

# Unidad 4: Bootstrap

- 2º DAW - Mari Cruz Gómez García
- Curso 2023/2024

## ÍNDICE

- 1) Desbordamientos
- 2) Display
- 3) Visibility
- 4) Posicionamientos
- 5) Posicionamiento flotante
- 6) Opacidad
- 7) Layout de Bootstrap
- 8) Alineación vertical
- 9) Alineación horizontal
- 10) Offset
- 11) Column break
- 12) Margen en el layout de Bootstrap
- 13) Orden de elementos

- 1) Bootstrap flexbox
- 2) Alineación con flexbox
- 3) Alineación multilínea
- 4) Flexbox fill, Grow, Shrink
- 5) Orden de elementos con flexbox

# 1. Desbordamientos

- <https://getbootstrap.com/docs/5.3/utilities/overflow/>
- Ver ejemplo overflow.html

```
<style>
  .container div{
    width: 200px;
    height: 100px;
    background: #f4f4f4;
    display: inline-block;
    margin: 0 15px;
  }
</style>
```

```
<div class="container mt-5">
  <div class="overflow-auto">Lorem ipsum dolor sit amet consectetur, adipisicing elit.
    A incidunt, temporibus quaerat ut ipsum commodi repudiandae.
    Nam illo tempore quisquam iusto beatae eligendi officia doloribus, eos, quis officiis facere nihil!</div>

  <div class="overflow-hidden">Lorem ipsum dolor sit amet consectetur adipisicing elit. Esse, at commodi laboriosam c

  <div class="overflow-visible">Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati pariatur esse reru

  <div class="overflow-scroll">Lorem ipsum dolor sit amet consectetur adipisicing elit. Tenetur sed praesentium quod
</div>
```

# 2. Display

- <https://getbootstrap.com/docs/5.3/utilities/display/>

```
<h1>Display block</h1>
<div class="d-block"></div>
<div class="d-block"></div>
<div class="d-block"></div>
<div class="d-block"></div>
<div class="d-block"></div>
```

Display block



```
<h1>Display Inline block</h1>
<div class="d-inline-block"></div>
<div class="d-inline-block"></div>
<div class="d-inline-block"></div>
<div class="d-inline-block"></div>
```

Display Inline block



```
<h1>Display Table cell</h1>
<div class="d-table-cell"></div>
<div class="d-table-cell"></div>
<div class="d-table-cell"></div>
<div class="d-table-cell"></div>
```

Display Table cell



- ¿Cómo hacer que se visualice en display block por defecto pero en dispositivos mayores al tamaño mediano, no lo van a desplegar pero los dispositivos grandes hacia arriba, que lo visualice en formato display inline block?

## 2. Display

- Compruébalo con las herramientas DevTools.

```
<!--  
.d-{breakpoint}-{value} for sm, md, lg, xl, and xxl.  
-->  
  
<h1>Display Con breakpoints</h1>  
<div class="d-block d-md-none d-lg-inline-block"></div>  
<div class="d-block d-md-none d-lg-inline-block"></div>  
<div class="d-block d-md-none d-lg-inline-block"></div>
```

## 3. Propiedad visibility

- <https://getbootstrap.com/docs/5.3/utilities/visibility/>
- Los elementos con la propiedad visibility a valor hidden, no van a mostrarse.
- ¿Cuál es la diferencia con display-none?

### 3. Propiedad visibility

- Los elementos que tengan la propiedad display:none, van a ocultar el elemento, pero también van a quitar el espacio reservado para él.
- En cambio, los elementos que tienen la propiedad visibility a hidden, no se muestran pero sí mantienen el espacio que ocupaban.

### 3. Propiedad visibility

```
<div class="container vh-100">  
  <div class="w-100 h-25 bg-success d-inline-block"></div>  
  <div class="w-100 h-25 bg-primary d-inline-block"></div>  
  <div class="w-100 h-25 bg-warning d-inline-block"></div>  
</div>
```



```
<div class="container vh-100">  
  <div class="w-100 h-25 bg-success d-inline-block"></div>  
  <div class="w-100 h-25 bg-primary d-inline-block d-none"></div>  
  <div class="w-100 h-25 bg-warning d-inline-block"></div>  
</div>
```



```
<div class="container vh-100">  
  <div class="w-100 h-25 bg-success d-inline-block"></div>  
  <div class="w-100 h-25 bg-primary d-inline-block invisible"></div>  
  <div class="w-100 h-25 bg-warning d-inline-block"></div>  
</div>
```



