

PROGRAMACIÓN WEB  
teoría/práctica



GUÍA BÁSICA DE CSS



# Guía Básica de CSS

**Complejidad de esta currícula:** baja

**Tiempo aproximado de desarrollo:** de (1-3) encuentros.

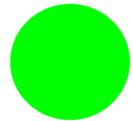
Qué necesitás saber para esta currícula?



Introducción a la programación



Introducción a la creación de sitios web



Guía básica de HTML



# Guía de CSS

- ▶ Con esta currícula aprenderemos a como darle estilos a nuestras páginas html.
- ▶ Vamos a seguir trabajando con el proyecto que creamos en la "**Guía Básica de HTML**"
- ▶ Así como trabajamos con archivos HTML, los archivos de CSS se crean y editan usando un editor (Atom, Sublime o el que nos guste más)
- ▶ El lenguaje CSS opera "sobre" el lenguaje HTML...sin etiquetas HTML, CSS no podría existir
- ▶ Para ver los cambios que vayamos haciendo con CSS solo debemos ejecutar en el navegador el archivo HTML.
- ▶ Al finalizar la currícula, vamos a tener nuestro sitio web con estilos...



# CSS (Cascading Style Sheet)

- ▶ CSS significa "**Hojas de estilo en cascada**", y es un lenguaje que no es de programación...pero al igual que HTML, es un lenguaje!
- ▶ La potencia de CSS está en que puede tomar los elementos de HTML (como párrafos, títulos, el cuerpo entero, etc) darles estilo... y también animación :)
- ▶ CSS permite crear "reglas" que especifican estos estilos.
- ▶ La clave para entender como trabaja CSS es imaginarnos que hay una "**caja invisible**" alrededor de cada elemento de HTML.
- ▶ Hay tres formas de escribir CSS:
  - ▶ 1. Dentro del mismo archivo HTML
  - ▶ 2. En un archivo CSS aparte
  - ▶ 3. Dentro de la misma etiqueta HTML





# 1. CSS dentro del mismo archivo HTML

- ▶ Esta es una de las formas menos usadas, pero en general se utiliza cuando los estilos son muy pocos.
- ▶ Dentro de un archivo HTML, se "incrusta" el código CSS.
- ▶ Esto es posible gracias a la etiqueta **<style>**
- ▶ Veamos un pedacito de código HTML con CSS:

```
<head>
  <style type="text/css">
    body {
    }
    h1 {
    }
  </style>
</head>
```

-----> Etiqueta HTML para escribir CSS

background-color: linen;
color: maroon;
margin-left: 40px;

CSS

> Fin del código CSS

-----



## 2. CSS externo a un archivo HTML

- ▶ Crear un archivo con los estilos es la práctica más común cuando se tienen grandes volúmenes de reglas de CSS, es decir que nuestras páginas web tendrán muchos estilos!
- ▶ Desde los archivos HTML "llamaremos" a los archivos CSS
- ▶ Ejemplo: Nuestro archivo index.html contiene una llamada al archivo "estilo.css"

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="estilo.css">
  </head>
  <body>
    <h1> Hola Mundo </h1>
  </body>
</html>
```



**Llamada al archivo CSS**



## 3. CSS dentro de una etiqueta de HTML

- ▶ Hoy en día es el método menos frecuente y menos recomendado porque presta a confusión, además de que cuando el código crece, se hace difícil de mantener. Tampoco esto es bueno para cuando se deben reutilizar reglas de CSS o si hay que cambiar un valor se debería cambiar sentencia por sentencia.
- ▶ Ejemplo, si queremos que un título tenga color y un margen de 30 píxeles:

