

# Bootstrap 5

Antonio Espín Herranz

Del 27/02/2023 al 16/03/2023

# Contenidos

- Introducción
- Grid
- Tipografía
- Imágenes
- Tablas
- Alertas
- Badges (Insignias)
- BreadCrumbs (Migas de pan)
- Acordeón
- Botones
- Grupo de botones
- Tarjetas
- Carrousel
- Collapse
- Formularios
- Dropdown
- Grupos de listas
- Ventanas modales
- Navs y Tabs
- NavBar
- OffCanvas
- Paginación
- Popover
- Barras de progreso
- Spinners
- Toast
- Tooltips
- Clases de Utilidad
- Scrollspy
- Glyphicons

# Introducción

# ¿Qué es Bootstrap?

- Framework de desarrollo Web
- Herramientas y elementos que ya están programados (HTML5 - CSS3 y JS)
- En Bootstrap se trabaja con etiquetas HTML y clases de CSS
- El componente principal es el grid que permite crear sitios Web adaptables a cualquier dispositivo: SmartPhone, Tablet, PC o TV
- Permite añadir nuevos estilos CSS

# Necesidades para el curso

- Navegadores FireFox y Chrome
  - Tiene soporte en todos los navegadores
- Editor de código: VSCode + plugins
  - <https://code.visualstudio.com/download>
- Acceso a la doc de Bootstrap
  - <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

# Descargar Bootstrap

- Distintas formas:
  - **Descargar**: carpetas **CSS** y **JS** (se adjuntan a nuestro proyecto).
  - **CDN**. Los archivos **CSS** y **JS** se encuentran en **jsDelivr**
  - Source files. Archivos de **Saas**. Se requieren otras herramientas y tiene sentido si vas a modificar Bootstrap.
  - Se puede instalar también con **npm**
- Utilizar las versiones comprimidas:

# La plantilla

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grid</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <!-- Debajo mis propios estilos -->
</head>
<body>
  <!-- Las etiquetas HTML -->
  <script src="js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Crear un html:

Generar la plantilla con:  
**! Tab**

También:  
**html:5**

# Extensiones VSCode

- Firefox (debugger)
- HTML y CSS
- Bootstrap 5 quick Snippets
- Utilizar atajos de teclado:
  - div.col → <div class="col"></div>
  - link:css → Genera la etiqueta css
  - script:src → Genera la etiqueta js
- **Multicursores** en Code:
  - Click en el punto de inserción en la 1ª fila
  - Control+Alt+ (flecha abajo) marcar todas las filas y escribir (cambiar todas simultáneamente)

En **Firefox**:

Menú → mas herramientas

**Vista de diseño adaptable**

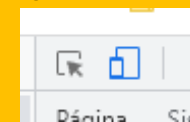
O la combinación de teclas:

**Control+shift+m**

En **Chrome**:

Menú → mas herramientas

**Herramientas para desarrolladores**





# Grid de BootStrap

# El container

- La etiqueta principal el container o container-fluid (sin margen lateral al 100%)
  - `<div class="container"></div>`
- A partir de esta etiquetas se realiza un diseño por filas y dentro de cada filas por columnas.
- Trabajamos con un sistema de **12 columnas**.


# El Grid de Bootstrap

- Podemos asignar tamaño a las columnas dependiendo del dispositivo con el que entre el usuario.
- Incluso de pueden ir cambiando el tamaño de las columnas según vamos cambiando de dispositivo.
- Hay que probar con la vista de diseño adaptable activada.

# El grid de Bootstrap

- El esqueleto del container:
- `div.container`
  - `div.row`
    - `div.col`
- Las filas las tenemos que ver como grupos de columnas.
- Las etiquetas HTML irán dentro de las etiquetas HTML
- Organización del sitio Web en filas:
- Header
- Section → main / aside
- Widgets
- footer

# Tamaños de las columnas



• .col	Extra Small	Teléfono en vertical	< 576 px
• .col-sm	Small	Teléfono H.	> 576 px < 768 px
• .col-md	Mediano	Tabletas	> 768 px < 992 px
• .col-lg	Largo	Ordenadores	> 992 px < 1200 px
• .col-xl	Extra largo	""	> 1200 px
• .col-xxl	Extra Extra Largo	TVs	> 1400 px

# Columnas

- Dentro de las etiquetas div utilizaremos las clases de las columnas:
- col-6      Tamaño (se lee: a partir del tamaño extra small hacia delante)
- col-12 col-sm-6    queremos tamaño 12 pero al llegar a small se pase a 6.
  - Al reducir la página automáticamente cambia de 12 a 6 cuando llega al tamaño sm
- Se pueden especificar muchas más medidas:
  - col-12    col-sm-6      col-md-4      col-lg-2      col-xl-12

# Columnas

- Se pueden saltar tamaños: col col-lg-6
  - Saltamos sm y md (mediano y small)

Para insertar cols en VSCode  
**.col \* 3 {Columna #}\$ Tab**



Genera:

```
<div class="col">Columna #1</div>  
<div class="col">Columna #2</div>  
<div class="col">Columna #3</div>
```

# Alineaciones

- Columnas Anidadas
  - row
    - column
      - row
        - col-12
- row columns
  - Clases: row-cols-4 (si metemos más irán a la siguiente fila)
- Alinear verticalmente las columnas
  - Clases:
    - align-items-start
    - align-items-center
    - align-items-end
  - `<div class="row align-items-start"></div>`
  - Otras clases (a nivel de columna):
    - `div class="col align-self-start" | align-self-center | align-self-end`