## 4. Posicionamientos

- <https://getbootstrap.com/docs/5.3/utilities/position/>
- Tenemos los posicionamientos, absolutos, relativos, estáticos, fijados y pegajoso.
- Para poder posicionar un elemento en relación a otro recordamos que el contenedor padre tiene que ser relativo y el hijo absoluto.
- Podemos aplicar además para moverlo:

```
<div class="container vh-100 position-relative">  
  <div class="position-absolute top-50 start-50 translate-middle"></div>  
</div>
```

Permite centrar el contenido

These properties are one of:

- **top** - for the vertical **top** position
- **start** - for the horizontal **left** position (in LTR)
- **bottom** - for the vertical **bottom** position
- **end** - for the horizontal **right** position (in LTR)

Where *position* is one of:

- **0** - for **0** edge position
- **50** - for **50%** edge position
- **100** - for **100%** edge position

## 5. Posicionamiento flotante

- <https://getbootstrap.com/docs/5.3/utilities/float/>

```
<div class="bg-primary float-start"></div>  
<div class="bg-success float-end"></div>  
<div class="bg-warning float-start"></div>
```

## 6. Opacidad

- <https://getbootstrap.com/docs/5.3/utilities/opacity/>
- Puedes indicar el nivel de transparencia:

```
<div class="opacity-100"></div>  
<div class="opacity-75"></div>  
<div class="opacity-50"></div>  
<div class="opacity-25"></div>
```

## 7. Layout

- <https://getbootstrap.com/docs/5.3/layout/grid/>
- El sistema de rejilla de Bootstrap se basa en rejillas flexibles, las cuales se escalan al tamaño y posición adecuadas dependiendo de la pantalla en la que se esté visualizando la web.
- El sistema de rejilla tiene que ser utilizado dentro de uno de los dos elementos contenedores que tiene Bootstrap: **.container** o **.container-fluid**. Es importante tener en cuenta que estos elementos se utilizan como raíz de la rejilla y no se podrán anidar unos dentro de otros.
- Si lo que queremos es que el contenido de nuestra web aparezca centrado y con un ancho fijo, entonces utilizamos la clase **.container**. En cambio, si queremos que ocupe todo el ancho disponible (width: 100%), usamos la clase **.container-fluid**.

## 7. Layout


- El Sistema de rejilla está pensado para ayudarnos en la disposición del contenido de nuestra web y su adaptación a las diferentes pantallas.
- Para ello, tenemos que poner el contenido dentro de celdas o columna que irán dentro de filas. **Cada fila se puede dividir hasta en 12 columnas**, somos nosotros los que definimos las columnas para cada tamaño de pantalla.
- El funcionamiento de las rejillas es:
  - Las columnas deben ir agrupadas dentro de filas (.row)
  - Las filas (.row) se deben colocar dentro de una etiqueta contenedora: .container (para ancho fijo) o .container-fluid (para poder ocupar todo el ancho), esto permitirá alinear las celdas y asignarles el espacio correcto.

## 7. Layout

- **El contenido se debe disponer dentro de columnas o celdas**, las cuales deben de ser el único hijo posible de las filas (.row), que a su vez será el único hijo posible del contenedor (.container ó .containerfluid)
- Al seguir este orden el sistema de rejilla funcionará creando el espaciado interior y los márgenes apropiados dependiendo de las dimensiones de la pantalla.
- **Cada fila se puede dividir hasta un máximo de 12 columnas**, pero somos nosotros los que tendremos que definir el número de columnas en el que queremos dividir cada fila y su ancho para cada tamaño de pantalla.
- **Si el tamaño total de las columnas de una fila excede de 12, el sobrante se colocará en la siguiente fila.**
- El tamaño de las columnas se especificará con clases CSS que Bootstrap define para cada tamaño de pantalla, por ejemplo .colmd-XX donde XX es el tamaño de la columna, que podrá tomar valores entre 1 y 12. Si quieres que todas las columnas tengan el mismo tamaño se usaría: .col-4 por ejemplo

## 7. Layout

- <https://getbootstrap.com/docs/5.3/layout/grid/>
- Ejemplo:



```
HTML
<div class="container text-center">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>
```

Si usamos la clase col-4, esa columna ocuparía 4 de los 12 espacios

## 7. Layout

- Además, se pueden usar los breakpoints para especificar que una columna ocupe 12 espacios en tamaños medianos para arriba y 6 espacios en dispositivos más pequeños.