```
<h1 style="color:blue;margin-left:30px;">Hola Mundo</h1>
```

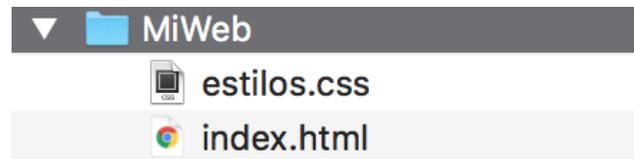


**Nosotras usaremos archivos externos de CSS para hacer nuestras páginas web!**



# 1er Paso: Incluimos el archivo CSS

- ▶ 1. Nosotras vamos a elegir el método 2, que es el de incluir un archivo externo de CSS.
- ▶ 2. Dentro de `<head></head>` vamos a agregar la etiqueta:  
`<link rel="stylesheet" href="estilos.css">`
- ▶ Esta etiqueta lo que hace es "llamar" al archivo CSS que vamos a crear. Lo llamamos según su nombre y su ubicación... En este caso el archivo `estilos.css` estará dentro de la carpeta "MiWeb", junto a el archivo `index.html`



- ▶ Que pasaría si el archivo `estilos.css` se encuentra dentro de una carpeta "css" y esta a su vez dentro de la carpeta "MiWeb"? Entonces deberíamos cambiar la ruta de datos:

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



# 1er Paso: Incluimos el archivo CSS

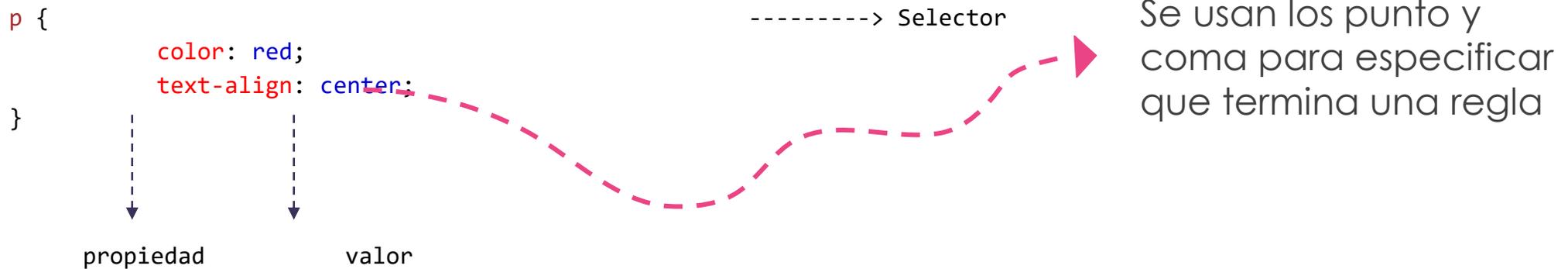
- ▶ 3. Creamos el archivo estilos.css desde el editor

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Chicas Programadoras</title>
5
6     <!-- con esta etiqueta llamamos al archivo css -->
7     <link rel="stylesheet" href="estilos.css">
8
9   </head>
```

- ▶ El archivo "estilos.css" ya está creado y enlazado a nuestro HTML, es decir que, si empezamos a escribir CSS ya podríamos ver los cambios! :)

# Reglas de CSS

- ▶ Empecemos a ver como darle vida a nuestras páginas web.
- ▶ Una regla de CSS se compone de:



- ▶ Selector: sirven para especificar a que etiqueta de HTML vamos a darle un valor (un estilo). La misma regla puede ser aplicada a múltiples selectores
- ▶ Propiedades: son las características del elemento que queremos cambiar



# Reglas de CSS

- ▶ Con la regla anterior estamos diciendo que **"todos los elementos <p> de HTML (todos los párrafos) deben tener letra de color rojo y deben estar alineados hacia el centro"**.
- ▶ Si quisiéramos que además de los párrafos, los títulos también tengan el color rojo y estén centrados, entonces escribiríamos:

```
p, h1 {  
    color: red;  
    text-align: center;  
}
```



# Que es un ID ?

- ▶ Pensemos en nuestro número de documento (DNI), es un número único que nos identifica... nadie más tiene un DNI igual al nuestro.
- ▶ Para los elementos de HTML ocurre lo mismo. Podemos asociarles un único identificador (id)...
- ▶ Para que sirve? Para diferenciarlo de los demás elementos!
- ▶ Cuando queremos diferenciarlo de otros elementos? Cuando queremos darle un estilo diferente a ese elemento.
- ▶ Si por ejemplo en nuestra página web tenemos 3 párrafos y solo uno queremos que tenga letras en color azul, entonces deberíamos identificarlo desde HTML con un nombre único y luego mediante CSS darle la propiedad del color azul.



