

## Week 4\_1(es)

---

# Float y Clear

---

Aprendimos sobre box model en CSS. Hoy, vamos a aprender cómo podemos modificar la posición de los elementos de HTML en una página en varias maneras!

Antes de empezar, hagan una carpeta nombrada `fecha-su nombre` en escritorio o en su llave. Luego copien y peguen o descarguen siguientes archivos.

[index.html](#)

[seoul.jpg](#)

Adicionalmente, en la carpeta, hagan un archivo nombrado `blog.css`.

## HTML Flow

---

Como nos dimos cuenta cuando utilizamos el código puro de HTML, un documento de HTML ya tiene sus reglas propias sobre *cómo demostrar y posicionar* los elementos sin aplicar CSS.

- **fluidity**: cómo los contenidos se modifican según las dimensiones del navegador
- **ordering**: el orden en el cual los elementos se aparecen
- **stacking**: cómo los elementos se aparecen encima de otros

Este valor por defecto se llama *flow* de los elementos de HTML.

Podemos utilizar varias propiedades de CSS properties como `height`, `width`, `float`, `position`, `z-index`, `display`, y `clear` para **cambiar** el flow.

## Float

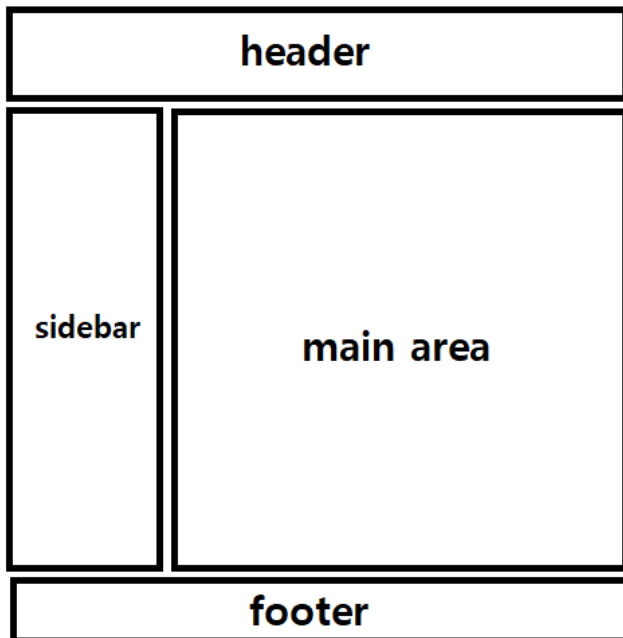
---

Ahora, la estructura de la página parece como esto.



Todos los elementos están ubicados encima o abajo de otros, cada uno ocupando el espacio entero horizontal.

Vamos a añadir un **sidebar** que nos permite navegar a otras páginas como se muestra abajo.



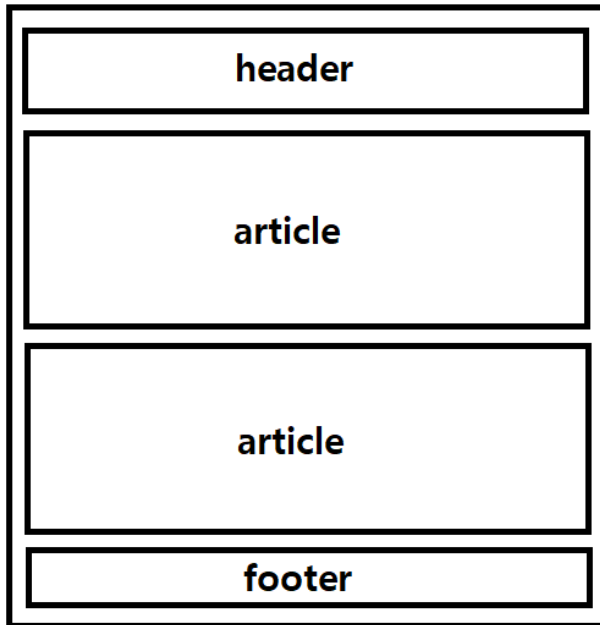
Aquí, el sidebar y el área principal están ubicados al lado del otro.

Antes de crear un sidebar, vamos a hacer que el contenido de la página tenga el ancho máximo (maximum width).

Es importante porque es difícil para los ojos leer un texto que ocupa todo el espacio horizontal de una pantalla.

Para hacerlo, vamos a crear un `<div>` con `class="container"` y vamos a poner todo el contenido dentro

del elemento `<body>` que está dentro de `<div>`.



