



Programación 3 Frontend - 2025

Actividad Práctica: CSS Parte 1

1. Selectores, Propiedades y valores

1.1 Reiniciando estilos del navegador

Objetivo: Comprender qué estilos aplica el navegador por defecto y cómo resetearlos con selectores de elementos.

Consigna:

- 1) Creá un archivo HTML con los elementos: h1, p, ul, li, a, button, input, textarea.
- 2) Sin aplicar ningún estilo, observá cómo se muestran en el navegador.
- 3) Ahora creá un archivo CSS y escribí reglas para aplicar un "reset":

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

- 4) Luego, usá selectores de elemento para darles estilos iniciales "neutros":

```
body {  
  font-family: sans-serif;  
  line-height: 1.5;  
}  
  
ul {  
  list-style: none;  
}  
  
a {  
  text-decoration: none;  
  color: inherit;  
}
```

Reflexión: ¿Qué estilos desaparecieron al hacer esto? ¿Por qué es útil empezar de esta forma?

1.2 Creando clases utilitarias

Objetivo: Diseñar clases que se puedan reutilizar sin acoplarse al contenido.

Consigna:

- 1) Creá un archivo HTML con varios div que tengan distintos textos.
- 2) Creá un archivo CSS con clases utilitarias como las siguientes:

```
.text-center {  
  text-align: center;  
}  
  
.bg-light {  
  background-color: #f5f5f5;  
}  
  
.p-1 {  
  padding: 1rem;  
}  
  
.mb-2 {  
  margin-bottom: 0.5rem;  
}
```

- 3) Usá esas clases en tu HTML para dar formato a los div.
- 4) Intentá combinar varias clases en un solo div.

Reflexión: ¿Qué ventajas tiene usar clases utilitarias? ¿Qué desventajas podés encontrar?

Adicional: Intenta crear tus propias clases utilitarias

1.3 Componé tu primer "componente" con HTML y CSS

Objetivo: Aplicar la idea de composición de estilos.

Consigna:

- 1) Diseñá una "card" de producto o perfil que tenga:
 - Imagen
 - Título
 - Descripción
 - Botón
- 2) Creá las siguientes clases básicas y reutilizables:

```
.card {  
  border: 1px solid #ccc;  
  border-radius: 8px;  
  overflow: hidden;  
  max-width: 300px;  
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
}  
  
.card-title {  
  font-size: 1.2rem;  
  font-weight: bold;  
  margin-bottom: 0.5rem;  
}  
  
.card-description {  
  font-size: 1rem;  
  color: #555;  
}  
  
.btn {  
  background-color: #3498db;  
  color: white;  
  padding: 0.5rem 1rem;  
  border: none;  
  cursor: pointer;  
}
```

- 3) Armá el HTML aplicando esas clases. Luego, creá otra card usando las mismas clases pero cambiando el contenido (por ejemplo, sin descripción).

Adicional: Cambia la apariencia general de todas las cards sin tocar el HTML.

1.4 Aprovechando la Herencia

Hace que todo el sitio web tenga una fuente determinada. El tamaño de texto base de todos los elementos debe ser de 18px. La página debe de tener un color de fondo. Por último, todo el texto debe estar centrado y con el color #066687.

2. Box Model

De ahora en más, en todos los ejercicios aplicar la siguientes reglas al inicio de la hoja de estilos:

```
* {  
  box-sizing: border-box;  
}  
  
body {  
  margin: 0;  
  padding: 0;  
}
```

2.1 "Caja simple"

Crear una caja de tamaño fijo y aplicar *margin*, *border* y *padding*.

```
<!-- HTML -->  
<div class="caja">Hola mundo</div>
```

```
/* CSS */  
.caja {  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 5px solid black;  
  margin: 30px;  
  background-color: lightblue;  
}
```

Consigna:

- Modificar cada una de las propiedades (*padding*, *border*, *margin*, *width*, *height*) y observar cómo se ve afectado el tamaño total de la caja.
- Usar las herramientas del navegador (inspector) para ver el modelo de caja.

2.2 Cajas anidadas

```
<div class="outer">
  <div class="inner">Contenido</div>
</div>
```

```
.outer {
  width: 300px;
  padding: 30px;
  background-color: #eee;
}

.inner {
  width: 100%;
  padding: 20px;
  border: 3px dashed red;
  background-color: white;
}
```

Consigna:

- Probar cambiar el *width* de *.inner* a valores fijos (ej. 200px) y ver qué ocurre.
- Cambiar los valores de padding y border en ambos elementos.
- Agregar *margin* al *.inner* y observar cómo afecta en relación con el *.outer*.

2.3 Banner con fondo y texto

Crear un “Hero Banner”. Este componente debe tener:

- Una imagen de fondo.
- El elemento con el fondo debe de tener un *padding* bastante alto, por ejemplo 50px.
- Por encima del fondo debe de haber un texto centrado horizontalmente.

2.4 Caja centrada

Crear una caja con fondo azul:

- Debe tener un ancho y alto con unidades absolutas.
- Debe poseer un borde en todos sus lados.
- En su interior, debe poseer un texto que diga “UTN FRRa”
- Debe estar centrado horizontalmente en la página web.

3. Display block, inline e inline-block

3.1 Comparando displays

Crear tres cajas iguales pero con distintos valores de *display*, y observar su comportamiento.

```
<p class="block">Block</p>
<span class="inline">Inline</span>
<span class="inline-block">Inline-Block</span>
```

```
.block, .inline, .inline-block {
  background-color: lightcoral;
  border: 2px solid black;
  padding: 10px;
  margin: 5px;
  width: 200px;
  height: 50px;
}

.block {
  display: block;
}

.inline {
  display: inline;
}

.inline-block {
  display: inline-block;
}
```

- ¿Cuál ocupa toda la línea?
- ¿En cuáles se respeta el *width* y *height*?
- ¿Cuál se puede alinear junto a otro elemento en la misma línea?

3.2 Botones en fila

Crear tres botones utilizando el elemento `<div>`, que estén en la misma línea:

- Cada botón debe tener un color diferente.
- Cada color se debe aplicar con una clase css diferente, los demás estilos se aplican a una clase llamada "btn".
- Cada botón debe tener una separación derecha de 8px.

Ejemplo:



3.3 Menú de navegación horizontal

Crear un menú simple de navegación con el siguiente marcado HTML:

```
<div class="menu">
  <a href="#">Inicio</a>
  <a href="#">Servicios</a>
  <a href="#">Nosotros</a>
  <a href="#">Contacto</a>
</div>
```

- Cada enlace del menú debe estar distribuido de forma horizontal.
- Asignarle un color de fondo y de texto.
- Asignarle un relleno.
- Cuando el usuario posicione el cursor sobre el elemento que genera el link, su color de fondo debe cambiar.

3.4 Ocultar elemento

Crear un `<div>` que cumpla con lo siguiente:

- Asignar un *padding* en todos sus lados de *2rem*.
- Debe tener en su interior un texto que diga "UTN FRRa".
- Debe tener un color de fondo azul.
- Su color de texto debe ser blanco.
- El texto debe estar centrado horizontalmente.
- Cuando el usuario posicione el cursor sobre el elemento, este debe desaparecer.