# Alineaciones

- Alinear cols. horizontalmente:
  - Si metemos 2 columnas de tamaño de 4 (tenemos 8 cols ocupadas y sobran otras 4) podemos hacer alineaciones horizontales de estas 2 columnas.
- **Clases:**
  - justify-content-start (a la izq)
  - justify-content-center (en el centro)
  - justify-content-right (a la der)
  - justify-content-around (deja espacios a cada lado de las 2 cols)
  - justify-content-between (deja el hueco en el centro y alinea a los extremos)
  - Justify-content-evenly (2 cols con 3 huecos equilibrados)

# Gutters

- Espaciados entre columnas:
  - Se pueden quitar o añadir
  - g-0 los quita (este es el que más se suele utilizar)
  - `<div class="row g-0">`
    - 2 columnas a col-6
  - `</div>`
- gx-5      Gutters horizontal
- gy-5      en vertical
- Luego tenemos las distintas combinaciones con **g-sm-0** (con el tamaño del dispositivo).

# Utilidades

- Ordenar columnas.
  - Para no tener que mover las etiquetas HTML.
  - order-4
  - Por ejemplo:
    - `<div class="columna col order-4 order-md-1">`
    - Diríamos que la columna por defecto va en 4 posición pero a partir del tamaño mediano pasa a primera posición.
    - OJO solo hay orden del 1 al 5.
- Otras clases:
  - order-first y order-last

# Utilidades

- offset:
  - Desplazar a la derecha las columnas.
  - offset-3 si tenemos col-3 desplaza el equivalente a 1
  - offset-sm-6: 6 de offset en extra pequeño.
- col-auto:
  - Las columnas con el tamaño ajustado que necesitan
  - 2 columnas con col-auto, por defecto alinea a la izq., pero se pueden llevar a los extremos con la clase justify-content-between row también se puede hacer añadiendo margin-end (me-auto) o margin-start (ms-auto)

# Otras etiquetas

- Se pueden ocultar elementos o hacer visibles incluso relacionando los tamaños del dispositivo
- Si queremos cambiar un párrafo según el tamaño tenemos que poner los dos. Por defecto, se verán o no y tendremos que hacer lo mismo para un tamaño específico.
  - Son las abreviaturas d-none → display:none, d-block → display:block
- Por ejemplo:
  - `<p class="d-none d-md-block">xxx</div>`
  - Por defecto, no se muestra, pero **a partir de** tamaño mediano se verá

# Tipografía

# Tipografía

- Bootstrap aporta clases y etiquetas relacionadas con la tipografía.
- Hay que tener en cuenta el contenido semántico de las etiquetas H1 .. H6. Sólo deberían utilizarse para los títulos importantes de la página.
- Los banners promocionales de la propia página podrían tener ese mismo tamaño.
  - Tenemos las clases: display-1 a display-6

# Tipografía

- Para párrafos tenemos la clase **lead** con letra más grande y suave.
- Se pueden añadir citas (***cite***) como hacemos en los libros:
  - Ejemplo:

```
<div class="col">
  <figure class="text-center">
    <blockquote class="blockquote">
      <p>Lorem ipsum, dolor sit amet</p>
    </blockquote>
    <figcaption class="blockquote-footer">
      Lorem <b cite title="título de la fuente">ipsum</cite> ipsum, dolor sit amet
    </figcaption>
  </figure>
</div>
```

Para añadir un párrafo de relleno: Lorem (control+espacio).  
Si lo queremos de 5 palabras: lorem5 y así sucesivamente.



# Tipografía

- Bootstrap también aporta etiquetas con estilos ya configurados sobre etiquetas en vez de utilizar clases.
  - **mark**: marcar el texto
  - **del / s**: Para tachado.
  - **ins / u**: Insertar nuevo texto
  - **strong**: negrita
  - **em**: itálica

# Tipografía

- Clases para estilos de texto:
  - **fw-bold**: Font-weight
  - **fw-normal**: Font-weight
  - **fw-light**: Font-weight
  - **fst-italic**: Font-style
  - **fst-normal**: Font-style

# Tipografía

- Alinear texto y se puede configurar de acuerdo al tamaño del dispositivo → **diseño responsive**

- **text-start** (izquierda), **text-center** (centro) y **text-end** (derecha)

- Se pueden combinar:

`<p class="text-start text-sm-end">Alineado en dis.pequeños en adelante</p>`

`<p class="text-start text-md-end">Alineado en dis.med en adelante</p>`

`<p class="text-start text-lg-end">Alineado en dis.largos en adelante</p>`

`<p class="text-start text-xl-end">Alineado en dis.extra largos en adelante</p>`

# Tipografía

- Subrayar elementos con clases:
  - **text-decoration-underline**: Subrayado
  - **text-decoration-line-through**: tachado
  - **text-decoration-none**: No subrayado, por ejemplo un enlace
- Transformar a mayúsculas / minúsculas / letra capital (clases):
  - **text-lowercase**, **text-uppercase**, **text-capitalize**
- Distintos tamaños del texto de los párrafos: **fs-1** hasta **fs-6**
  - Para crear párrafos en VSCode:
    - **p\*6{texto \$} tabulador**

# Tipografía

- Colores del texto:

```
<div class="col-6">  
  <p class="text-primary">primary</p>  
  <p class="text-secondary">secondary</p>  
  <p class="text-success">success</p>  
  <p class="text-danger">danger</p>  
  <p class="text-warning">warning</p>  
  <p class="text-dark">dark</p>  
  <p class="text-body">body</p>  
  <p class="text-muted">muted</p>  
</div>
```

## Colores de Fondo:

bg-dark

## Texto transparente 50% y otros colores:

text-info          bg-dark

Text-warning

Text-White

Text-light

Text-black-50

Text-White-50

# Imágenes

# Imágenes

- Cuando incrustamos imágenes nos ocurre lo mismo, se añaden con un tamaño.
- Por ejemplo, si son col-6. La imagen puede superar este tamaño, pero se puede ajustar al tamaño reservado de las columnas con la clase **img-fluid** a la etiqueta **img**.
- Si añadimos imágenes muy grandes se pueden reducir con la etiqueta **width** de HTML o con CSS.