**div class="container"**

Luego, vamos a agregar el código siguiente.

```
.container {  
  max-width: 800px;  
  margin-left: auto;  
  margin-right: auto;  
}
```

Si ponen la propiedad `max-width` a un elemento, el elemento tendrá el ancho máximo de 800 píxeles, y no más grande.

Si ponen la propiedad de `margin-left` y la propiedad de `margin-right` como `auto`, el contenido estará situado en el centro horizontalmente.

## width vs. max-width

Como mencionó antes, un elemento de block-level ocupa toda la línea.

Para cambiarlo (*break the flow*), necesitamos utilizar la propiedad `width` o `max-width`.

Si utilizan la propiedad `width` (con píxeles), el elemento tendrá un ancho fijo que *no se cambia* según el tamaño del navegador.

Pero, si utilizan la propiedad `max-width`, el tamaño del elemento va a *cambiarse* cuando el ancho del navegador es más pequeño del valor de `max-width`.

`max-width` se usa cuando hacen un web para los aparatos pequeños.

Vamos a cambiar el tamaño del navegador a menos que el ancho de 800 píxeles, y vamos a ver cómo el ancho del contenido se cambia!

**Nota:** Si ponen solo `max-width`, el ancho del elemento tiene que estar especificado con %, no con píxeles. Es porque ponerlo en píxeles tal vez resulta en exceso del contenido en los aparatos pequeños.

Ahora, vamos a agregar un elemento para sidebar encima del elemento `<article>`.

Vamos a utilizar el elemento `<nav>` porque el sidebar va a servir como una barra de navegación para navegar las varias categorías de la página.

```
<nav class="sidebar">
  <ul>
    <li><a href="cities.html">Cities</a></li>
    <li><a href="movies.html">Movies</a></li>
    <li><a href="books.html">Books</a></li>
  </ul>
</nav>
```

Todavía no tenemos los archivos de HTML que están en `href`, por eso no hay ningún cambio cuando hacemos clic en los elementos `<a>`, pero no se preocupen!

Solo queremos tener un sidebar en la página por ahora.

Luego, vamos a crear un elemento `<div>` con `class="main-area"` y vamos a poner dos elementos `<article>` dentro del elemento `<div>`.

Ahora, se ve el sidebar encima del área principal.

Para hacer que el sidebar y el área principal se situen al lado del otro, necesitamos utilizar la propiedad `float`.

La propiedad `float` mueve(o *floats*) los elementos a la izquierda o a la derecha. Puede tener un valor entre, `left` o `right`.

Vamos a agregar el siguiente código de CSS.

```
.sidebar {
  width: 20%;
  float: left;
  background-color: #466995;
}

.main-area {
  width: 80%;
  float: left;
  background-color: #DBE9EE;
}
```

El `background-color` está añadido para que podamos ver el área de cada elemento.

Si revisamos el navegador, podemos ver que el sidebar y el área principal están situado horizontalmente.

**Nota:** Los elementos con `float` debe tener el ancho especificado. Si no, el elemento tendrá un ancho completo y no se cambia nada aunque tiene un valor de `float`.

## Clear

---

Si los elementos con float en una página tienen las alturas diferentes, eso puede afectar al diseño de la página.

La propiedad clear decide en cuál lado no se puede tener otros elementos.

Puede tener los cuatro valores:

- `none` - Permite tener elementos en ambos lados (por defecto)
- `left` - No permite tener elementos en el lado izquierda
- `right` - No permite tener elementos en el lado derecha
- `both` - No permite tener elementos en ambos lados
- `inherit` - El elemento tiene el mismo valor de clear como su padre

Para revisar la propiedad `clear`, vamos a cambiar el ancho del `área principal` a `"70%"`.

Luego, el ancho del sidebar y el área principal suman a 90% de la pantalla y queda el 10% del espacio. Si revisan el navegador, pueden ver que el footer ya está ubicado justo al lado del área principal para llenar el espacio.

Para evitarlo, podemos usar la propiedad `clear` en elemento `<footer>`.

Vamos a agregar `class="blog-footer"` al elemento footer con el nombre de corporación.

Luego, vamos a agregar el siguiente código de CSS.

