE** — `section article {`

Selector **HIJO** — `background-color: #cccccc;`

```
width: 500px;
height: 500px;
}
```


Padres e hijos

En este caso, se observa la forma correcta de declarar cada estilo. Cuando quieres seleccionar una etiqueta, debes incluir la/s etiqueta/s padre/s para que las reglas sean más específicas a la hora de aplicar estilos.

```
<section>
  <article>
    <h2> Título </h2>
    <p> Lorem ipsum dolor sit amet,
    consectetur adipiscing elit, sed
    do eiusmod tempor incididunt ut
    labore et dolore magna aliqua. Ut
    enim ad minim veniam, quis
    nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea
    commodo consequat.
    </p>
  </article>
</section>
```

```
section {
  padding: 50px 30px 20px 60px;
  margin-left: 40px;
}

section article {
  background-color: #cccccc;
  width: 500px;
  height: 500px;
}

section article p {
  line-height: 4;
}
```

Insertar CSS

Insertar CSS en el HTML

Linkear archivos externos - 1

Forma **externa:** dentro de la etiqueta **<head>**, llamas al archivo CSS que necesites (recuerda el uso de rutas relativas y absolutas).

```
<link rel="stylesheet" href="archivo.css" />
```

Insertar CSS en el HTML

Etiqueta <style> - 2

Forma **interna**: se agrega una etiqueta **<style>** dentro de la etiqueta **<head>**. Puede estar en **<body>**, pero sería más desprolijo. Esta práctica no es recomendable.

```
<style>  
    /* comentario de CSS, dentro de esta etiqueta, va el  
    codigo CSS, */  
</style>
```

Insertar CSS en el HTML

Atributo style="" (en línea) – 3

Otra forma **interna**, muy poco recomendable, consiste en usar el atributo style directamente en las etiquetas HTML para “parches” específicos, o pruebas. Se hace difícil mantenerlo, y es considerado una mala práctica.

```
<h1>Un encabezado sin formato</h1>
<h2 style="CODIGO CSS">H2 con formato CSS</h2>

<p>Párrafo sin formatear</p>
<p style="CODIGO CSS">Párrafo formateado</p>
<p>Otro párrafo sin formatear</p>
```

Selectores CSS



PARA RECORDAR

Selectores

👉 Los selectores se utilizan para **darle estilos a elementos html determinados**. Por ejemplo, si quieres que una imagen tenga bordes, y que además sean redondeados.



PARA RECORDAR

Selectores

👉 Hay tres tipos de selectores básicos:

- ✓ IDs
- ✓ Clases
- ✓ Etiquetas

Atributo ID



PARA RECORDAR

ID

☞ Generalmente se usa para **nombrar porciones de código y sectores**, como por ejemplo cuando quieres nombrar distintas secciones.

☞ Es posible ponerle ID a cualquier elemento HTML para darle un "nombre".

☞ No necesitas escribir varias veces el mismo CSS, ni repetir el ID.



ID desde CSS

Desde CSS, puedes **usar los nombres que quieras**, siempre y cuando empiecen con **letras**, y pongas un **"#"** adelante. Lo recomendable es poner un nombre que haga referencias a los estilos que tendrá. Por ejemplo:

```
#productos {  
  /* código CSS */  
}
```



HTML: Atributo ID =""

Para aplicar un ID en el HTML, debes usar el atributo "id", y luego en el valor el nombre del ID (que has especificado en CSS). Por ejemplo:

```
<section id="productos">
```

```
</section>
```

Atributo **Class**



PARA RECORDAR

Class

👉 Las clases se utilizan cuando quieres aplicar el mismo estilo a más de un elemento, y la búsqueda por etiqueta no sirve para lograrlo.

👉 No necesitas escribir varias veces el mismo CSS, ni repetir el ID, simplemente utiliza una clase aplicada a todos los elementos que deban compartir el estilo.