# Imágenes

- Las imágenes pueden flotar a la izquierda o a la derecha. Añadir las clases **float-start** (izq) y **float-end** (der).
- Para añadir un borde decorativo: clase **img-thumbnail**
- También se puede redondear las esquinas de la imagen con la clase: **rounded**
- Se puede poner un título o texto descriptivo por debajo de la imagen. Hay que añadir la etiqueta figure de HTML y además la clase figure con la **img** dentro.
- Debajo de la **figure** se añade **figcaption** y la clase **figure-caption**. El texto se añade en un **párrafo** con la clase **text-center**.



# Ejemplo

```
<div class="row">
  <div class="col-12">
    <figure class="figure">
      
      <figcaption class="figure-caption">
        <p class="text-center">Lorem ipsum dolor sit amet.</p>
      </figcaption>
    </figure>
  </div>
</div>
```

# Ejemplo

```
<div class="row">  
  <div class="col-12">  
      
      
  </div>  
</div>
```

# Tablas

# Tablas HTML

- Table
  - Caption título /caption
  - Thead
    - Tr
      - Th nombreCol /Th
    - /Tr
  - Tbody
    - Tr
      - Td valor col /Td
    - /Tr
  - /Tbody
- /Table

# Clases de Bootstrap

- A la tabla se le coloca la clase: **table**
- Colorear las filas impares: clase **table-striped** a nivel de table
- Iluminar al pasar el ratón: clase **table-hover** a nivel de fila
- Compactar la tabla: **table-sm**
- Se pueden aplicar colores a 3 niveles: tabla, fila o celda

# Colores

- table-primary
- table-secondary
- table-success
- table-danger
- table-warning
- table-dark
- table-light

# Tablas

- Las tablas se pueden hacer responsive.
- Cuando sea un dispositivo más pequeño se puede colocar el scroll sólo en la tabla en vez de toda la página.
- Hay que encerrar la tabla en una etiqueta div con la clase: **table-responsive** y el scroll sólo se aplica la tabla.

```
<div class="table-responsive">  
  <table class="table table-striped table-hover table-sm">  
    <caption>Lista de Amigos</caption>  
    <thead>  
      ....
```

# Tablas más clases

- **table-active** en un td. Marca una celda como activa.
- **border-danger**: Colorear el borde
- **table-borderless**: Quita todos los bordes
- **table-bordered**: Agrega los bordes. Se pueden aplicar a una fila.



# Alertas

## Encabezado



**Aviso** Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor, ipsam.

# Alertas

- Aviso que queremos mostrar en una Web.
- No confundir con ventanas popup.
- Se añade un div con clase alert.
  - Se pueden añadir colores: alert-primary, alert-secondary ... los de siempre con alert-color
  - Añadir un atributo **role="alert"**
    - Este atributo es importante para dispositivos de lectura.
- Dentro se colocan los componentes.

# Ejemplo

```
<div class="alert alert-warning mt-3" role="alert">  
  <strong>Aviso</strong> Lorem ipsum...  
  <a href="#" class="alert-link"></a>  
</div>
```

- Los enlaces dentro del alert se colocan con la clase alert-link
- **mt-3** → margin-top de tamaño em 3

# Alert con botón cerrar

- Se puede añadir un botón para cerrar la alerta.
- Dentro de la alert utilizar: encabezados, separadores: hr y un botón para cerrar.
- Se pueden añadir animaciones para cerrar la alerta.

# Ejemplo

```
<div class="alert alert-success mt-3 alert-dismissible fade show" role="alert">  
  <h3>Encabezado</h3>  
  <hr>  
  <strong>Aviso</strong> Lorem ipsum dolor ....  
  <a href="#" class="alert-link"></a>  
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Cerrar"></button>  
</div>
```

- Las etiquetas **aria** favorecen a las herramientas de lectura para personas discapacitadas.

# Badges

Insignias

---

Encabezado H1 **Etiqueta**

Encabezado H2 **Etiqueta**

Encabezado H3 **Etiqueta**

Encabezado H4 **Etiqueta**

Encabezado H5 **Etiqueta**

Encabezado H6 **Etiqueta**

Prueba de insignia en span

Prueba de insignia en p

# Badges

- Para resaltar ofertas dentro de una web (sin etiquetas)
- Se pueden añadir al lado de un encabezado y luego para marcar una oferta se puede hacer con el **badge**.
- Se pueden poner dentro de una etiqueta **span** con la clase: **badge**
  - `<h1>Encabezado H1 <span class="badge bg-primary">Etiqueta</span></h1>`
- Se pueden añadir colores: **bg-primary**, **bg-secondary**, **bg-success**, etc.

# Badges

- Se pueden redondear las esquinas con la clase: **rounded-pill**
- Se añade dentro de la misma etiqueta span.
- Con el color **bg-light** si tenemos un fondo blanco no se verá el texto, añadir **text-dark**.
- La clase **badge** se puede añadir a otros elementos como etiquetas span sin encabezado o un párrafo.



