



Universidad Juárez del Estado de Durango

Facultad de Ingeniería Ciencias Y Arquitectura

Ingeniería en Tecnologías Computacionales

Programación Web

Reporte de Practica 09

Diego Rea Morales

Índice

Contenido_____3

Conclusiones_____5

Contenido

En esta práctica trabajamos con un layout sencillo para una página web, utilizando HTML y CSS. La idea principal es mostrar cómo se pueden organizar diferentes secciones como el header, el menú de navegación, el contenido principal y el footer, para que la página se vea ordenada. Además, se busca que el diseño sea flexible, es decir, que se adapte a distintos tamaños de pantalla sin perder su estructura.

El layout está construido con un header en la parte superior que sirve como título, un menú de navegación a la izquierda y un área de contenido principal a la derecha. Finalmente, en la parte inferior encontramos el footer. Todo esto está organizado usando propiedades como *float* y un sistema de anchos en porcentaje, lo cual permite que los elementos se acomoden de forma proporcional.

Para hacerlo responsivo, se aplicaron *media queries* que cambian el comportamiento del diseño en pantallas más pequeñas. Por ejemplo, cuando el ancho de la pantalla es menor a 600px, tanto el menú como el contenido se muestran uno debajo del otro, logrando que el sitio sea más fácil de leer en dispositivos móviles. Esto es importante porque hoy en día la mayoría de las visitas web provienen de celulares o tablets.

```
Practica08 > index.html > html > body > header
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <title>CSS Template</title>
5    <meta charset="utf-8" />
6    <meta name="viewport" content="width=device-width, initial-scale=1" />
7    <link rel="stylesheet" href="./styles.css" />
8  </head>
9  <body>
10   <h2>CSS Layout Float</h2>
11   <p>
12     In this example, we have created a header, two columns/boxes and a footer.
13     On smaller screens, the columns will stack on top of each other.
14   </p>
15   <p>
16     Resize the browser window to see the responsive effect (you will learn
17     more about this in our next chapter - HTML Responsive.)
18   </p>
19   <header>
20     <h2>Cities</h2>
21   </header>
22   <section>
23     <nav>
24       <ul>
25         <li><a href="#">London</a></li>
26
27         <li><a href="#">Paris</a></li>
28
29         <li><a href="#">Tokyo</a></li>
30       </ul>
31     </nav>
32     <article>
33       <h1>London</h1>
34       <p>
35         London is the capital city of England. It is the most populous city in
36         the United Kingdom, with a metropolitan area of over 13 million
37         inhabitants.
38       </p>
39       <p>
40         Standing on the River Thames, London has been a major settlement for
41         two millennia, its history going back to its founding by the Romans,
42         who named it Londinium.
43       </p>
44     </article>
45   </section>
46   <footer>
47     <p>Footer</p>
48   </footer>
49 </body>
50 </html>
51
```

```

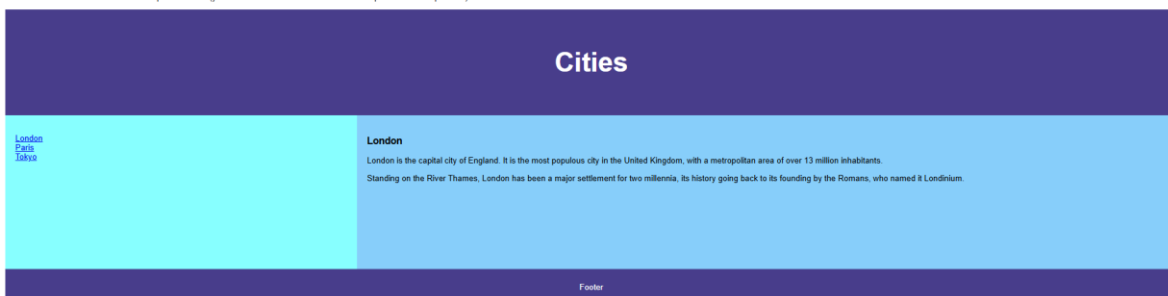
index.html M  styles.css M X
Practica08 > styles.css > {} @media (max-width: 600px) > nav
1  * {
2    box-sizing: border-box;
3  }
4  body {
5    font-family: Arial, Helvetica, sans-serif;
6  }
7  header {
8    background-color: #483d8b;
9    padding: 30px;
10   text-align: center;
11   font-size: 35px;
12   color: white;
13 }
14 nav {
15   float: left;
16   width: 30%;
17   height: 300px;
18   background-color: #87ffff;
19   padding: 20px;
20 }
21 nav ul {
22   list-style-type: none;
23   padding: 0;
24 }
25 article {
26   float: left;
27   padding: 20px;
28   width: 70%;
29   background-color: #87cefa;
30   height: 300px;
31 }
32 section::after {
33   content: "";
34   display: table;
35   clear: both;
36 }
37 footer {
38   background-color: #483d8b;
39   padding: 10px;
40   text-align: center;
41   color: white;
42 }
43 @media (max-width: 600px) {
44   nav,
45   article {
46     width: 100%;
47     height: auto;
48   }
49 }
50

```

Código CSS con el funcionamiento de la página

CSS Layout Float

In this example, we have created a header, two columns/boxes and a footer. On smaller screens, the columns will stack on top of each other. Resize the browser window to see the responsive effect (you will learn more about this in our next chapter - HTML Responsive.)



Conclusiones

En resumen, esta práctica demuestra cómo con unas cuantas reglas de CSS se puede lograr un diseño web responsivo. Aunque es un ejemplo básico, deja claro que no se necesitan herramientas avanzadas para adaptar un sitio a diferentes dispositivos. A partir de esta base, se pueden agregar más estilos y técnicas modernas como *flexbox* o *grid*, pero lo importante es entender primero cómo funciona lo esencial del layout.