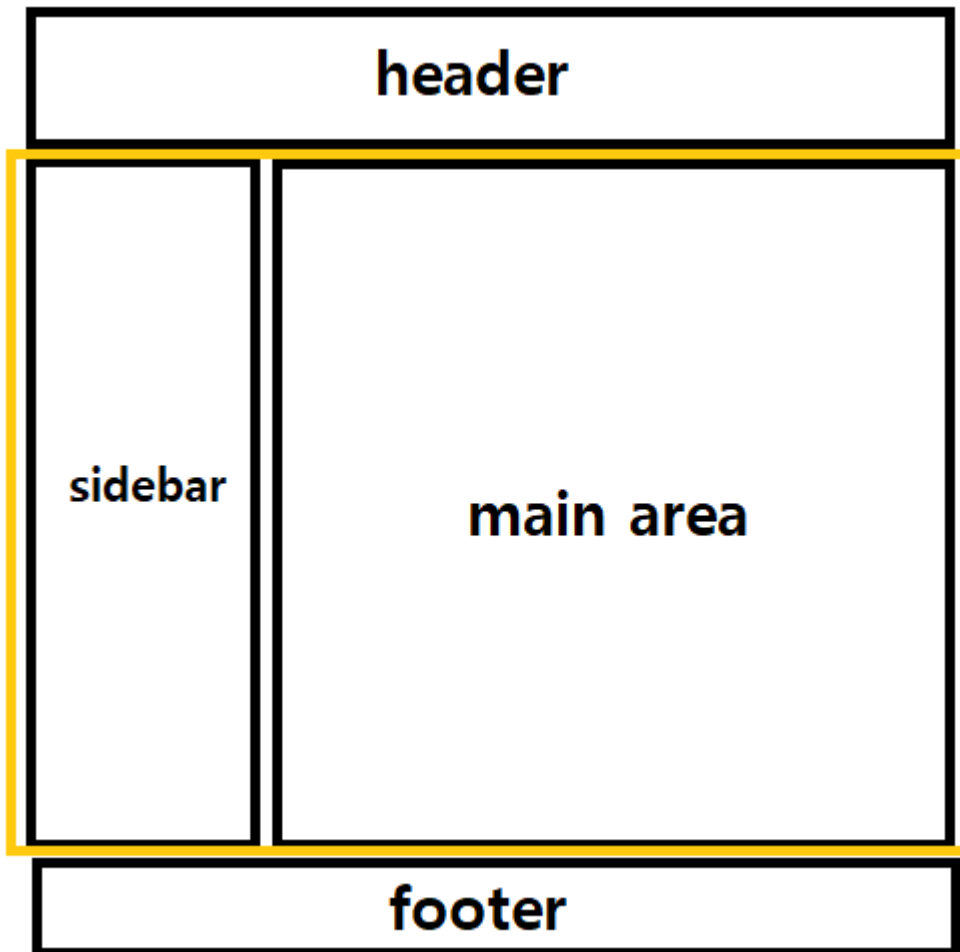
```
footer {  
  clear: left;  
}
```

Verán que el footer está en el final de la página como antes.

Ahora, vamos a cambiar el ancho del área principal a 80% de nuevo.

## El método Clearfix

Ahora vamos a agregar el borde al sidebar y al área principal como se muestra abajo.



Para hacerlo, vamos a crear un elemento `<div class="content-area">` y poner `<nav class="sidebar">` y `<div class="main-area">` dentro del elemento.

Luego, vamos a agregar el siguiente código.

```
.content-area {  
  border: 3px solid black;  
  border-radius: 5px;  
}
```

Si revisan el navegador, solo pueden ver el borde en la parte más alta.

Por qué sucede eso?

Es porque el elemento padre (`content-area`) no reconoce que los elementos hijos tienen `float` (`sidebar` y el `área principal`).

Es como los elementos con `float up` desde su padre y hermanos y por eso no los reconocen.

Hay varios métodos para arreglarlo. Uno de ellos es *el método de clearfix*.

Vamos a agregar la clase `clearfix` al div elemento `content-area`.

Luego, vamos a agregar el siguiente código de CSS.

```
.clearfix:before,  
.clearfix:after {  
  content: "";  
  display: table;  
}  
  
.clearfix:after {  
  clear: both;  
}  
  
.clearfix {  
  zoom: 1;  
}
```

No es necesario memorizarlo, sino pueden copiar y pegarlo cada vez cuando tienen este problema en usar `float`.

Ahora pueden ver que todo el borde parece bien.

Podemos reutilizar la clase `clearfix` cuando es necesario que un elemento reconozca sus elementos hijos en una página con `float`.