Pantalla (tamaño)	Prefijo de la clase	Ancho del contenedor							
Extra pequeño (<576px)	.col-	Ninguno (automático)							
Pequeño (≥576px)	.col-sm-	540px	xs	sm	md	lg	xl	xxl	
Mediano (≥768px)	.col-md-	720px	<576px	≥576px	≥768px	≥992px	≥1200px	≥1400px	
Largo (≥992px)	.col-lg-	960px							
Extra largo (≥1200px)	.col-xl-	1140px							
			Container max-width	None (auto)	540px	720px	960px	1140px	1320px
			Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
			# of columns	12					
			Gutter width	1.5rem (.75rem on left and right)					
			Custom gutters	<a href="#">Yes</a>					
			Nestable	<a href="#">Yes</a>					
			Column ordering	<a href="#">Yes</a>					



## 7. Layout - ejemplo

- Tenemos el siguiente código:

```
<div class="container py-4">  
  <div class="bg-light p-4 border">  
    <h1>Hola mundo</h1>  
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Cupiditate, eaque.  
      Consequuntur corporis culpa, sit deserunt laborum animi dolore eos blanditiis  
      soluta optio quia, quibusdam perspiciatis architecto excepturi eaque ipsam reiciendis.</p>  
    <button class="btn btn-primary">Click aquí</button>  
  </div>  
</div>
```

Clase base container con padding arriba y abajo de 4 (que es 1,5 rem)

### Hola mundo

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cupiditate, eaque. Consequuntur corporis culpa, sit deserunt laborum animi dolore eos blanditiis soluta optio quia, quibusdam perspiciatis architecto excepturi eaque ipsam reiciendis.

Click aquí

## 7. Layout - ejemplo

- Lo que queremos conseguir es:

### Hola mundo

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cupiditate, eaque. Consequuntur corporis culpa, sit deserunt laborum animi dolore eos blanditiis soluta optio quia, quibusdam perspiciatis architecto excepturi eaque ipsam reiciendis.

Click aquí

### Hola mundo

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cupiditate, eaque. Consequuntur corporis culpa, sit deserunt laborum animi dolore eos blanditiis soluta optio quia, quibusdam perspiciatis architecto excepturi eaque ipsam reiciendis.

Click aquí

### Hola mundo

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cupiditate, eaque. Consequuntur corporis culpa, sit deserunt laborum animi dolore eos blanditiis soluta optio quia, quibusdam perspiciatis architecto excepturi eaque ipsam reiciendis.

Click aquí

## 7. Layout

- Si lo distribuimos así, él detecta que hay una fila con 3 columnas y reparte el espacio automáticamente.

## 7. Layout

- ¿Qué pasa si lo ponemos así?

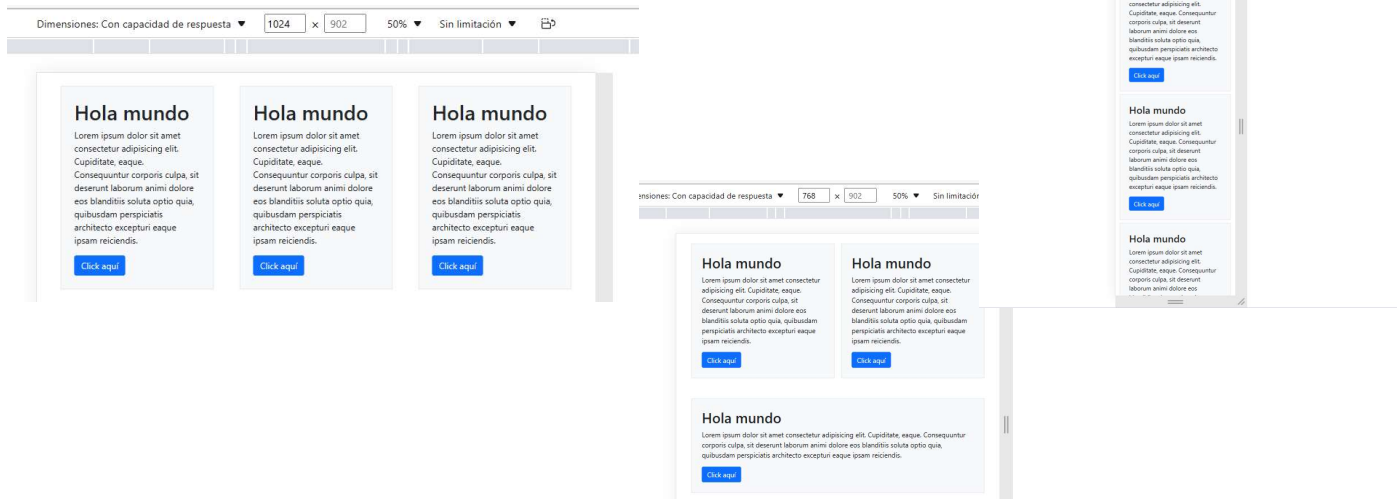
```
<!-- GRID LAYOUT -->
<div class="container py-4">
  <div class="row">
    <div class="col-6">
      <div class="bg-light p-4 border">
        <h1>Hola mundo</h1>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing
        <button class="btn btn-primary">Click aquí</button>
      </div>
    </div>
    <div class="col-6">
      <div class="bg-light p-4 border">
        <h1>Hola mundo</h1>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing
        <button class="btn btn-primary">Click aquí</button>
      </div>
    </div>
    <div class="col">
      <div class="bg-light p-4 border">
        <h1>Hola mundo</h1>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing
        <button class="btn btn-primary">Click aquí</button>
      </div>
    </div>
  </div>
</div>
```