# Que es un ID ?

- ▶ Llevemos a la práctica el concepto:

## HTML:

### CSS:

```
<p id="primerparrafo"> Este es el primer párrafo </p>
<p> Este es el segundo párrafo </p>
  color: blue;
<p> Este es el tercer párrafo </p>
```

```
#primerparrafo {
}
```

## En Pantalla:

Este es el primer párrafo  
Este es el segundo párrafo  
Este es el tercer párrafo

**La forma de "llamar" al identificador a través de CSS es anteponiendo #**



# Que es una Clase?

- ▶ Cuantos medios de transporte conocemos? Podríamos empezar con mencionar una bicicleta, una moto, un cuatriciclo, un auto... Todos tienen la misma función: movilizarnos.
- ▶ Si bien son diferentes entre sí, todos pertenecen a una misma clase o categoría:  
**VEHÍCULOS**
- ▶ Entonces una clase es un grupo de elementos que tienen características similares
- ▶ Para que sirve? Para identificar varios elementos que queramos que tengan estilos similares
- ▶ Volviendo al ejemplo de los párrafos... que tal si ahora en vez de poner un párrafo en letras azules, ponemos dos?



# Que es una Clase?

- ▶ Llevemos a la práctica el concepto:

## HTML:

```
<p class="parrafoazul"> Este es el primer párrafo </p>  
<p> Este es el segundo párrafo </p>  
<p class="parrafoazul"> Este es el tercer párrafo </p>
```

## CSS:

```
.parrafoazul {  
  
}
```

## En Pantalla:

Este es el primer párrafo  
Este es el segundo párrafo  
Este es el tercer párrafo

**La forma de "llamar" a una clase a través de CSS es anteponiendo un punto "."**

- ▶ Tanto los identificadores como las clases...son SELECTORES



# 2do Paso: IDs y Clases

- ▶ 1. Vamos al archivo index.html y pensamos... que elementos queremos que tengan el mismo estilo? En este caso vamos a plantear que queremos que todos los títulos (h1 a h6) tengan una tipografía distinta al resto del texto.
- ▶ 2. Para esto, vamos a buscar todos los elementos título del archivo html y agregamos una **clase** que represente lo que mencionamos...Por ejemplo:

```
10 <body>
11   <h1 class="titulo">Hola Mundo</h1>
12   <p>Esta es mi primer página web. Aquí voy a nombrar cuales son mis series preferidas</p>
13
14   <h2 class="titulo">Series Cómicas</h2>
```

- ▶ De esta forma, identificamos los elementos mediante HTML y luego mediante CSS (en otro paso más adelante) les daremos la propiedad que queremos.



## 2do Paso: IDs y Clases

- ▶ 3. Lo mismo hacemos con algún elemento único que queramos que tenga un estilo diferente. Usamos el ID, y en este caso seleccionamos la imagen:

```
111 
```

- ▶ Ahora la imagen de nuestro sitio se encuentra identificada de manera única a través del identificador "gotgirls"



# Colores

▶ Hay varias formas de escribir colores:

- ▶ Colores en hexadecimal: se expresan en "código", son 6 dígitos que representan las cantidades de rojo, verde y azul en un color. Ejemplo: #333042, #ee3e80
- ▶ RGB: Red Green Blue (rojo, verde y azul), expresa la cantidad de cada color. Ejemplo: rgb(100,100,90).
- ▶ Nombres: Hay 147 nombres de colores predefinidos para usar. Ejemplo: red, blue, DarkCyan, etc..



