

# Introducción a CSS

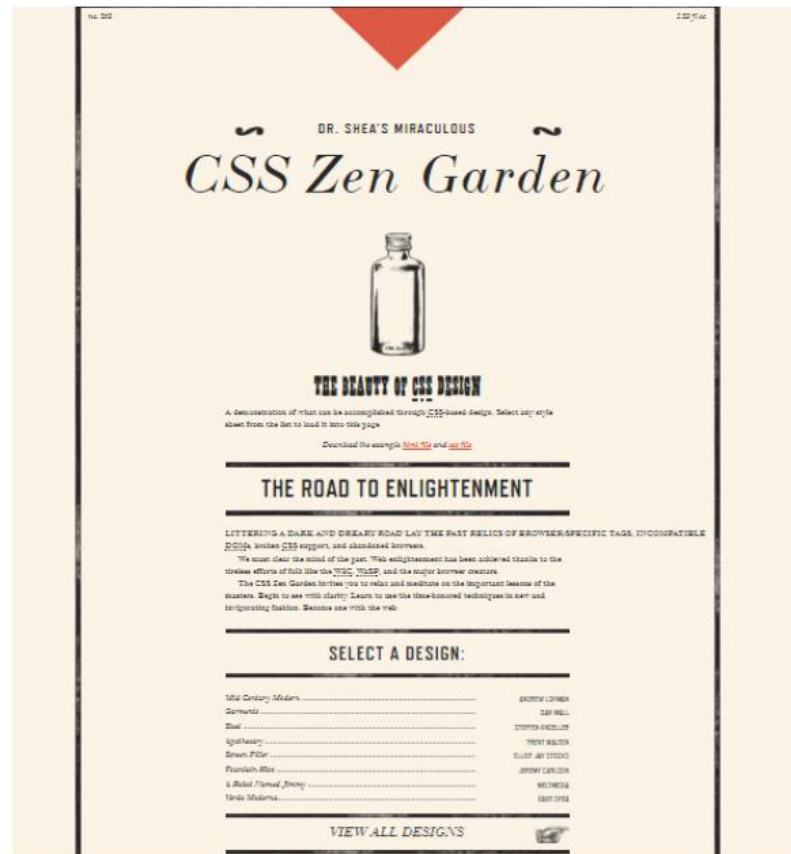
# Contenidos de la Clase

- Sintaxis y Reglas
- Jerarquías
- Selectores
- Propiedades Básicas
- Fuentes
- Background
- Box Shadow
- Degradados
- Bordes

# Recapitulamos...

# ¿Qué es CSS?

CSS (Cascading Style Sheets) es un lenguaje web para aplicar formato visual al HTML. Con CSS se puede cambiar por completo el aspecto de cualquier etiqueta HTML, sin modificarla.



# Vincular con HTML

- ❑ Forma EXTERNA: dentro de <head> llamamos al archivo CSS (recordar rutas relativas y absolutas).

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

- ❑ Forma INTERNA 1: recomendable utilizarla dentro de <head> (se puede usar en <body> pero es más desprolijo).

```
<style>
  /*Código CSS*/
</style>
```

- ❑ Forma INTERNA 2: no es recomendable ya que es difícil de mantener. Se utiliza para parches o pruebas.

```
<section style="background-color: aqua;">
```

# iComencemos!

# Sintaxis de CSS

```
/* Ejemplo de estilos por etiquetas */
body {
    background-color: lightblue;
    border: 5px dotted black;
}

/* Ejemplo de estilos por clases (recordar llamarlo en el archivo html) */
.textoCentrado {
    text-align: center;
}
```

# Jerarquías en CSS

Cuando hay una etiqueta “dentro” de otra, aplicamos el concepto de padres e hijos. En este caso, `<section>` es padre de `<article>` y `<article>` es padre de `<h1>` y `<p>`.

```
<body>
  <section>
    <article>
      <h1>Primer título</h1>
      <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. </p>
    </article>
  </section>
</body>
```

Esto nos habilita a agregar atributos específicos a etiquetas “hijas” sin alterar los del “padre”.

Selector **HIJO**

↓

Selector **PADRE** → `section article {`  
`background-color: #cccccc;`  
`width: 500px;`  
`height: 500px;`  
`}`



# Clases en CSS

Se utilizan para darles estilos a partes específicas del código.

El nombre de las clases es libre, pero debería relacionado con su contenido.

