

## Devolución del trabajo práctico N°3

Alumnos: Josue Terrazas, Dayana Terrazas, Ivan Tarifa, Santiago Farfán.

Calificación: Aprobado.

Revisiones:

### Punto 1:

En el primer punto colocaron la etiqueta “header” fuera del body. No pueden existir etiquetas de contenido fuera del body. Las etiquetas contenedores (header, main, footer, section, div, etc) deben ir siempre en el body. Veo que este error fue corregido en puntos posteriores (punto 4). A continuación voy a marcar una serie de puntos para revisar:

```
* {  
  margin: 0%;  
  padding: 0cap; 🙌 /* 1 */  
}  
header {  
  width: 1500px; 🙌 /* 2 */  
  height: 200px;  
  background-image: linear-gradient(red, orange); 🙌 /* 3 */  
}  
h1 {  
  text-align: center;  
  line-height: 200px; 🙌 /* 4 */  
}
```

1. Padding: 0cap no es válido. “cap” no es una unidad reconocida de medida. Deben ser pixeles, rem, porcentajes, vh o vw. Cualquier unidad de tamaño reconocida es válida.
2. No podemos ponerle el ancho a las cosas manualmente. Si o si tenemos que hacer que se ajuste al ancho de la página o del contenedor padre. Esto se hace por ejemplo estableciendo width: 100% o width: 100vw (viewport width o ancho de la página). El problema de poner 1500px es la siguiente pregunta: Qué pasa si viene una persona con una pantalla de 1600px? No va a ocupar el ancho total.
3. Si bien esto es correcto, es importante agregar los grados que el gradiente va a inclinarse, por ejemplo “45deg” (a 45 grados) o palabras como “to top right” (hacia arriba a la derecha).
4. Esta línea no sirve ningún propósito en el código ya que no necesitamos hacer la altura del texto más alta.

### Punto 2:

Si bien el punto 2 cumple con los requisitos, hay una importante mejora que hacer al código ya que es repetitivo. Actualmente el código se ve así:

```
#uno {
    background-color: rgb(183, 241, 222);
}
#dos {
    background-color: aqua;
}
#tres {
    background-color: rgb(183, 241, 222);
}
#cuatro {
    background-color: aqua;
}
#cinco {
    background-color: rgb(183, 241, 222);
}
```

Asigna las mismas propiedades a través de ids muchas veces, o sea, cae en repetición. Esto no puede pasar teniendo la posibilidad de escribir clases que apliquen dichas propiedades. Veamos la siguiente solución:

```
.fila-impar {
    background-color: rgb(183, 241, 222);
}
.fila-par {
    background-color: aqua;
}
```

De esta manera evitamos tener que repetir el código muchas veces.

#### Punto 4:

Veamos correcciones individuales en el punto 4:

```
header {
    width: 30%;
    background-color: darkcyan;
    height: 750px; 🙌 /* 1 */
}
.container {
    height: 100%;
    width: 1600x; 🙌 /* 2 */
}
```

```
.menu {
  padding: 16px;
  font-size: 30px;
  display: block; 🙌 /* 3 */
  text-align: center;
}

.menu a { 🙌 /* 4 */
  line-height: 80px;
  display: block;
  color: black;
  border-left: 6px solid white;
}
```

1. Nuevamente, y si viene una persona con una pantalla de 1080px? 100vh para la altura soluciona este problema.
2. Error de tipeo, corregir.
3. Esta línea de código no sirve de ningún propósito para el menú que se está desarrollando. Verán más adelante que poner display: block (ocupar el largo de la página) en este caso es perjudicial si queremos tener algún contenido a la derecha del menú.
4. line-height y display: block no son correctos en este caso. En lugar de utilizar line-height es mejor recurrir a margin para dar espacio entre los links.

### Punto 5:

En el C, se repite mucho código y se colocan distancias en pixeles (La pregunta prevalece, que pasa si viene un usuario con X cantidad de pixeles?). Si bien es un poco ambigua, dejo la solución a este problema:

```
div {
  background-color: #49e;
  height: 100px;
  width: 100px;
  border: solid #000;
  position: relative;
  left: 50%;
  right: 50%;
  top: 50%;
  bottom: 50%;
}
```

El D incurre en el mismo problema de cantidades en pixeles. Recordemos que es más fácil pensar esto como en el siguiente ejemplo: Cuánto quiero alejarme del lado derecho? un 30%? bien, right: 30%. Por medio de esta lógica, llegamos a la siguiente solución para el D:

```
#caja-1 {  
  position: absolute;  
  top: 0;  
  left: 50%;  
}  
#caja-2 {  
  position: absolute;  
  right: 0;  
  top: 50%;  
}  
#caja-3 {  
  position: absolute;  
  left: 50%;  
  bottom: 0;  
}  
#caja-4 {  
  position: absolute;  
  left: 0;  
  top: 50%;  
}
```

Si bien no queda perfectamente centrado, la idea es manejarse por medio de porcentajes o cantidades relativas al ancho o alto de una pantalla.