# BreabCrumbs

Migas de pan

[Inicio](#) / [Productos](#) / [Tablets](#) / IPads

# BreadCrumbs

- Marcan la ubicación de estamos navegando.
- Si hemos ido entrando por distintas secciones de la web nos indican la ruta seguida.
- Se añade sobre una etiqueta **nav** de HTML.
- Dentro de la clase **nav** se añade la clase **breadcrumb**, cualquier separación que queramos añadir utilizaremos la clase **mt-3**.
  - Dentro de **nav** se añaden enlaces con la clase: **breadcrumb-item**
  - Podemos marcar el enlace activo con la clase **active**.
- En el último enlace que tengamos será en la sección donde estamos situados se puede sustituir por

# Ejemplo

```
<nav class="breadcrumb mt-3">  
  <a href="#" class="breadcrumb-item">Inicio</a>  
  <a href="#" class="breadcrumb-item">Productos</a>  
  <a href="#" class="breadcrumb-item">Tablets</a>  
  <span class="breadcrumb-item active">IPads</span>  
</nav>
```

- En el ponemos una etiqueta **span** porque no queremos un enlace, nos dejaría en el mismo sitio que estamos.

# Acordeón

Capítulo 1



Capítulo 2



Lorem ipsum dolor sit amet consectetur adipisicing elit. Reprehenderit est, ab saepe quidem soluta labore illo ipsum animi sequi optio eius facere libero. Quidem explicabo obcaecati architecto mollitia aperiam totam?

Capítulo 3



# Acordeón

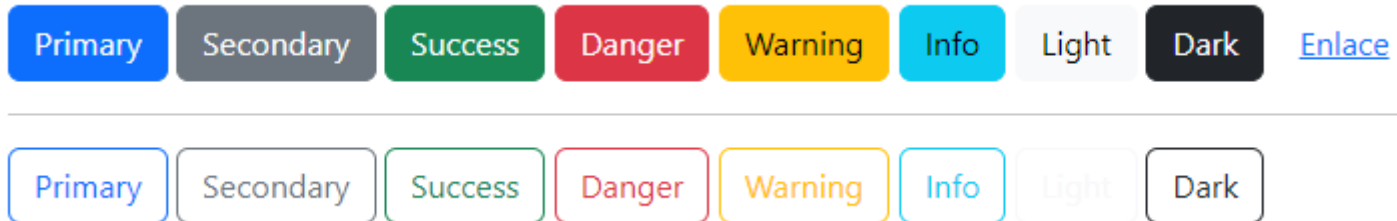
- Para este componente es **imprescindible** tener vinculado el archivo de **js**: “**js/Bootstrap.bundle.min.js**” al final del **body**.
- Para añadir el acordeón añadimos:
  - Container
    - Row
      - Col
        - Una div con la clase: `accordion` y un atributo `id`
- Cada elemento del acordeón consta de un encabezado y un cuerpo. Necesito un contenedor que tiene dentro la cabecera y el cuerpo.
  - Esto se repite por cada elemento que tengamos dentro del acordeón.

# Ejemplo de un elemento

```
<div class="accordion accordion-flush" id="temario">
<div class="accordion-item">
  <h2 class="accordion-header" id="encabezado-1">
    <button
      class="accordion-button"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#elemento-1"
      aria-expanded="true"
      aria-controls="elemento-1"
    >Capítulo 1</button>
  </h2>
  <div
    id="elemento-1"
    class="accordion-collapse collapse show"
    aria-labelledby="encabezado-1"
    data-bs-parent="#temario"
  >
    <div class="accordion-body">
      <p>Lorem ipsum dolor ...</p>
    </div>
  </div>
</div>
</div>
```

La clase: **accordion-flush**  
Quita los bordes.

# Botones



# Botones

- Utilizar la etiqueta **button**, con el atributo **type=button** (por cuestiones de renderizado).
- `button type="button" class="btn btn-primary">`
- Se puede utilizar como colores:
  - btn-secondary
  - btn-success
  - btn-warning
  - btn-danger
  - btn-info
  - btn-light
  - btn-dark



# Botones

- El botón puede tener una apariencia de enlace si le ponemos como clase: **btn-link**
- El botón se puede dejar con el contenido vacío con las clases:
  - **btn btn-outline-primary**
  - primary lo podemos sustituir por cualquier color de los que tiene Bootstrap
- Este tipo de botón (**btn-outline-color**) tienen un comportamiento distinto entre Chrome y Firefox. Al aproximar el ratón no se rellena el hueco (en Firefox)

# Botones

- Los botones también se puede deshabilitar con la clase **disabled**.
  - `<button type="button" class="btn disabled btn-primary">Primary</button>`
- Hay otros elementos que se puede visualizar como botones:
  - `input type="button", ... input type="submit", párrafos ...`
  - `<input type="button" class="btn btn-secondary" value="boton-input">`
  - `<input type="submit" class="btn btn-secondary" value="boton-submit">`
  - `<p class="btn btn-primary">Prueba p</p>`

# Botones

- Los botones pueden tener distintos tamaños.
  - Grandes: clase **btn-lg**
  - Pequeños: clase: **btn-sm**
- Los botones se pueden expandir a todo el tamaño que tengan.
  - Para ello se encierran los botones en un div y se añaden las clases:
  - **d-grid** para expandir los botones al tamaño de la columna.
    - Si está la **col-12** → toda la fila, **col-6** → la mitad de la fila.
  - **gap-2** Separación entre los botones.
  - **col-12** (según el número de cols ...)