Además hay que tener en cuenta que deben comenzar con un “.”

```
.textoCentrado{  
|   text-align: center;  
}
```

Una vez creada, en el archivo HTML se debe usar el atributo “class” para llamarla, y en el valor se debe colocar su nombre. Todos los elementos HTML admiten este atributo.

```
<h1 class="textoCentrado">Primer título</h1>
```

Si queremos aplicar más de una clase, quedaría así:

```
<h1 class="textoCentrado textoRojo">Primer título</h1>
```

# Atributo ID

- ✓ Se usa para nombrar porciones de código y sectores.
- ✓ Es posible ponerle un **ID** a cualquier elemento HTML para darle un “nombre”.
- ✓ Al querer aplicar un estilo a distintos lugares de la página, y no poder hacerlo con las etiquetas, podemos utilizar este ID.
- ✓ Los nombres siguen las mismas reglas que los de las clases, pero en el CSS comienzan con un “#”.

```
#nombreid {
|   text-align: right;
}
```

En el archivo HTML utilizamos el atributo “id” y en su valor irá el nombre.

```
<section id="nombreid">
```

# Class vs ID

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

# Selección de HTML + CSS

Por etiquetas

Por clase (".")

Por ID ("#")

```
section article p {  
  line-height: 4;  
}  
  
.textoCentrado{  
  text-align: center;  
}  
  
#nombreid {  
  text-align: right;  
}
```

## Nombres de Clases e ID

No se pueden crear nombres separados por espacios, por lo que se recomienda utilizar "camelCase".

camelCase nos permite leer de forma más simple palabras compuestas.

```
.textoCentrado{
```

# Herencia y Cascada

- ❑ Se heredan de un elemento padre algunos estilos, y el elemento hijo podría tener otros estilos aplicados, por su parte, por lo que tendremos que saber cuál será el estilo dominante en ese caso.

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

```
<div>
|   <p>
|   |   Este párrafo quedará rojo por herencia.
|   </p>
</div>
```

- ❑ Para saber cuáles son los estilos dominantes utilizaremos el concepto de cascada. El navegador leerá de arriba hacia abajo, de ahí viene el nombre de cascada.

```
div {
  color: red;
}

div {
  color: green;
}
```

# Precedencia

Cuando hay reglas distintas que apuntan hacia el mismo objeto:

- Si son propiedades distintas, se suman (se combinan).
- Si tienen alguna propiedad repetida, se reemplazan.

Con respecto a la precedencia para reemplazarse:

- ✓ ID pisa cualquier otra regla.
- ✓ **Class** sobrescribe las reglas de etiqueta, pero no las de ID.
- ✓ Las **etiquetas** tienen la menor precedencia.
- ✓ Los estilos inline (atributo style) sobrescribirán cualquier estilo externo de CSS. Si bien tienen mayor especificidad, no es recomendable utilizarlos.

## !IMPORTANT

Si tenemos 3 reglas CSS, es poco probable que se pisen, pero si tenemos un CSS extenso puede pasar. Para estos casos se utiliza la declaración “**!important**”, que corta la precedencia. Se escribe después del valor de la propiedad CSS “más importante”. Se utiliza uno por cada valor.

OJO!! No deberías necesitar más de 5 “!important”.

# Color

Desde Google, podemos buscar “[color picker](#)”.

Desde VSC, podemos posicionar el mouse en el cuadradito de color y elegir.

Tenemos varios tipos de valores, aunque nos centraremos en los siguientes 3:

- Por nombre del color (Ej. red)
- Hexadecimal (Ej. #FFFFFF)
- RGB (Ej. 50, 212, 227). Si agregamos un valor más, determinaremos la opacidad. Cada color admitirá hasta 256.



# Reseteo

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

Por ejemplo, para subsanar la diferencia entre los márgenes por defecto del sitio y la ventana, los diseñadores de sitios webs suelen declarar lo siguiente al comienzo de sus hojas de estilo:

“\*”: todos los elementos contenidos en el HTML.

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

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.



# Listas y Texto

- ❑ Aplicando la propiedad `list-style-type` eliminamos las viñetas de las listas.

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

- ❑ Con `font-style` decidiremos entre normal e itálica.

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

- ❑ Con `font-weight` decidiremos entre normal e itálica.

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

# Texto

- ❑ Aplicando la propiedad **text-align** definimos hacia dónde se alinea el texto. Se puede elegir entre “center, right, left o justify”.

```
.textoCentrado{  
|   text-align: center;  
}
```

- ❑ Con **line-height** definimos el interlineado. Podemos elegir entre “none, números, longitud o porcentaje”.

```
.interlineado {  
|   line-height: 1.6;  
}
```

- ❑ Con **text-decoration** agregamos características al texto. Podemos elegir entre...

```
❏ dashed  
❏ dotted  
❏ double  
❏ line-through
```

```
❏ none  
❏ overline  
❏ solid  
❏ underline  
❏ wavy
```

# Fuentes

- ❑ Aplicando la propiedad `font-size` definimos el tamaño de la letra.

```
.textoGrande {
|   font-size: 20px;
}
```

```
.textoRelativo {
|   font-size: 200%;
}
```