Class desde CSS

Desde CSS, puedes **usar los nombres que quieras**, siempre y cuando empiecen con **letras**, y pongas un "." adelante. Lo recomendable es poner un nombre que haga referencias a los estilos que tendrá. Por ejemplo:

```
.bordesRedondeados {  
  /* código CSS */  
}
```



HTML: Atributo Class=""

En el HTML, para aplicar una clase debes usar el atributo **"class"**, y luego **colocar en el valor el nombre de la clase** (que has especificado en CSS).

```
<img src="" class="bordesRedondeados" />
```




Más de una class

Puedes aplicar **más de una clase** a cada etiqueta separada por un espacio. De esta manera, podrás tener estilos diferenciados para cada clase.

```
<img src="" class="bordesRedondeados imgChica"/>
```

Class vs ID

	¿Se puede reutilizar su nombre en el HTML?	¿Se puede usar varias veces en un atributo en el HTML?	¿Cuándo lo uso?
ID	NO	NO	Nombrar secciones, divisiones de código
CLASS	SI	SI	Especificar diseño aparte del código
Ejemplo ID	id="productos" id="productos2"		<section id="productos">
Ejemplo CLASS	class="bordes" class="bordes"	class="bordes destacado"	<p class="destacado">

Ejemplo HTML



```
<section id= "prod">
  <article class= "rojo">
  </article>
  <article id= "prod">
  </article>
</section>
```



```
<section id= "prod">
  <article class= "rojo">
  </article>
  <article class= "rojo">
  </article>
</section>
```

Tanto **ID** como **Class** pueden ser utilizadas dentro del HTML en diferentes etiquetas. Sin embargo, **los nombres otorgados a las clases se pueden repetir**, mientras que utilizados en **los IDs no**.

Etiquetas



PARA RECORDAR

Selectores de etiqueta

👉 Se usa para **aplicar estilos a todos los elementos de un mismo tipo**, por ejemplo, que todos los párrafos sean azules.

👉 El selector de etiqueta coincide con la etiqueta html del elemento a modificar.

👉 Es un método mucho menos específico para agregar estilos.



Etiquetas desde CSS

Desde CSS, debes **usar los nombres de las etiquetas HTML**. No hace falta agregar ningún símbolo extra para identificar a un selecto de etiqueta:

```
p {  
  /* código CSS */  
}
```



HTML: etiqueta

No hace falta hacer ningún cambio en las etiquetas de los elementos. Por ejemplo:

```
<p>  
  <!--Este párrafo será modificado por el selector p  
desde css -->  
</p>  
<p>  
  <!--Este párrafo será modificado por el selector p  
desde css -->  
</p>
```


Herencia y cascada

Herencia

En general, estas propiedades son intuitivas. Por ejemplo, podrás heredar de un elemento padre el tamaño de letra y color de la misma, *a menos que el elemento hijo tenga otros estilos aplicados*. Puedes ver más al respecto [aquí](#).

```
div {  
  color: red;  
}
```

```
<div>  
  <p>Este párrafo quedará  
  en rojo, por herencia</p>  
</div>
```



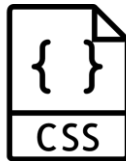
Para pensar: cascada

El navegador lee de arriba hacia abajo (forma de cascada)
¿De qué color crees que se aplicará al párrafo (p) al ver el siguiente código?

```
p {  
    color: red;  
}  
  
p {  
    color: green;  
}
```

Contesta mediante el chat de Zoom

Ejemplo



HTML:

```
<ul>
  <li class="rojo">Item
R1</li>
  <li class="rojo">Item
R2</li>
  <li class="rojo">Item
R3</li>
  <li class="rojo">Item
R4</li>
</ul>
```

CSS:

```
.rojo{ color: red; }
.azul{ color: blue; }
```

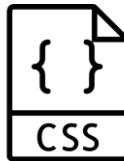
HTML:

```
<ul>
  <li>Item R1</li>
  <li>Item R2</li>
  <li>Item R3</li>
  <li>Item R4</li>
</ul>
<ol>
  <li>Item A1</li>
  <li>Item A2</li>
</ol>
```

CSS:

```
ul li{ color: red; }
ol li{ color: blue; }
```





Precedencia de declaraciones

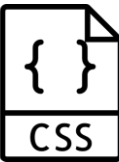
Cuando reglas distintas apuntan al mismo objeto:

- ✓ Si son **propiedades distintas**, se suman (se combinan).
- ✓ Si tienen **alguna propiedad repetida**, solo una queda.

Esto es lo que se denomina *precedencia*.

- ✓ **ID** pisa cualquier otra regla.
- ✓ **Class** sobrescribe las reglas de etiqueta, pero no las de **ID**.
- ✓ **Etiquetas** tienen la menor precedencia.

