



Fundamentos de desarrollo web

CSS

Introducción

En la actualidad, el tema de maquetar web sin tocar CSS, es inconcebible. Esto se debe a que para la maquetación, el lenguaje de hojas de estilo en cascada es fundamental, por las características propias del lenguaje y las características que le brinda a una estructura de etiquetas ya definida, es decir, hecha, por ejemplo, en HTML.

La tecnología CSS fue diseñada para distinguir la separación del contenido de los elementos descritos en el archivo HTML. Es la tecnología encargada de la presentación visual, pues es utilizada para afinar la presentación final del documento; se encarga de hacer un sitio web “responsive”, y en general, trabajar con estándares la estructura y presentación de la página.



Explicación

Introducción



Antes de que existiese CSS, se disponía exclusivamente del HTML, las carencias a la hora de posicionar elementos en la página eran pobres o inexistentes, esto es porque cuando fue creado, no se esperaba que la web se convirtiera en un medio masivo de consulta y divulgación de información, adonde los profesionales aportarían grandes cantidades de creatividad y una necesidad de diseño atractivo. El HTML primitivo permitía tratar bloques de texto en párrafos, acompañado de enlaces, listas, imágenes, tablas, pero no

algo visualmente atractivo como a lo que hoy en día estamos acostumbrados.

Con HTML, los programadores utilizaron el único recurso que existía en la época para posicionar elementos en la página, las tablas. Las bondades de las tablas permiten presentar información mediante filas y columnas, pero las tablas son un recurso que se queda corto al momento de maquetar una web entera.

Luego de la aparición de CSS, además de permitir los programadores aprender un lenguaje nuevo, permitió explorar la posibilidad de que diseñadores puedan participar en los proyectos y trabajar de la mano con programadores. De esa manera nació una nueva profesión: el desarrollador web.

Hoy en día, HTML y CSS están íntimamente ligados. Pretender posicionarse en el mundo del diseño y desarrollo web requiere el conocimiento y correcto uso de, al menos, el par de lenguajes mencionados y deseablemente el manejo de más herramientas, como *frameworks* o lenguajes de *scripting*.

Vinculación

Existen tres formas de vincular una hoja de estilo con un documento HTML (freecodecamp, 2021):

- CSS interno.
- CSS externo.
- CSS en línea.

Para el **CSS externo**, en el HTML debemos incluir una referencia al archivo de hoja de estilos (el archivo con extensión .css), dentro del elemento <link>, en cualquier línea dentro de la etiqueta <head>. Observa la línea 7 del siguiente ejemplo:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="style.css">
8   <title>Document</title>
9 </head>
10 <body>
11
12 </body>
13 </html>
```

El **CSS interno** se define dentro de la etiqueta <style>, dentro de la etiqueta <head> de una página HTML. Observa el bloque de la línea 8 a la línea 16, en la siguiente imagen:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     body {
10       background-color: white;
11     }
12
13     h1 {
14       color: black;
15     }
16   </style>
17 </head>
18 <body>
19
20 </body>
21 </html>
```

Para declarar **CSS en línea**, basta definir en la línea de la etiqueta HTML con el atributo `style`, la regla CSS deseada. Por ejemplo, observa la línea 10 del siguiente código:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1 style="font-size: 42px;">Tema 5. CSS</h1>
11 </body>
12 </html>
```

Es importante mencionar que los comentarios en CSS se realizan entre los símbolos `/*` y `*/`.