## 7. Layout

- Prueba a redimensionar en distintos tamaños con DevTools. ¿Se adaptan bien las columnas?
- Solución: breakpoints
- Aplicamos por ejemplo que:
  - Para tamaños de pantalla grandes (>992px) → que se dividan en 3 columnas
  - Para tamaños medianos (>768px) → que se dividan en dos columnas y la tercera columna que ocupe todo el espacio
  - Para tamaños pequeños → que cada div de texto ocupe el ancho máximo

## 7. Layout - ejemplo

- Ver ejemplo grid.html



## 7. Layout - ejemplo

```
<div class="col-lg-4 col-md-6 col-12">
  <div class="bg-light p-4 border">
    <h1>Hola mundo</h1>
    <p>Lorem ipsum dolor sit amet consectetur</p>
    <button class="btn btn-primary">Click
  </div>
</div>

<div class="col-lg-4 col-md-6 col-12">
  <div class="bg-light p-4 border">
    <h1>Hola mundo</h1>
    <p>Lorem ipsum dolor sit amet consectetur</p>
    <button class="btn btn-primary">Click
  </div>
</div>

<div class="col-lg-4 col-md-12 col-12">
  <div class="bg-light p-4 border">
    <h1>Hola mundo</h1>
    <p>Lorem ipsum dolor sit amet consectetur</p>
    <button class="btn btn-primary">Click
  </div>
</div>
```

## 7. Layout - ejemplo

- Un pequeño detalle → la separación entre las columnas en el grid en CSS se hacía con gap.
- Ahora se hace con gutters: <https://getbootstrap.com/docs/5.3/layout/gutters/>
- g → para el eje X (horizontal) e Y(vertical)
- gx
- Gy
- Admite valores entre 1 y 5
- También admite breakpoints

```
<!-- GRID LAYOUT -->
<div class="container py-4">
  <div class="row gx-3 gx-lg-5 gy-2 gy-md-5">
    <div class="col-lg-4 col-md-6 col-12">
      <div class="bg-light p-4 border">
        <h1>Hola mundo</h1>
        <p>Lorem ipsum dolor sit amet consectetur</p>
        <button class="btn btn-primary">Click aquí
      </div>
    </div>
```

## 7. Layout - ejemplo

- También puedes usar la clase row-cols para establecer el número de columnas de cada fila:

```
!-- GRID ROW-COLS -->


<div class="row row-cols-1 row-cols-md-2 row-cols-lg-3">
    <div class="col">
      <div class="bg-light p-4 border">
        <h1>Hola mundo</h1>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing
        <button class="btn btn-primary">Click aquí</butt
      </div>
    </div>
  </div>