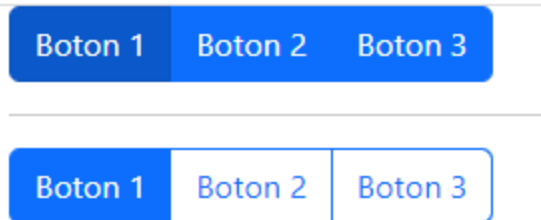
# Botones

- Los botones se pueden distribuir de otras formas para ello podemos utilizar flex:
- Cambiamos d-grid por **d-flex**, mantenemos **d-gap** y utilizamos clases: **justify-content-XXX**
- Donde XXX puede ser:
  - between
  - center
  - around

# Ejemplo

```
<div class="col-12">  
  <hr class="mt-3">  
  <div class="d-flex gap-2 justify-content-around">  
    <button type="button" class="btn btn-primary">Primario</button>  
    <button type="button" class="btn btn-secondary">Secundario</button>  
  </div>  
</div>
```

# Grupo de botones



# Grupos de botones

- Nos va a permitir agrupar botones.
- Partimos de varios botones:
  - `<button type="button" class="btn btn-primary">Botón 1</button>`
  - `""`      `""`      `""`
- Se agrupan estos botones en una capa con la clase "btn-group" y añadir el role="group" (este atributo es importante para los dispositivos de lectura).
  - También viene bien el atributo aria-label="Grupo de botones" (para personas discapacitadas lo puedan escuchar).

# Grupos de botones

- Se puede añadir la clase: **btn-outline-primary** (para solo marcar el borde del botón).
- En los grupos de botones se puede marcar uno de los botones como activo, para ello tenemos la clase “**active**”.
- Los grupos de botones pueden ser grandes o pequeños:
  - Clases:
    - **.btn-group-lg**
    - **.btn-group-sm**



# Grupos de botones: Checkbox

- Para dejar los botones marcados al hacer click tendríamos que indicar que son botones de tipo checkbox.
  - Con la clase btn-check
  - `Input type="checkbox" name="checkbox1" id="checkbox1" class="btn-check"`
  - Se añade también una etiqueta label for haciendo referencia al id del checkbox
  - `Label for="checkbox1" class="btn btn-primary">Checkbox1`
  - Con los checkbox se podrán pulsar varios.

# Grupos de botones: RadioButtons

- Si queremos que los botones sean excluyentes, tienen que ser radio-buttons y todos tienen que tener el mismo nombre para sólo poder pulsar uno de ellos.
- `<div class="btn-group" role="group" aria-label="Grupo de botones">`
- `<input type="radio" name="radio" id="radio-1" class="btn-check">`
- `<label for="radio-1" class="btn btn-primary">Radio 1</label>`

# Toolbar

- Podemos crear barras de botones y agrupar varios grupos de botones dentro de la misma toolbar.
- Bueno cuando tenemos una aplicación que tiene muchos botones.

```
<div class="btn-toolbar" role="toolbar" aria-label="Toolbar">  
  <div class="btn-group me-2" role="group" aria-label="Grupo de botones">  
    ....  
  </div>  
  <div class="btn-group" role="group" aria-label="Grupo de botones">  
    ....  
  </div>  
</div>
```

- me-2      margin-end (separación a la derecha)

# Alineación vertical

- Los grupos de botones, también se pueden alinear verticalmente:
  - Se utiliza la clase: **btn-group-vertical**
  - En este caso para separar verticalmente utilizar **mt-2** o **mb-2**
    - **margin-top** o **margin-bottom**

# Tarjetas



## Título

Subtítulo

Lorem ipsum dolor sit amet consectetur adipiscing elit. Neque obcaecati tempora voluptates blanditiis repudiandae dolore sunt iusto natus dignissimos possimus!

Botón

# Tarjetas

- Añadir container → row → col
- La tarjeta se representa por un div con la clase **card**.
  - Dentro de esta añadimos h1, p, etc.
- Para mejorar el aspecto (separación)
  - Añadir dentro de la **card** otra div con la clase **card-body**.
- Las tarjetas se adaptan al tamaño del contenedor donde se encuentran.

# Clases

- Relación de clases de la card:
- Se indica con tabulaciones que clases tienen que estar anidadas:
- **.card**: La tarjeta
  - **.card-body**: Contenedor del contenido de la tarjeta
    - **.card-title**: El título de la tarjeta.
    - **.card-subtitle**: El subtítulo de la tarjeta
    - **.card-text**: El texto de la tarjeta
    - **.card-link**: En enlace como botón en la card
  - **.card-img-top**: Imagen para la card en la parte superior
  - **.card-img-bottom**: Imagen de la card para la parte inferior

# Clases

- Al subtítulo se le suele añadir la clase **text-muted** para cambiar el color y si queremos separar este del texto de abajo añadir la clase **mb-2** (margin-bottom 2)
- Margen en el eje y: Equivale en top y bottom: **my-2**
- **mb-0** Quita el margen bottom
- Se puede añadir botones dentro de la **card**.
  - Un enlace como botón. Clase **btn btn-outline-primary**
- Las imágenes se añaden arriba o abajo, pero siempre fuera de cuerpo de la tarjeta.



# Clases

- Dentro de la tarjetas también se pueden añadir listas.
- Una lista desordenada, utilizar las clases:
  - `ul.list-group`
    - `li.list-group-item`
  - La lista ya agrega los bordes por defecto, pero se pueden quitar:
    - **`list-group-flush`**
- También podemos agregar encabezados y pies de página.