▶ Ejemplos:

```
#p3 {  
  background-color: #0000ff;  
  color: rgb(100,100,90);  
}
```

```
#p3 {
```

```
background-color: red;  
}
```

```
#p3 {
```

```
background-
```



# Tipografía

- ▶ Un sitio web siempre debe tener un tipo de letra acorde, hay muchas para elegir...
- ▶ Hay algunas que podemos usar por defecto (que en general todos los sistemas operativos la tienen instalada) y otras que es necesario instalar o referenciarlas desde el sitio web donde estén (por ejemplo Google ofrece muchas tipografías).
- ▶ Por ejemplo si queremos que un párrafo tenga tipografía "Times New Roman":

```
p {  
    font-family: "Times New Roman", Times, serif;  
}
```

**Times y serif son la "familia de letras", es decir que son del mismo tipo ya que si "Times New Roman" no se encuentra disponible, una letra similar de la familia serif se va a mostrar.**

- ▶ Utilizamos la propiedad **"font-family"** de CSS para esto.



# Tipografía

- ▶ Podemos encontrar muchas tipografías diferentes en: [//fonts.google.com](https://fonts.google.com)
- ▶ Para poder utilizarla debemos seleccionar la fuente que queremos, apretar el signo "+" y luego click en "Family Selected" y copiar la etiqueta de HTML que aparece allí.
- ▶ Esta etiqueta HTML debemos pegarla en nuestro archivo HTML dentro de <head> y es la "llamada" a esa tipografía, es decir, que la vamos a cargar para poder utilizarla en nuestra página.
- ▶ Luego copiamos la font-family de esa tipografía y la agregamos a nuestro archivo de CSS



# Tamaño de la tipografía

- ▶ Ahora que ya sabemos como elegir tipografías también tenemos que saber que podemos cambiar su tamaño.
- ▶ Para esto se utiliza la propiedad: **font-size**

```
p {  
    font-family: "Times New Roman", Times, serif;  
    font-size: 15px;  
}
```

- ▶ El tamaño no solamente puede estar en píxels, por ahora daremos también en porcentajes:

```
p {  
    font-family: "Times New Roman", Times, serif;  
    font-size: 200%;  
}
```



## 2do Paso: Reglas de CSS

- ▶ 1. Podemos empezar a escribir reglas de CSS que apliquen a los elementos HTML que ya tenemos en nuestra página. Por ejemplo:

```

MiWeb
├── estilos.css
└── index.html

index.html
estilos.css

1  /*definimos que tipografía nos gustaría para toda la página*/
2  body {
3      font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
4  }
5
6  /*si queremos que los títulos tengan otra tipografía...*/
7  .titulo {
8      font-family: Georgia, Times, "Times New Roman", serif;
9  }
10
11 /*podemos elegir el tamaño de letra para los párrafos*/
12 p {
13     font-size: 14px;
14 }
15
16 #gotgirls {
17     background-color: #672F91;
18 }
```



## 2do Paso: Reglas de CSS

- ▶ 2. Agreguemos tipografía de Google... para que la página luzca más moderna! Seleccionamos la fuente, copiamos la etiqueta HTML (en este caso seleccionamos la fuente Griffy) y la pegamos en index.html
- ▶ 3. En nuestro archivo de CSS, los títulos h1 a h6 tendrán esta tipografía, escribimos la regla de CSS para dichos elementos.

### HTML

```
<!-- tipografia de google -->  
<link href="https://fonts.googleapis.com/css?family=Griffy" rel="stylesheet">
```

### CSS

```
/*si queremos que los títulos tengan otra tipografía...*/  
h1, h2, h3, h4 {  
    font-family: 'Griffy', cursive;  
}
```



# Negritas y Cursivas

- ▶ Si bien es posible poner en negritas o cursivas un texto mediante las etiquetas de HTML: `<b></b>` y `<i></i>` ... nosotras lo haremos mediante CSS:

- ▶ Cursiva (itálica, italic):

```
p {  
    font-style: italic;  
}
```

- ▶ Negrita (bold):

```
P {  
    font-weight: bold;  
}
```

- ▶ En estos casos estaríamos declarando que todos los párrafos de nuestra página estén en negrita y cursiva. Si quisiéramos aplicarlo solo a algunos elementos podríamos utilizar clases y/o ID.