```

- <https://getbootstrap.com/docs/5.3/layout/grid/#row-columns>

## 8. Alineación vertical

- <https://getbootstrap.com/docs/5.3/layout/columns/>
- Ver ejemplo alineacionV.html
- row align-items-start
- row align-items-center
- row align-items-end
- col align-self-start
- col align-self-center
- col align-self-end



```
<div class="container">

  <!-- Alineacion vertical -->
  <div class="row align-items-start">
    <div class="col">Columna 1</div>
    <div class="col">Columna 2</div>
    <div class="col">Columna 3</div>
  </div>

  <div class="row align-items-center">
    <div class="col">Columna 1</div>
    <div class="col">Columna 2</div>
    <div class="col">Columna 3</div>
  </div>

  <div class="row align-items-end">
    <div class="col">Columna 1</div>
    <div class="col">Columna 2</div>
    <div class="col">Columna 3</div>
  </div>

  <div class="row">
    <div class="col align-self-start">Columna 1</div>
    <div class="col align-self-center">Columna 2</div>
    <div class="col align-self-end">Columna 3</div>
  </div>
</div>
```

## 9. Alineación horizontal

- <https://getbootstrap.com/docs/5.3/layout/columns/#horizontal-alignment>
- row justify-content-start
- row justify-content-center
- row justify-content-end
- row justify-content-around
- row justify-content-between
- row justify-content-evenly

## 9. Alineación horizontal

- <https://getbootstrap.com/docs/5.3/layout/columns/>



```
<div class="container">

  <!-- Alineación horizontal -->
  <div class="row justify-content-start">
    <div class="col-2">Columna 1</div>
    <div class="col-2">Columna 2</div>
    <div class="col-2">Columna 3</div>
  </div>

  <div class="row justify-content-center">
    <div class="col-2">Columna 1</div>
    <div class="col-2">Columna 2</div>
    <div class="col-2">Columna 3</div>
  </div>

  <div class="row justify-content-end">
    <div class="col-2">Columna 1</div>
    <div class="col-2">Columna 2</div>
    <div class="col-2">Columna 3</div>
  </div>

  <div class="row justify-content-around">
    <div class="col-2">Columna 1</div>
    <div class="col-2">Columna 2</div>
    <div class="col-2">Columna 3</div>
  </div>

  <div class="row justify-content-between">
    <div class="col-2">Columna 1</div>
    <div class="col-2">Columna 2</div>
    <div class="col-2">Columna 3</div>
  </div>

  <div class="row justify-content-evenly">
    <div class="col-2">Columna 1</div>
    <div class="col-2">Columna 2</div>
    <div class="col-2">Columna 3</div>
  </div>

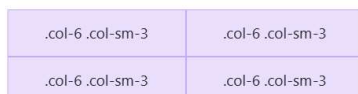
</div>
```

## 10. Offset

- <https://getbootstrap.com/docs/5.3/layout/columns/#offset-classes>
- Permite mover columnas con la clase offset que indica el desplazamiento.
- col offset-2 → le indica a esa columna que se desplace 2 columnas

## 11. Column break

- <https://getbootstrap.com/docs/5.3/layout/columns/#column-breaks>



.col-6.col-sm-3	.col-6.col-sm-3
.col-6.col-sm-3	.col-6.col-sm-3

HTML

```
<div class="container text-center">
  <div class="row">
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
    <!-- Force next columns to break to new line -->
    <div class="w-100"></div>
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
  </div>
</div>
```

## 12. Margen en el layout de Bootstrap

- <https://getbootstrap.com/docs/5.3/layout/columns/#margin-utilities>
- Clase margin start auto → ms-auto (aplica margen a la izquierda y lleva ese elemento hacia la columna más a la derecha posible)
- Clase margin end auto → me-auto (igual pero al revés)

## 13. Orden de los elementos

- <https://getbootstrap.com/docs/5.3/layout/columns/#reordering>
- Clases:
- Order-first → va el primero
- Order-last → va el último
- Order-número → especifica el orden



# EJEMPLO GUIADO

Vamos a pasar del diseño de la izquierda al de la derecha, SOLO con clases de Bootstrap, Sin nada de código CSS

## Header

### Aside

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Molestias eum velit unde ipsam dicta, aspernatur pariat vitae fugiat illo quis aliquid adipisci praesentium laboriosam. Corporis adipisci aliquam sunt unde in!

### Contenido

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eos animi nesciunt mollitia impedit eligendi asperiores, iusto, odit provident molestiae saepe totam dignissimos amet voluptas dolore expedita cupiditate error a officia.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Ea dignissimos ullam, iure error dolore iusto fuga ipsum a ratione voluptates commodi, nihil aperiam soluta animi? Neque facilis sapiente illo soluta!

Lorem ipsum dolor sit amet consectetur adipiscing elit. Ea dignissimos ullam, iure error dolore iusto fuga ipsum a ratione voluptates commodi, nihil aperiam soluta animi? Neque facilis sapiente illo soluta!