ID > Class > Etiquetas



Estilos inline

Si utilizas estilos inline, sobrescribirán cualquier estilo de las páginas externas de CSS.

Se podría decir que los estilos inline son los que tienen una mayor especificidad, por lo tanto:



no es recomendable utilizarlos en tu página.

```
<p style= "color: red">Párrafo rojo</p>
```

Importancia de los selectores



Estilo aplicado
a la **Etiqueta.**



Estilo aplicado
a la **Class.**



Estilo
aplicado al **ID.**



Estilo aplicado
Inline.



PARA RECORDAR

!IMPORTANT;

☞ Si tienes 3 reglas CSS, es poco probable que “choquen”, pero en un CSS extenso es más común.

☞ La declaración `!important;` corta la precedencia. Se escribe después del valor de la propiedad CSS que se quiere convertir en la más importante. Se utiliza un `!important;` por cada valor a pisar.

Si necesitas más de 5 `!important;` en todo tu CSS, algo estás haciendo mal 😞

¿Preguntas?

#CoderTip: Ingresa al [siguiente link](#) y revisa el material interactivo que preparamos sobre **Preguntas Frecuentes**, estamos seguros de que allí encontrarás algunas respuestas.



Break

¡10 minutos y volvemos!

Primeras propiedades

Propiedad: color

Mediante esta propiedad, podrás **agregar color a los textos de tu sitio.**

Existen distintos valores, pero nos centraremos en 3:

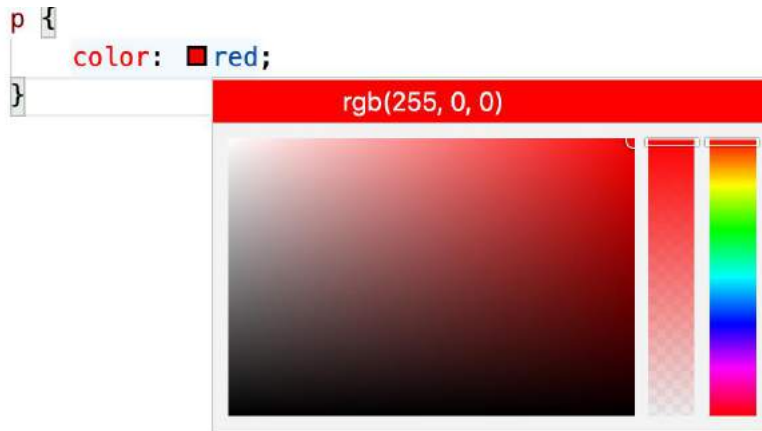
- ✓ Por **nombre del color** (ej: red).
- ✓ **Hexadecimal** (ej: #ffffff).
- ✓ **RGB** (red, green, blue. Por ejemplo: 50, 212, 227). Si agregas un valor más, puedes manejar su opacidad. Cada color permite valores entre 0 y 255.

¿Cómo se eligen los colores?

Desde Google, puedes buscar “[color picker](#)” (alternativa [w3schools](#)).



Desde Visual Studio Code, simplemente te “paras” sobre el color. Por ejemplo, escribe “red” y haz la prueba:



Estilos

Estilos de lista

LIST-STYLE-TYPE

CSS

```
ol {  
  list-style-type: none;  
}  
ul {  
  list-style-type: none;  
}
```

Aplicando esta propiedad y este valor, vamos a poder eliminar las bullets y los números.

Valores posibles: [ver aquí](#)

Estilos de texto

FONT-FAMILY

CSS

```
.impact {  
    font-family: Impact, sans-serif;  
}  
  
.comicSans {  
    font-family: "Comic Sans MS",  
    sans-serif;  
}
```

Valores posibles: <familia o nombre genérico>

Se ve de la siguiente manera:

Familia Impact

Familia Comic

FONT-SIZE

CSS

```
.textoGrande {  
    font-size: 20px;  
}  
  
.textoRelativo {  
    font-size: 200%;  
}
```

Se ve de la siguiente manera:

Texto en 20 px

Texto en 200%

Valores posibles: <medida de longitud> |
<porcentaje>

FONT-WEIGHT

CSS

```
.negrita {  
  font-weight: bold;  
}
```

```
.normal {  
  font-weight: normal;  
}
```

Se ve de la siguiente manera:

Texto normal

Texto en negrita

Valores comunes: normal | bold (luego verán, que puede tener otros valores, en números)

FONT-STYLE

CSS

```
.normal {  
    font-style: normal;  
}
```

```
.italica {  
    font-style: italic;  
}
```

Valores comunes: normal |
italic

Se ve de la siguiente manera:

Texto normal

Texto en itálica

TEXT-ALIGN

CSS

```
.centrar {  
  text-align: center;  
}
```

```
.aLaDerecha {  
  text-align: right;  
}
```

Valores posibles: left | right | center |
justify

Se ve de la siguiente manera:

texto

texto

TEXT-DECORATION

CSS

```
.subrayado {  
    text-decoration: none;  
}  
  
.tachado {  
    text-decoration: line-through;  
}
```

Se ve de la siguiente manera:

Enlace

Parrafo

Valores posibles: none | underline | overline | line-through

Estilos de background

BACKGROUND-COLOR

CSS

```
.fondoFuerte {  
    background-color: yellow;  
}
```

Se ve de la siguiente manera:

Párrafo fondo amarillo

Valores posibles: [color]

BACKGROUND-IMAGE

CSS

```
.catsandstars {  
    background-image:  
url("https://mdn.mozillademos.org/  
files/11991/startransparent.gif"),  
url("https://mdn.mozillademos.org/  
files/7693/catfront.png");  
}
```

Valores posibles: url | none

Se ve de la siguiente manera:



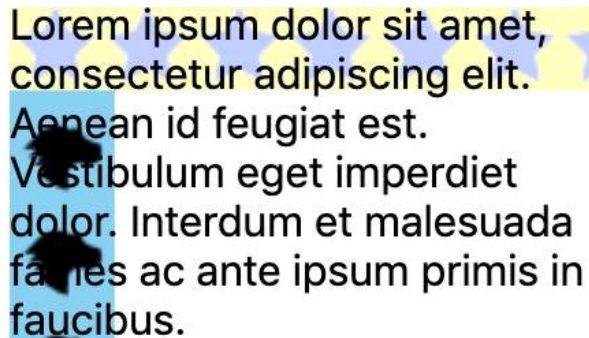
Párrafo sin fondo.

BACKGROUND-REPEAT

CSS

```
.ejemplo {  
  background-image:  
url(https://mdn.mozillademos.org/files/12  
005/starsolid.gif),  
url(https://developer.cdn.mozilla.net/med  
ia/redesign/img/favicon32.png);  
  background-repeat: repeat-x,  
                    repeat-y;  
}
```

Se ve de la siguiente manera:



Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean id feugiat est.
Vestibulum eget imperdiet
dolor. Interdum et malesuada
fames ac ante ipsum primis in
faucibus.

Valores posibles: repeat | repeat-x | repeat-y | no-repeat | space | round ([ver ejemplos](#))

BACKGROUND-POSITION

CSS

```
.ejemplo {  
    background-image:  
url("https://mdn.mozillademos.org/fi  
les/12005/starsolid.gif");  
    background-repeat: no-repeat;  
    background-position: right  
center;  
}
```

Valores posibles: posicionX posicionY ([ver ejemplos](#))

Se ve de la siguiente manera:

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean id feugiat est.
Vestibulum eget imperdiet
dolor. Interdum et malesuada
fames ac ante ipsum primis in
faucibus.



BACKGROUND-SIZE

CSS

```
.ejemplo {  
    background-image:  
url("https://mdn.mozillademos.org/fi  
les/12005/starsolid.gif");  
    background-repeat: no-repeat;  
    background-size: cover;  
}
```

Se ve de la siguiente manera:

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean id feugiat est.
Vestibulum eget imperdiet
dolor. Interdum et malesuada
fames ac ante ipsum primis in
faucibus.

Valores posibles: [ancho] | [alto] | cover | contain ([ver ejemplos](#))

Unidades de medidas

Unidades de medida

Hay una amplia variedad de absolutas y relativas, pero nos centraremos en:

Absolutas

- ✓ Px (pixels): es la unidad que usan las pantallas.

Relativas

- ✓ Rem: relativa a la configuración de tamaño de la raíz (etiqueta html).
- ✓ Porcentaje: tomando en cuenta que 16px es 100%.
- ✓ Viewport: se mide en relación a la parte visible de la ventana del navegador. Se utiliza para layouts responsivos (más adelante).