- ❑ Con `font-family` definiremos la familia de fuentes de la página.

```
'Courier New', Courier, monospace
'Franklin Gothic Medium', 'Arial Narrow', Arial, ...
'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet ...
'Lucida Sans', 'Lucida Sans Regular', 'Lucida Gra...
'Segoe UI', Tahoma, Geneva, Verdana, sans-serif
'Times New Roman', Times, serif
'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Gr...
Arial, Helvetica, sans-serif
Cambria, Cochin, Georgia, Times, 'Times New Roman...
Georgia, 'Times New Roman', Times, serif
Impact, Haettenschweiler, 'Arial Narrow Bold', sa...
Verdana, Geneva, Tahoma, sans-serif
```

[Link](#) para ver cómo funciona cada fuente en distintos sistemas operativos.

# Tipografías



Las tipografías Serif llevan detalles adicionales en los bordes de las letras. Estos detalles transmiten clasicismo, formalidad, precisión, tradición, delicadeza y/o refinamiento.

Las tipografías Sans Serif no tienen estos detalles. Se utilizan en entornos digitales porque estos detalles son difíciles de plasmar en píxeles. Transmiten fuerza, modernidad, elegancia y actualidad.

También podemos utilizar las fuentes alojadas en “Google Fonts”.

```
<head>
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
  <title>Document</title>
  <link rel="stylesheet" href="pruebas.css"/>
</head>
```

# Fondos

- ❑ background-color: color de fondo.
- ❑ background-image: agregar una imagen de fondo.
- ❑ background-repeat: repetición de las imágenes de fondo.
- ❑ background-position: posiciono las imágenes de fondo.
- ❑ background-size: tamaño del fondo.

```
.ejemplo {
    background-image: url(./images/foto.webp);
    background-repeat: no-repeat;
    background-size: cover;
}
```

# Box Shadow

La propiedad box-shadow se utiliza para aplicar sombras a los elementos. Ejemplo de su sintaxis:



```
.box {  
  box-shadow: 2px 2px 4px □ rgba(0, 0, 0, 0.4);  
}
```

En este ejemplo, la sombra tendrá un desplazamiento horizontal de 2px, un desplazamiento vertical de 2px, un desenfoque de 4px y un color de sombra semitransparente.



<https://developer.mozilla.org/es/docs/Web/CSS/box-shadow>

# Degradados

Degradados lineales: permiten mezclar dos o más colores en una dirección específica.

```
.gradientLineal {  
  background: linear-gradient(to right,  #ff0000,  #00ff00);  
}
```

Degradados radiales: mezclan colores en forma de círculo o elipse desde un punto central.

```
.gradientRadial {  
  background: radial-gradient(circle,  #ff0000,  #00ff00);  
}
```

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

# Unidades de Medida

Tenemos unidades absolutas y relativas.

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 utilizan para layouts responsivos.

Las más convenientes para los textos son las unidades relativas, ya que al cambiar el tamaño de la ventana, cambiarán los tamaños de las letras.

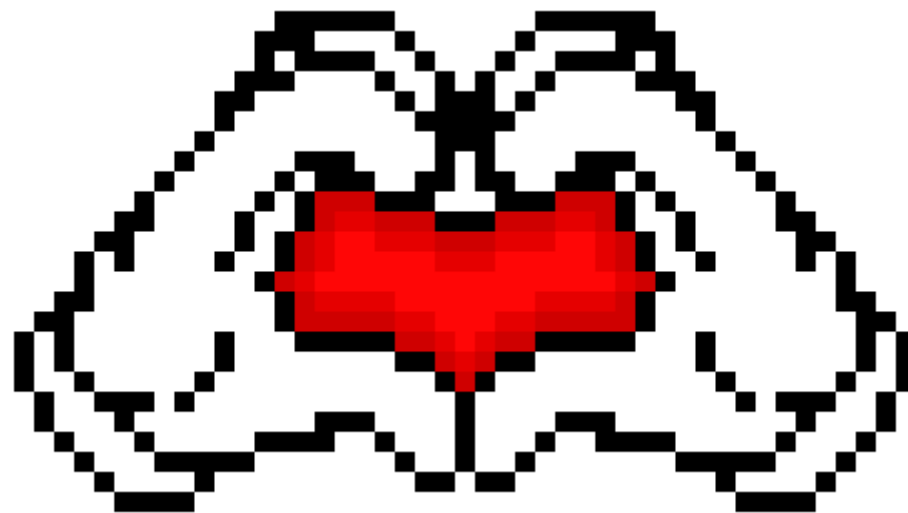


# ¡Practiquemos!

# ¿Preguntas?



# ¡Nos vemos la próxima clase!



# BA MULTIPLICA 2.0

jóvenes  jóvenes



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