Lorem ipsum dolor sit, amet consectetur adipiscing elit. amet consectetur adipiscing elit.  
Lorem ipsum dolor sit, amet consectetur adipiscing elit. amet consectetur adipiscing elit.  
Lorem ipsum dolor sit, amet consectetur adipiscing elit. amet consectetur adipiscing elit.

## Footer

## HEADER

### Aside

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Molestias eum velit unde ipsam dicta, aspernatur pariat vitae fugiat illo quis aliquid adipisci praesentium laboriosam. Corporis adipisci aliquam sunt unde in!

### Contenido

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eos animi nesciunt mollitia impedit eligendi asperiores, iusto, odit provident molestiae saepe totam dignissimos amet voluptas dolore expedita cupiditate error a officia.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Ea dignissimos ullam, iure error dolore iusto fuga ipsum a ratione voluptates commodi, nihil aperiam soluta animi? Neque facilis sapiente illo soluta!

Lorem ipsum dolor sit amet consectetur adipiscing elit. Ea dignissimos ullam, iure error dolore iusto fuga ipsum a ratione voluptates commodi, nihil aperiam soluta animi? Neque facilis sapiente illo soluta!

Lorem ipsum dolor sit, amet consectetur adipiscing elit. amet consectetur adipiscing elit.

Lorem ipsum dolor sit, amet consectetur adipiscing elit. amet consectetur adipiscing elit.

Lorem ipsum dolor sit, amet consectetur adipiscing elit. amet consectetur adipiscing elit.

## FOOTER

# EJEMPLO GUIADO

- Primero centramos el contenido aplicando la clase container y dejando un padding top de 4 para que se separe de arriba: `<div class="container pt-4">`
- Ahora vamos a aplicar nuestro esquema de grid y estilos a cada contenedor.
- Header:
- Le indicamos que ocupe la fila entera y el contenido que se expanda las 12 columnas, ponemos el color de fondo negro, el texto blanco, alineado y en mayúsculas:

```
<div class="container pt-4">
  <header class="row">
    <div class="col-12 bg-dark text-white text-center text-uppercase">
      <h1>Header</h1>
    </div>
  </header>
```

## EJEMPLO GUIADO

- Sección aside y contenido:
- Esa sección de nuevo debe ocupar una fila completa y le decimos py-4 para que se separe arriba y abajo.
- Vamos a repartir 4 columnas para el aside y 8 para el contenido. Aplicamos breakpoint para cuando los dispositivos sean más pequeños, cada parte ocupe las 12 columnas.

```
<section class="row py-4">
  <aside class="col-12 col-md-4 bg-light p-3 border order-2 order-md-1 d-none d-sm-block">
    <h2 class="text-center">Aside</h2>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Molestias eum velit unde ipsam dicta
  </aside>

  <div class="col-12 col-md-8 order-1 order-md-2">
    <h2>Contenido</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eos animi nesciunt mollitia impedit
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ea dignissimos ullam, iure error dolc
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ea dignissimos ullam, iure error dolc
  </div>
</section>
```

## EJEMPLO GUIADO

- Además añadimos un color de fondo claro, un borde, un padding de 3.
- Aplicamos la clase order para que en dispositivos de tamaño pequeño, el aside se coloque el segundo y el contenido el primero.
- Por último, con la clase d de display, hacemos que el aside desaparezca (d-none)

```
<section class="row py-4">
  <aside class="col-12 col-md-4 bg-light p-3 border order-2 order-md-1 d-none d-sm-block">
    <h2 class="text-center">Aside</h2>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Molestias eum velit unde ipsam dicta
  </aside>

  <div class="col-12 col-md-8 order-1 order-md-2">
    <h2>Contenido</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eos animi nesciunt mollitia impedit
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ea dignissimos ullam, iure error dolc
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ea dignissimos ullam, iure error dolc
  </div>
</section>
```

## EJEMPLO GUIADO

- En la siguiente sección, también aplicamos la clase row para que todo ocupe una fila. Además, padding arriba y abajo de 4 y justificamos contenido.
- Como queremos 3 columnas separadas, repartimos cada columna con un tamaño de 3 columnas para dejar espacio entre ellas, color de fondo primary (azul), texto blanco y padding de 3.

```
<section class="row py-4 justify-content-between gy-4">
  <div class="col-md-3 bg-primary text-white p-3">
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. amet consectetur adipisicing elit.
  </div>

  <div class="col-md-3 bg-primary text-white p-3">
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. amet consectetur adipisicing elit.
  </div>

  <div class="col-md-3 bg-primary text-white p-3">
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. amet consectetur adipisicing elit.
  </div>
</section>
```