Unidades de medida

Ahora veamos qué medida es más conveniente para los textos.

```
html { /* etiqueta raíz */  
  font-size: 62.5%;  
}  
  
p {  
  font-size: 2rem; /* 20px */  
}
```

**Texto
simulando
20px**

62.5%, hace que en vez de que 16px sea el valor a tomar en cuenta para calcular las unidades relativas, se use 10px.

Reset



PARA RECORDAR

Reseteo CSS

🔗 Los reset CSS contienen, en su código fuente, definiciones para propiedades problemáticas que los diseñadores necesitan unificar desde un principio.

Por ejemplo, la mayoría de navegadores establece un margen por defecto entre el contenido de la página web y su propia ventana, cuyo valor varía de un navegador a otro.

🔗 Para subsanar esa diferencia, los diseñadores y las diseñadoras de sitios webs suelen declarar la siguiente línea al comienzo de sus hojas de estilo:

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

Reset CSS

Esa única línea indica, con el selector universal de CSS representado por un asterisco, que todos los elementos contenidos en el HTML a los que se aplique, carecerán de márgenes. De esa manera, el diseñador o la diseñadora se verán obligados a declarar luego los márgenes necesarios en el diseño de su página web, en cada uno de los lugares donde se requiera, sin tener que dejar ese aspecto a decisión de ningún navegador, y minimizando las diferencias visuales entre los mismos.

Atención: los reset CSS pueden contener esa y otras muchas líneas de código que, en su conjunto, servirán al diseñador/a web para unificar su visualización entre navegadores.



Selectores

Utiliza los selectores para dar estilos.

Duración: **15 minutos**



ACTIVIDAD EN CLASE

Selectores

Al archivo HTML creado previamente, agrega un div con un párrafo lleno de texto, y asígnale la clase "actividad1". Además, crea un archivo CSS con una clase llamada "actividad1"; y asígnale la propiedad color con valor naranja (orange).

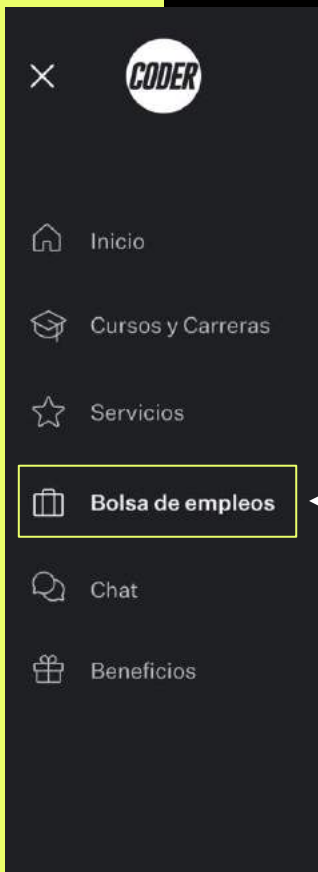
Profundiza y conoce atajos en
nuestra [Cheat Sheet](#)





Primera pre-entrega

En la clase que viene se presentará la primera parte del Proyecto Final, que **nuclea temas vistos entre las clases 1 y 4**.
Recuerda que tendrás 7 días para subirla en la plataforma.



Nuevo

¡Lanzamos la Bolsa de Empleos!

Un espacio para seguir **potenciando tu carrera** y que tengas más **oportunidades de inserción laboral**.

Podrás encontrar la **Bolsa de Empleos** en el menú izquierdo de la plataforma.

Te invitamos a conocerla y ¡postularte a tu futuro trabajo!

Conócela

¿Preguntas?

Resumen de la clase hoy

- ✓ Bases y sintaxis de CSS
- ✓ Padres e hijos
- ✓ Etiquetas
- ✓ Herencia y cascada
- ✓ Primeras propiedades
- ✓ Estilos
- ✓ Unidades de medida



¿Quieres saber más?
**Te dejamos material
ampliado de la clase**



MATERIAL AMPLIADO

Recursos multimedia

Título

- ✓ [Patrones sutiles](#) | Toptal
- ✓ [Recursos de Dominio Público](#) | Internet Archive
- ✓ [Jardin Zen CSS](#) | CSS Zen Garden

Opina y valora
esta clase

Muchas gracias.

#DemocratizandoLaEducación