# Alinear Texto

- ▶ Muchas veces debemos controlar el texto, tal vez quisiéramos que el texto este más a la izquierda, más a la derecha, justificado o más al centro. Con la propiedad *text-align* esto es posible!

Al centro:

Justificado:

```
p {  
    text-align: center;  
    text-align: justify;  
}
```

A la izquierda:

A la derecha:

```
p {  
    text-align: left;  
    text-align: right;  
}
```



# Pseudo Clases

- ▶ Una pseudo clase es "similar a una clase"... como las que definimos previamente cuando queríamos que varios elementos HTML compartan las mismas propiedades.
- ▶ Se utilizan para cambiar la apariencia de ciertos elementos cuando el usuario interactúa con estos elementos. Por ejemplo, podemos hacer que un botón cambie de color cuando posiciona el mouse sobre el mismo, o que el fondo cambie de color, etc...
- ▶ `:hover`, se utiliza para indicar "cuando el mouse pasa por" entonces si tuviéramos:

```
a:hover {  
    background-color: yellow;  
}
```

- ▶ `:focus`, se utiliza para indicar que se quiere "focalizar en el elemento" entonces si tuviéramos:

```
input:focus {  
    background-color: yellow;  
}
```



# 3er Paso: Más Reglas de CSS

▶ 1. Agregaremos algunas palabras en negrita y otras en cursiva a nuestra página. En este caso, asignamos un ID al primer párrafo y aplicamos ambas propiedades a dicho párrafo...

```
20 #intro {  
21     font-weight: bold;  
22     font-style: italic;  
23 }
```

▶ 2. Alineamos el texto! Podemos ubicar los elementos a nuestro antojo, en este caso vamos a centrar el título h1 y el primer párrafo.

```
25 h1, #intro {  
26     text-align: center;  
27 }
```

▶ 3. Vamos a utilizar la pseudo clase :hover para que cuando nos posicionemos con el mouse sobre los botones cambie de color el texto dentro del mismo

```
29 button:hover {  
30     color: blue;  
31 }
```

▶ 4. Podemos agregar colores, fondos a párrafos, títulos, etc...



# Selectores de CSS

- ▶ Hasta ahora aprendimos que podemos seleccionar un elemento HTML de las siguientes formas:

- ▶ Llamándolo por el nombre del elemento:

```
p {  
  text-align: left;  
}
```

- ▶ Llamándolo por el nombre de su clase:

```
.parrafoazul {  
  text-align:  
  left;  
}
```

- ▶ Llamándolo por su ID:

```
#parrafo {  
  text-align: left;  
}
```

- ▶ Pero hay muchas más formas de hacerlo... tengamos siempre a mano la siguiente lista con los más frecuentes:



Selector	Significado	Ejemplo
Selector Universal	Se aplica sobre TODOS los elementos HTML	<pre>* {   font-family: Arial, Verdana, sans-serif; } }</pre>
Selector de Tipo	Se aplica solo sobre los elementos que indique	<pre>p {   font-size: 14px; }</pre>
Selector de Clase	Se aplica solo sobre los elementos que tengan dicha clase	<pre>.parrafoazul {   font-size: 14px; }</pre>
Selector ID	Se aplica solo sobre EL elemento que tenga dicho ID	<pre>#parrafo {   font-size: 14px; }</pre>
Selector Hijo	Se aplica solo sobre los hijos directos de otro elemento	<pre>li&gt;a {   Font-size: 14px; }</pre>
Selector Descendente	Se aplica sobre todos los elementos dentro de otro elemento aunque no sean hijos directos	<pre>p a {   font-size: 14px; }</pre>



# 4to Paso: Selectores de CSS

- ▶ En el paso 2, usamos la clase "titulo" para diferenciar los elementos h1 a h4 de los demás elementos. Una opción era esta pero otra opción era utilizar directamente el "selector de tipo", de modo que todo elemento h1 a h4 de nuestra página web adquiriera esa propiedad, sin necesidad de agregar clases a través de HTML.