# Mas clases

- A las card se les puede añadir un header y footer.
- .card-header y .card-footer
  - Van por encima y debajo respectivamente del div.card-body
- Se pueden añadir citas dentro de las card con blockquote y footer.

# Ejemplo

```
<div class="col-6">  
  <div class="card">  
    <div class="card-header">cita</div>  
    <div class="card-body">  
      <blockquote class="blockquote">  
        <p>Lorem ipsum dolor sit, laudantium?</p>  
        <footer class="blockquote-footer">Lorem, ipsum dolor.</footer>  
      </blockquote>  
    </div>  
  </div>  
</div>
```

# Notas

- Se puede centrar todos los elementos que están dentro del card.
- Si ponemos la clase text-center con la clase card (afecta a todos los elementos que están dentro).
- `<div class="card text-center">...</div>`

# Mas sobre las tarjetas

- Partimos de una card típica con un encabezado, cuerpo (con título, texto y un botón).
- Con la clases:
  - .card
  - .card-header / .card-title
  - .card-body
    - h5 / p / button



# Tarjetas con pestañas

- Se puede añadir un menú de navegación dentro del header de la card.
  - Una forma de hacerlo es con una lista desordenada: ul
  - Dentro de elemento card-header
    - Se añade ul con las clases: nav **nav-tabs** y **card-header-tabs**
    - Dentro los li serán de la clase: **nav-item**
      - Con un enlace: **nav-link** (al primero se le suele poner con la clase “**active**”)
      - Para dotarles de funcionalidad haría falta javascript.

# Otro tipo de menú de pestañas

- Dentro del **card-header** la lista que representa el menú de pestañas se pueden cambiar las clases de:
  - `nav nav-tabs card-header-tabs` por ... `nav nav-pills card-header-pills`
  - La pestaña activa ahora se rellena de color.

# Personalizar el color

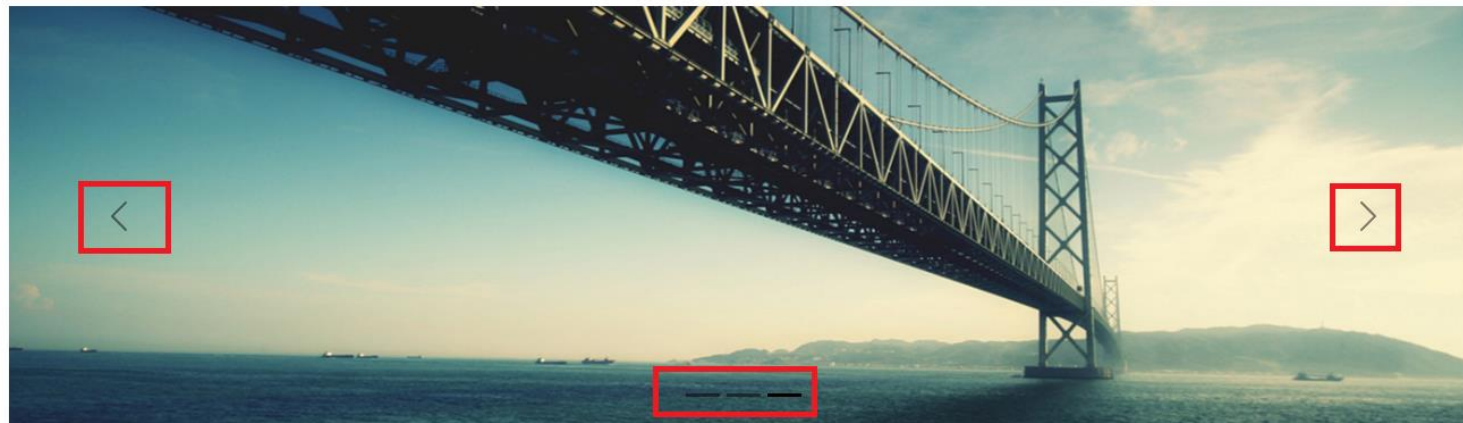
- Podemos personalizar el color de las card, a nivel de toda la tarjeta, el header o el body.
- Utilizar las clases **bg-color** (con los colores de siempre) y si metemos colores oscuros el texto se puede realzar con las clases: **text-white**, etc.
- Se puede quitar el color o manteniendo el color , personalizar el color del borde de la tarjeta: clase **border-primary**, ... con distintos colores.



# Agrupar cards

- Las cards se puede agrupar. Añadir varias cards dentro de la misma columna:
- La clase que me permite agrupar varias cards → **card-group**
  - Al encerrar las cards dentro del grupo se colocan en horizontal, todas en la misma fila.

# Carrousel



# El carousel

- El carousel de imágenes se añade dentro de una columna.
- Este componente es responsive y se ajusta al tamaño de la pantalla.
- A partir de un div se le añaden las clases: **carousel** y **slide**.
- Además tiene que tener los atributos:
  - **id**, que luego se utilizará.
  - **data-bs-ride** = “carousel”
- De forma opcional se puede añadir el atributo: **data-bs-touch=“false”**
  - Si lo ponemos a false no se permite a los usuarios de dispositivos móviles que cambien de imagen arrastrando con el dedo.