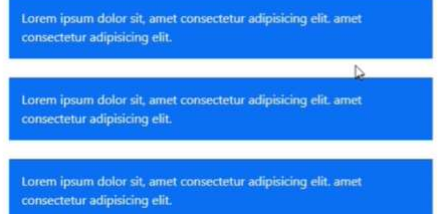
## 13. Orden de los elementos

- Aplicamos gy-4 (glutter vertical) para que se muestren en dispositivos pequeños, se vea cada uno en una fila separadas unas de otras.

```
<section class="row py-4 justify-content-between gy-4">
  <div class="col-md-3 bg-primary text-white p-3">
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. amet consectetur adipisicing elit.
  </div>

  <div class="col-md-3 bg-primary text-white p-3">
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. amet consectetur adipisicing elit.
  </div>

  <div class="col-md-3 bg-primary text-white p-3">
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. amet consectetur adipisicing elit.
  </div>
</section>
```



# 1. Bootstrap flexbox

- <https://getbootstrap.com/docs/5.3/utilities/flex/>
- Aplicamos flexbox con la clase d-flex o d-inline-flex

```
<div class="container">  
  <!-- Tipos de flexbox -->  
  <div class="d-flex p-2 bg-primary text-center my-2">Soy un flexbox container!</div>  
  <div class="d-inline-flex p-2 bg-primary text-center my-2">Soy un inline flexbox container!</div>  
</div>
```

Soy un flexbox container!

Soy un inline flexbox container!

# 1. Bootstrap flexbox

```
<!-- Direcciones en flexbox -->  
<p>Flex-row</p>  
<div class="d-flex flex-row mb-5">  
  <div>Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
</div>  
  
<p>Flex-row reverse</p>  
<div class="d-flex flex-row-reverse mb-5">  
  <div>Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
</div>  
  
<p>Flex-column</p>  
<div class="d-flex flex-column mb-5">  
  <div>Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
</div>  
  
<p>Flex-column reverse</p>  
<div class="d-flex flex-column-reverse mb-5">  
  <div>Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
</div>
```