```
1 body { /* Este un comentario */
2   background-color: blue;
3   /* background-color: azure; */
4   margin: 0; /* Este es otro comentario */
5   padding: 0;
6 }
```

Maquetación

Existen técnicas simples y técnicas más especializadas para la maquetación. A continuación, exploraremos tecnologías que nos permiten maquetar bajo las especificaciones y necesidades actuales:



• Maquetación por *floats*.

Con la existencia de CSS en su versión 2, la maquetación era rudimentaria, se hacía uso de una propiedad de CSS llamada *float*, que permitía llevar cosas a la derecha o a la izquierda de la pantalla.

Cabe destacar que el sistema de *floats* no estaba pensado para crear sitios con columnas, sino para que los elementos representados en el documento se ubicaran de un lado y el texto rodeara los elementos flotantes, de la misma forma cuando una imagen era alineada a la izquierda o a la derecha de un párrafo. Esto trajo muchos problemas de diseño, por ejemplo, al tratar fondos continuos en las columnas o al intentar maquetar diseños con varias columnas.

La mayoría de esos inconvenientes lograron resolverse parcialmente mediante el uso de *hacks* o trucos avanzados en CSS, pero esos trucos requerían mucho conocimiento sobre el lenguaje, lo que conlleva un nivel de especialización muy alto para personas que no son programadores.

• Maquetación con CSS Flexbox.

Flexbox fue la primera propuesta seria que resolvía el tema del maquetado para diseñadores y desarrolladores web. Fue un nuevo modelo de caja que permite realizar el posicionamiento en filas y columnas y ayuda a ser más específico sobre el manejo en que las filas y columnas se van a distribuir y a dimensionar. Con flexbox es posible resolver el problema de la alineación vertical con mucha facilidad.

• Maquetación mediante CSS Grid Layout.

Fue en la segunda especificación de CSS en su versión 3, la que proporcionó una de las herramientas más valiosas y fáciles para realizar la maquetación con una curva de aprendizaje significativamente menor. CSS Grid Layout hace uso de un sistema de rejillas para colocar los elementos completamente personalizables mediante el uso de atributos. En el modelo de cajas, los elementos se pueden distribuir en filas y columnas completamente adaptables, lo más parecido a diseñar en papel.

Mira el siguiente video, en donde se exponen conceptos importantes sobre los diferentes estilos de maquetación:



Víctor Robles WEB. (2020, 19 de julio). *Floats vs Flexbox vs CSS Grid* 🏆 ¿Cuál es Mejor para Maquetación Web con CSS? ¿Cuál usar y cual no? 📁 [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=HHCRrxfcGLk>

El siguiente enlace es externo a la Universidad Tecmilenio, al acceder a éste considera que debes apegarte a sus términos y condiciones.

Para un diseñador web que está entrando al mundo de la maquetación, la maquetación por *floats* tendrá que ser el punto de partida, no solo porque marca las bases para el resto de las técnicas, sino porque la maquetación por *floats* sigue, al día de hoy, resolviendo de forma eficiente y fácil la mayoría de los problemas.

Cierre

La ruta correcta para aprender a maquetar depende completamente de tu proyecto, y la técnica que debes elegir para seleccionar tu estrategia de maquetación es válida según tus gustos y preferencias. Las tres técnicas de maquetación son utilizadas actualmente, y todas tienen sus bondades y sus limitaciones, principalmente en las curvas de aprendizaje y a que tanto para Flexbox como CSS Grid Layout, se debe saber y comprender muy bien la maquetación por floats.



Referencias bibliográficas

- Desarrolloweb.com. (2020). *Maquetación CSS*. Recuperado de <https://desarrolloweb.com/articulos/1771.php>
- Freecodecamp. (2021). *Hojas de estilo CSS externas: Cómo vincular CSS a HTML e importar al Head*. Recuperado de <https://www.freecodecamp.org/espanol/news/hojas-de-estilo-css-externas-como-vincular-css-a-html-e-importar-a-head/>
- MDN Web Docs. (2022). *Diseño CSS*. Recuperado de https://developer.mozilla.org/es/docs/Learn/CSS/CSS_layout

Para saber más

Lecturas

Para conocer más acerca de **maquetación con CSS**, te sugerimos leer lo siguiente:

- MDN Web Docs. (2022). *Libro de recetas de maquetación CSS*. Recuperado de: https://developer.mozilla.org/es/docs/Web/CSS/Layout_cookbook
- desarrolloweb.com. (s.f.). *Manual de Maquetación CSS*. Recuperado de: <https://desarrolloweb.com/manuales/maquetacion-css.html>

Videos

Para conocer más acerca de **maquetación por floats**, te sugerimos revisar lo siguiente:

- José Pol Lezcano. (2020, 11 de mayo). *L22 Aprendemos a colocar cajas. La propiedad CSS float*. [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=L_bctronMFO

Para conocer más acerca de **maquetación por Flexbox**, te sugerimos el siguiente video:

- Kiko Palomares. (2021, 30 de enero). *♥ Cómo hacer un LAYOUT con FLEXBOX CSS (Tutorial paso a paso)* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=IP0ZZFBSH0I>

Para saber más

Para conocer más acerca de **maquetación por SCC Grid Layout**, te sugerimos:

- Yoelvis Mullen { code }. (2020, 24 de mayo). *Maquetado responsive con CSS Grid / Diseño adaptativo #css3 #responsive #tutorial* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=T4t00Hd3qZc>

Checkpoints

Asegúrate de:

- Comprender las ideas generales de la maqueta.
- Identificar, en términos generales, la estructura de una página web.
- Comprender qué información va en cada sección de una página web.
- Diferenciar cada sección en una página web.

Requerimientos técnicos


- Computadora con acceso a Internet.
- Visual Studio Code o cualquier editor de código.

1. Observa con atención los videos propuestos en la sección “Para saber más”, en el siguiente orden:

I. Maquetación por floats:

- José Pol Lezcano. (2020, 11 de mayo). *L22 Aprendemos a colocar cajas. La propiedad CSS float* [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=L_bctronMFo

II. Maquetación por Flexbox

- Kiko Palomares. (2021, 30 de enero).  *Cómo hacer un LAYOUT con FLEXBOX CSS (Tutorial paso a paso)* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=IPOZZFBSH0I>

III. Maquetación por SCC Grid Layout

- Yoelvis Mullen { code }. (2020, 24 de mayo). *Maquetado responsive con CSS Grid | Diseño adaptativo #css3 #responsive #tutorial* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=T4t00Hd3qZc>

2. Revisa y guarda como referencia las lecturas sugeridas en la sección de “Para saber más”.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educativo y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.