# El carousel

- Dentro del carousel se define otro contenedor para almacenar los carousel-item.
  - Este contenedor será de la clase: **carousel-inner**
  - Y cada imagen irá contenida dentro de: **carousel-item**
  - Importante señalar cual es el item activo, si no, no se verá ninguna imagen.
    - Se marca con la clase **active**.
  - Dentro de cada item se añade la imagen.
    - Para que se extienda a todo el contenedor se añade la clase: **img-fluid**

# Componentes: Controles

- Se pueden añadir botones encima de las imágenes para que el usuario pueda pasar de una imagen a otra.
- Los botones se colocan dentro del carousel pero fuera del componente `carousel-item`.
  - `<button class="carousel-control-prev" → indicar next para el otro botón`  
`type="button"`  
`data-bs-target="#mi-carousel" → es el id del carousel`  
`data-bs-slide="prev" → next para el otro botón`  
`>`
  - Dentro del botón (con etiquetas **span** se indica el icono).
    - Clase: **carousel-control-prev-icon** (para el otro botón será next)
    - Y el atributo **aria-hidden="true"** → para personas con discapacidad
      - Para personas con discapacidad verían el siguiente texto en vez del botón.
        - `span class="visually-hidden">Anterior`

# Componentes: Indicadores

- A parte de los botones se pueden añadir unos indicadores encima de cada imagen, para saber en que imagen estamos y además poder pinchar para cambiarla.
- Hay que indicar tantos botones como imágenes tengamos para poder hacer la transiciones de una, a otra.
  - Hay que marcar cual es el que está activo con la clase active.
  - `div class=carousel-indicators`
    - Dentro el `button`  
`type="button"`  
`class="active"`  
`data-bs-target="#mi-carousel"` → El id del carousel  
`data-bs-slide-to="0"` → Cuál es el slide que muestra.  
`aria-label="Slide 1"` → Etiquetas para discapacitados.
- De forma similar se van indicando el resto de botones (ojo, uno por cada imagen).

# Temas y transiciones

- Al carousel se le puede cambiar el tema a un color dark.
- Esto es conveniente si hemos añadido algún texto a alguna de las imágenes. Si la imagen es colores blanco puede que el texto no se vea muy bien, con el tema dark cambian los botones y el color del texto a un color oscuro. Clase: **carousel-dark**
- La transición de una imagen a otro por defecto entra por la derecha la nueva imagen, pero se puede hacer con un efecto de fade: Clase: **carousel-fade**.

# Collapse

Información

## Título

Lorem ipsum dolor sit amet.



# Collapse

- Este componente nos permite hacer efecto de colapsar o expandir un contenido según pulsemos a un **botón** o a un **enlace**.
- Se puede hacer a partir del **button** o con la etiqueta **a**.
- El elemento que queramos mostrar es libre, simplemente tiene que estar encerrado dentro de **div** con la clase **collapse** y tiene que tener un **id**.

# El collapse

- El elemento collapse por ejemplo puede contener una card.

```
<div class="collapse mt-3" id="contenido-2">  
  <div class="card">  
    <div class="card-body">  
      <h2 class="card-title">Título</h2>  
      <p>Lorem ipsum dolor sit amet.</p>  
    </div>  
  </div>  
</div>
```

# Collapse con el enlace

<a href="#**contenido-1**" → en href hace referencia al id del collapse  
type="button"  
class="btn btn-primary"  
**data-bs-toggle="collapse"**  
aria-expanded="false"  
aria-controls="contenido-1"  
>Información</a>

# Collapse con el button

```
<button  
    data-bs-target="#contenido-1"  
    type="button"  
    class="btn btn-primary"  
    data-bs-toggle="collapse"  
    aria-expanded="false"  
    aria-controls="contenido-1"  
>Información</button>
```

# Formularios

Correo electrónico

Te enviaremos un mensaje de confirmación

Mensaje

Enviar

# Controles de los formularios

- Input
- TextArea
- Select
- Check box
- Radio Button

# Input

- Como cada componente de Bootstrap el formulario se colocará en:
- Un container
  - Una fila
    - Una columna
- Utilizaremos etiquetas: form, label e input
  - Para las etiquetas clases: **form-label**
  - Para los controles del form: **form-control**
  - Texto adicional que queramos añadir de ayuda en un div
    - Clase **form-text**

# Ejemplo

```
<form action="">
```

```
  <div class="mb-3">
```

```
    <label for="correo" class="form-label">Correo electrónico</label>
```

```
    <input type="text" class="form-control" id="correo" aria-describedby="ayuda-correo">
```

```
    <div id="ayuda-correo" class="form-text">
```

Te enviaremos un mensaje de confirmación

```
    </div>
```

```
  </div>
```

```
</form>
```



# Input

- Los controles input se puede hacer mas grandes o mas pequeños, utilizando las clases: **form-control-lg** y **form-control-sm** estas clases irán a parte de la clase **form-control**.
- Se pueden utilizar también los atributos de HTML (**readonly** y **disabled** y **placeholder**)
- Se pueden crear input con texto plano con la clase:
  - **form-control-plaintext**

# Ejemplo

```
<form action="">  
  <input type="text" class="form-control form-control-lg mb-3"  
    placeholder="Input grande">  
  <input type="text" class="form-control form-control-sm mb-3"  
    placeholder="Input pequeño">  
  <input type="text" class="form-control mb-3"  
    placeholder="Input Deshabilitado" disabled>  
  <input type="text" class="form-control mb-3"  
    placeholder="Input Solo lectura" readonly>  
  <input type="text" class="form-control-plaintext mb-3"  
    placeholder="Input plain text" readonly>  
  <input type="file" class="form-control" name="" id="">  
  
</form>
```

# TextArea

```
<div class="mb-3">  
  <label for="mensaje" class="form-label">Mensaje</label>  
  <textarea class="form-control" name="mensaje" id="mensaje"></textarea>  
</div>
```

El botón de envío del form:

- button type="submit" con las clases btn btn-color (el color que sea: primary, secondary, etc.)