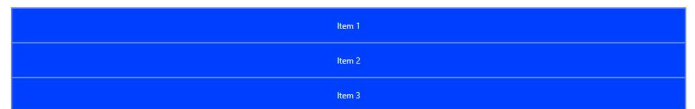
Flex-row



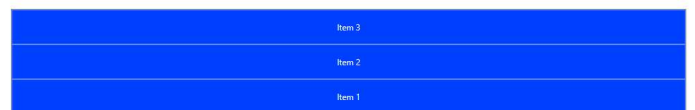
Flex-row reverse



Flex-column

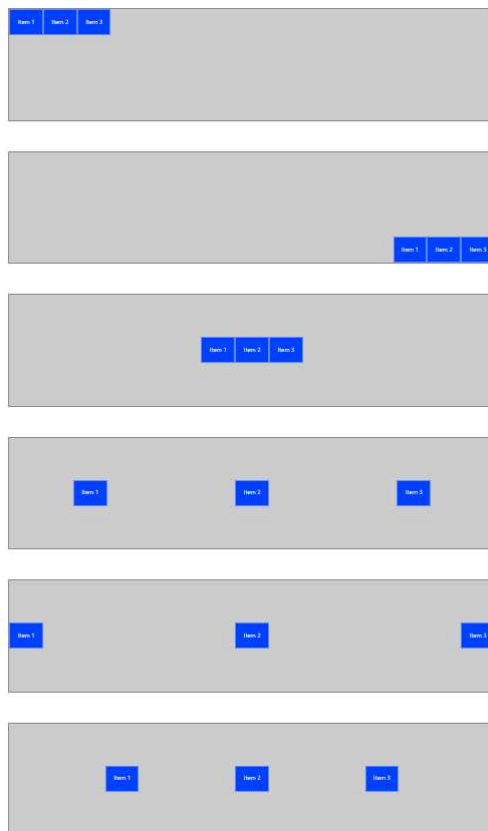


Flex-column reverse



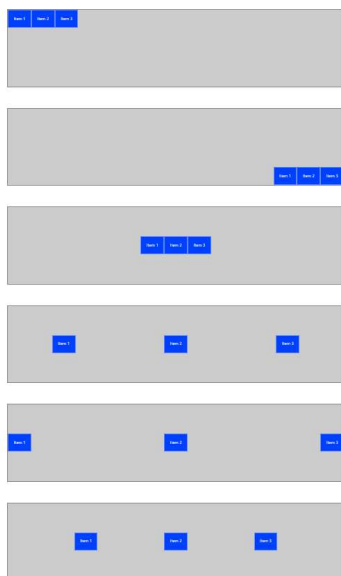
## 2. Alineación con flex

- <https://getbootstrap.com/docs/4.0/css/flex/>
- Clase justify-content



justify-content

## 2. Alineación con flexbox



```
<div class="container pt-5">

  <div class="d-flex justify-content-start align-items-start">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
  </div>

  <div class="d-flex justify-content-end align-items-end">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
  </div>

  <div class="d-flex justify-content-center align-items-center">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
  </div>

  <div class="d-flex justify-content-around align-items-center">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
  </div>

  <div class="d-flex justify-content-between align-items-center">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
  </div>

  <div class="d-flex justify-content-evenly align-items-center">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
  </div>

</div>
```



## 4. Flexbox fill, grow, shrink

- <https://getbootstrap.com/docs/5.3/utilities/flex/#fill>
- Utiliza la clase **.flex-fill** en una serie de elementos hermanos para obligarlos a tener anchos iguales a su contenido (o anchos iguales si su contenido no supera sus border-boxes) mientras ocupa todo el espacio horizontal disponible.
- Usa las utilidades **.flex-grow-\*** para alternar la capacidad de crecimiento de un elemento flexible para llenar el espacio disponible. En el ejemplo a continuación, los elementos **.flex-grow-1** usan todo el espacio disponible que pueden, mientras permiten que los dos elementos flexibles restantes tengan el espacio necesario.
- Cambia la forma en que los elementos flexibles se envuelven en un contenedor flexible. Elige entre no envolver en absoluto (el valor predeterminado del navegador) con **.flex-nowrap**, envolver con **.flex-wrap** o envolver en reversa con **.flex-wrap-reverse**.

## 4. Flexbox fill, grow, shrink

### Fill



### Grow



### Shrink



### Wrap



```
<div class="container pt-5">

  <!-- Fill -->
  <h1>Fill</h1>
  <div class="d-flex align-items-start">
    <div class="p-2 flex-fill">flex item with a lot of content</div>
    <div class="p-2 flex-fill">flex item</div>
    <div class="p-2 flex-fill">flex item</div>
  </div>

  <!-- Grow -->
  <h1>Grow</h1>
  <div class="d-flex align-items-end">
    <div class="p-2 flex-grow-0">flex item</div>
    <div class="p-2">flex item</div>
    <div class="p-2 flex-grow-1">third flex item</div>
  </div>

  <!-- shrink -->
  <h1>Shrink</h1>
  <div class="d-flex align-items-center">
    <div class="p-2 w-100">flex item</div>
    <div class="p-2 flex-shrink-1">flex item</div>
  </div>

  <!-- Wrap -->
  <h1>Wrap</h1>
  <div class="d-flex flex-wrap">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
    <div>Item 4</div>
    <div>Item 5</div>
    <div>Item 6</div>
    <div>Item 7</div>
    <div>Item 8</div>
    <div>Item 9</div>
    <div>Item 10</div>
    <div>Item 11</div>
    <div>Item 12</div>
    <div>Item 13</div>
    <div>Item 14</div>
    <div>Item 15</div>
  </div>

  <div class="d-flex flex-wrap-reverse">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
    <div>Item 4</div>
    <div>Item 5</div>
    <div>Item 6</div>
    <div>Item 7</div>
    <div>Item 8</div>
    <div>Item 9</div>
    <div>Item 10</div>
    <div>Item 11</div>
    <div>Item 12</div>
    <div>Item 13</div>
    <div>Item 14</div>
    <div>Item 15</div>
  </div>

</div>
```

## 5. Orden de los elementos con flexbox

- <https://getbootstrap.com/docs/5.3/utilities/flex/#order>



```
<div class="container pt-5">  
  
  <div class="d-flex justify-content-between">  
    <div class="w-25 order-last p-2">1</div>  
    <div class="w-25 order-first p-2">2</div>  
    <div class="w-25 order-2 p-2">3</div>  
    <div class="w-25 order-1 p-2">4</div>  
  </div>  
</div>
```