- ▶ Forma alternativa:

```
7 h1, h2, h3, h4 {  
8     font-family: Georgia, Times, "Times New Roman", serif;  
9 }
```

- ▶ 1. La idea ahora es ver a partir de nuestro código que podemos mejorar, reemplazando selectores, agregando/sacando clases y ids.

# Cajas

- ▶ Como mencionamos en la slide 3:

"La clave para entender como trabaja CSS es imaginarnos que hay una "caja invisible" alrededor de cada elemento de HTML"

- ▶ Estas cajas pueden ser controladas a través de CSS, de forma individual o grupal.
- ▶ Para representar estas cajas podemos usar la etiqueta `<div>`

**Hola Mundo**

---

**Series Cómicas**

1. Friends [Sitio Web](#)
2. Fuller House [Sitio Web](#)
3. The Big Band Theory [Sitio Web](#)

**Series Suspenso**

1. Games of Thrones [Sitio Web](#)
2. Stranger Things [Sitio Web](#)
3. The Walking Dead [Sitio Web](#)

**Personajes que más me gustan**

- Eleven
- Arya Stark
- Jon Snow
- Carl Grimes

**Personajes de Games of Thrones**

Nombre	Casa
Arya	Stark
Sansa	Stark
Daenerys Targaryen	
Cersei	Lannister
Margaery Tyrell	



# Bordes, Margen y Relleno

- ▶ Cada caja tiene 3 propiedades para controlar su apariencia en cuanto a espacio dentro de la página: Border (borde), Margin (margen) y Padding (relleno).

- ▶ Border: es la línea que rodea a la caja.

**Hola Mundo**

- ▶ Margin: se toma desde el exterior de la caja y es la distancia entre la caja y otro elemento.

**Hola Mundo**

Margin

**Hola Mundo**

- ▶ Padding: es la distancia entre el contenido de la caja (en este caso Hola Mundo) con el borde de la caja

Padding

**Hola Mundo**



# Bordes, Margen y Relleno

- ▶ Margen, al igual que padding puede especificarse para cada lado de la caja

```
div {  
  
    margin-top: 1px;  
    margin-right: 2px;  
    margin-bottom: 4px;  
    margin-left: 3px;  
}
```

- ▶ Padding, puede especificarse para que lado de la caja necesitamos exista un "relleno":

```
div {  
  
    padding-top: 10px;  
    padding-right: 15px;  
    padding-bottom: 20px;  
    padding-left: 5px;  
}
```



# Bordes, Margen y Relleno

- ▶ Border, tiene propiedades muy utilizadas como:

```
div {  
    border-width: 3px;  
    border-style: dotted;  
    border-color: #0088dd;  
}
```

