

Bootcamp Java Developer

Fase 2 - Java Web Developer
Módulo 15



Flex, Float y Grid

Uso de Float

Si bien el uso de **float** en CSS ha disminuido notablemente hasta hacerlo casi inexistente en algunos casos, podemos observar que Bootstrap tiene una clase asociada a esta propiedad.

El trabajo con **float-start** (para afectar al float left), **float-end** (para afectar al float right) o **float-none** (para trabajar con float none) nos permite, a través de clases, estructurar un *layout* utilizando Bootstrap.

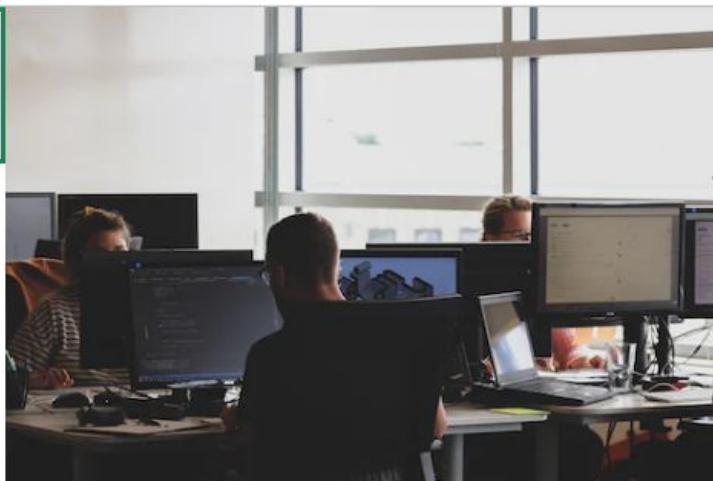
Ejemplo

```
<div class="float-end">Contenedor  
con float:left</div>
```



El uso que actualmente podemos darle es cuando tenemos en nuestro *layout* un texto y una imagen, por ejemplo:

Lorem ipsum dolor sit amet consectetur adipisicing elit. Reiciendis debitis iusto minima similique dolore quaerat, ea sunt magni culpa impedit vel, cumque quod mollitia iure odio repellendus obcaecati voluptate veritatis?



Para lograr el resultado anterior, utilizamos la siguiente estructura desde nuestro HTML:

```
<p class="float-start border border-success  
border-3">Lorem ipsum dolor sit amet consectetur  
adipiscing elit.  
    Reiciendis debitis iusto minima similique dolore  
    quaerat, ea sunt magni culpa impedit vel, cumque  
    quod mollitia  
    iure odio repellendus obcaecati voluptate  
    veritatis?</p>  
  

```

Como muchas clases (no todas) de Bootstrap encontramos la posibilidad también de **combinar esta clase con los ya conocidos *breakpoints***. Por ejemplo, un elemento puede estar hacia la derecha en pantallas superiores a 768px y hacia la izquierda en pantallas inferiores.

```
<div class="m-4 float-start float-md-end border  
border-success border-3">Lorem </div>
```

Nota: en el ejemplo hemos implementado **m** para **margin** y bordes para poder visualizar los elementos correctamente.



El resultado del código anterior será el siguiente:



Pantallas
inferiores
a 768px



Pantallas
superiores
a 768px

Uso de Flex

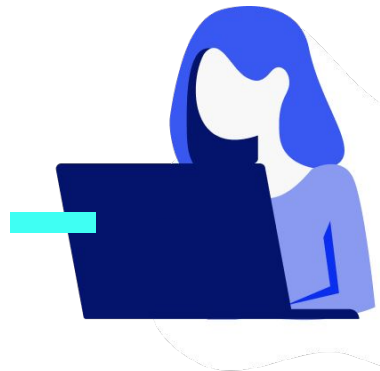
Flex nos permite trabajar con esta ya conocida propiedad más allá de no hacerlo con columnas y filas como hemos aprendido. Es decir, solamente implementamos `display:flex` sin contar con un sistema de grillado.

```
<div class="d-flex">Elemento con flex</div>
```

Nota: tenemos varios elementos que utilizan el prefijo **d**, dado que hace referencia a la propiedad **display**, por ejemplo, si quiero transformar una imagen en elemento de bloque.

```

```



Alineación con flex

Podemos como cualquier elemento que posea **flex**, utilizar **alineación tanto horizontal como vertical** por ejemplo:

```
<body>  
<main class="w-100 d-flex justify-content-center">  
  <p class="border border-success">lorem</p>  
  <p class="border border-success">lorem</p>  
</main>
```

Resultado:

lorem	lorem
-------	-------

Nota: hemos utilizado **w-100** para darle al contenedor **main** un **width** del 100% y un **border** a los párrafos para poder visualizarlos.

Dirección con flex

También desde **flex** podemos modificar el **direction** a **column**, **column-reverse** o **row-reverse**, dependiendo la orientación del *layout*, por ejemplo:

```
<body>
<main class="w-100 d-flex flex-row-reverse
justify-content-center">
  <p class="border border-success">lorem 1</p>
  <p class="border border-success">lorem 2</p>
</main>
```

Resultado:

lorem 2	lorem 1
---------	---------

Además, podemos **asociar las clases a los *breakpoints***, por ejemplo, para que en pantallas pequeñas los elementos estén encolumnados y en pantallas iguales o superiores a 768px estén con un **row-reverse** de dirección.

```
<body>
<main class="w-100 d-flex flex-column flex-md-row-reverse
justify-content-center">
  <p class="border border-success">lorem 1</p>
  <p class="border border-success">lorem 2</p>
</main>
```

Uso de Grid

En Bootstrap también podemos implementar **grid**, como lo hacemos manualmente en css. De esta forma en diseños, por ejemplo, de galerías de imágenes o *layouts* similares a tablas, si no queremos trabajar con **table**, que a veces nos genera mucha complejidad, podemos utilizar en vez del sistema de grillas ya conocido o la clase **d-flex**, esta clase llamada **grid**.

Nota: **grid** está en modo experimental, por eso, para poder utilizarlo debemos **activarlo desde nuestro propio Sass**. Por este motivo, por ahora sólo haremos una mención general y recomendamos revisar la documentación oficial ([CSS Grid - Bootstrap v5.2](#)) para futuros avances, aunque aún no es una propiedad que será tomada como estándar en este curso.

Ejemplo

```
<div class="grid text-center">
  <div class="g-col-4">.g-col-4</div>
  <div class="g-col-4">.g-col-4</div>
  <div class="g-col-4">.g-col-4</div>
</div>
```

**¡Sigamos
trabajando!**