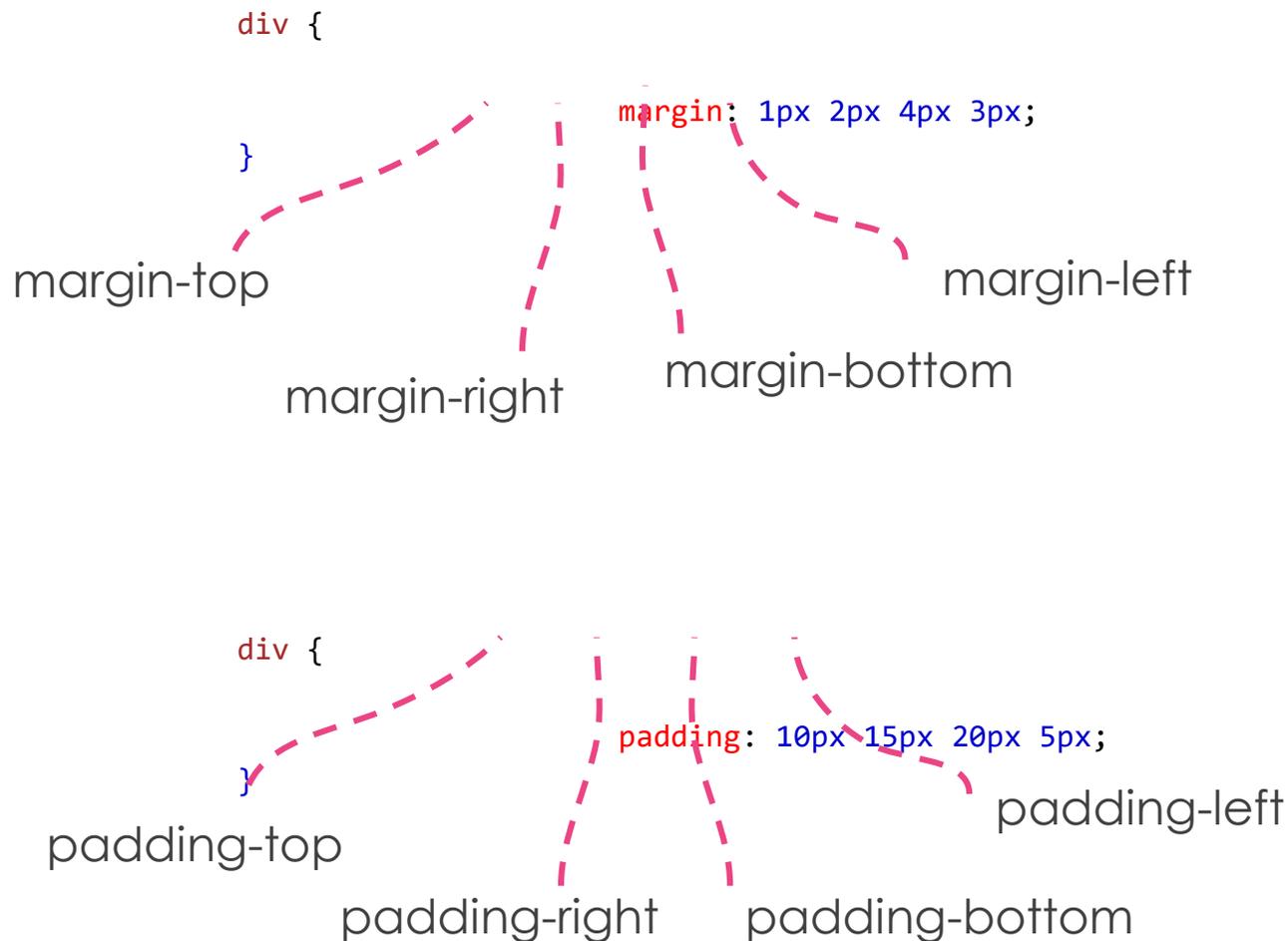
- ▶ Es posible unificar estas propiedades para escribir menos:

```
div {  
    border: 3px dotted #0088dd;  
}
```

The diagram illustrates the unification of border properties. A red dashed line connects the 'border-width' property from the first code block to the '3px' value in the unified 'border' property. Another red dashed line connects the 'border-style' property from the first code block to the 'dotted' value in the unified 'border' property. A third red dashed line connects the 'border-color' property from the first code block to the '#0088dd' value in the unified 'border' property. The unified 'border' property is shown in the second code block.

# Bordes, Margen y Relleno

- ▶ También es válido para margin y padding:



Para acordarnos de esto  
pensemos en el recorrido  
de las agujas de un reloj





# 5to Paso: Ubicación de elementos

- ▶ 1. Agreguemos border, margin y padding a los elementos que consideremos necesario según como queremos que luzcan. Recordar el uso de <div> en el archivo index.html

```
35 #series {  
36     border: 5px dotted #942796;  
37     margin: 40px;  
38     padding: 20px;  
39 }
```



# Desafío

- ▶ Investigá sobre el uso de CSS y sobre otras propiedades que pueden ser aplicadas, por ejemplo a las listas, al formulario y a cada elemento del formulario.
- ▶ El proyecto desarrollado en esta currícula podrás obtenerlo como referencia.

**En caso de que necesites, podés descargar el archivo de este proyecto en:**

<https://drive.google.com/open?id=0BydYIZzB2UyxZG1QV0hYa0RMTkU>

Y sino **pedí a tu mentora que te lo facilite**



¿Preguntas?



MUCHAS GRACIAS