# Select

- La etiqueta select de HTML se complementa con la clase: **form-select** de Bootstrap.
- Es conveniente poner la etiqueta **aria-label** para los lectores de páginas web.
- La clase form-select se puede complementar con las clases:
  - **form-select-lg** Tamaño más grande
  - **form-select-sm** Tamaño más pequeño

# Ejemplo

```
<select name="pais" id="pais" class="form-select mb-3" aria-label="Selección de país">  
  <option value="mexico">México</option>  
  <option value="españa">España</option>  
  <option value="colombia">Colombia</option>  
</select>
```

- También se puede utilizar el atributo **disabled** de la etiqueta **select**.

# Select

- Para crear listas con selección múltiple. Añadir el atributo **multiple** de HTML y **size** para indicar el tamaño.
  - `<select multiple size="3" name="pais" id="pais" class="form-select mb-3" aria-label="Selección de país">`
  - `<option value="mexico">México</option>`
  - `<option value="españa">España</option>`
  - `<option value="colombia">Colombia</option>`
  - `<option value="argentina">Argentina</option>`
  - `</select>`

# CheckBox

- Cada checkbox se encierra en un **div** con la clase **form-check**.
- El checkbox con una etiqueta input type="checkbox" y se añade la clase:
  - **form-check-input**
- El label asociado al checkbox tiene que ser de la clase:
  - **form-check-label**

```
<div class="form-check">  
  <input type="checkbox" name="checkbox1" id="checkbox1" class="form-check-input">  
  <label for="checkbox1" class="form-check-label">Checkbox1</label>  
</div>
```

# RadioButton

```
<div class="form-check">
```

```
  <input type="radio" name="grupo" id="radio1" class="form-check-input">
```

```
  <label for="radio1" class="form-check-label">Radio 1</label>
```

```
</div>
```

```
<div class="form-check">
```

```
  <input type="radio" name="grupo" id="radio2" class="form-check-input">
```

```
  <label for="radio2" class="form-check-label">Radio 2</label>
```

```
</div>
```



# Switch

- Nueva etiqueta para simular los checkbox de los dispositivos móviles.

```
<div class="form-check form-switch">  
    <input type="checkbox" name="switch-1" id="switch-1" class="form-check-input">  
    <label for="switch-1" class="form-check-label">Switch 1</label>  
</div>
```

```
<div class="form-check form-switch">  
    <input type="checkbox" name="switch-2" id="switch-2" class="form-check-input">  
    <label for="switch-2" class="form-check-label">Switch 2</label>  
</div>
```

# Checkbox Inline

- Para los checkbox inline disponemos de la clase: **form-check-inline**
- Cada checkbox se coloca en un **div**

```
<div class="form-check form-check-inline">
```

```
  <input type="checkbox" name="checkbox-inline1" id="checkbox-inline1" class="form-check-input">
```

```
  <label for="checkbox-inline1" class="form-check-label">checkbox-inline1</label>
```

```
</div>
```

- Se pueden poner también con estilo de botón. En este caso no se colocan los checkbox dentro de una capa.
- Se coloca la etiqueta input y la etiqueta label for.

# Checkbox y RadioButtons como botones

- Sin capa a diferencia de como ocurre con los checkbox
- Si los queremos a estilo check para marcar varios (ponemos nombre diferente)  

```
<input type="checkbox" name="checkbox1" id="checkboxb1" class="btn-check">  
<label for="checkboxb1" class="btn btn-primary">Checkbox1</label>
```
- Si los queremos que sean excluyentes se pone el mismo nombre en el atributo name.  

```
<input type="radio" name="radio-button" id="radiob1" class="btn-check">  
<label for="radiob1" class="btn btn-primary">radio1</label>  
<input type="radio" name="radio-button" id="radiob2" class="btn-check">  
<label for="radiob2" class="btn btn-primary">radio2</label>
```

# Inputs de Rango

- El control de rango de HTML5 también se le pueden dar estilos de Bootstrap.
- `<input type="range" class="form-range" id="rango" min="0" max="1000" step="10" >`
- La idea de poner un id es para poner la etiqueta label for con clase **form-label**.
- Además se puede poner un texto en cada extremo del rango.
  - Añadir debajo del rango dos etiquetas.
  - Añadir un **div** con la clase: **d-flex** y **justify-content-between** (para alinear cada uno a los extremos)
    - 2 párrafos con la clase form-text
- También se puede añadir el atributo **disabled** al control range

# Grupos de Inputs

- Nos permite agrupar input text, select, textarea con etiquetas.
- Disponemos de la clase **input-group**

Nombre

@gmail.com

Notas:

Nombre

Apellido

Pequeño

Grande

País

México

▼

Enviar

# Grupos de inputs

```
<div class="input-group mb-3">  
  <input type="text" class="form-control" placeholder="Nombre" aria-label="Nombre">  
  <span class="input-group-text">@gmail.com</span>  
</div>
```

```
<div class="input-group mb-3">  
  <span class="input-group-text">Notas:</span>  
  <textarea name="" id="label-notas" class="form-control" aria-label="Notas"></textarea>  
</div>
```

- La etiqueta span se puede colocar por encima o por debajo según queramos colocar la etiqueta a la izquierda o a la derecha.

# Grupos de inputs

- En el caso del select se le puede añadir un label:

```
<div class="input-group">  
  <label for="país" class="input-group-text">País</label>  
  <select name="pais" id="pais" class="form-select">  
    <option value="mexico">México</option>  
    <option value="España">España</option>  
    <option value="colombia">Colombia</option>  
  </select>  
  <button class="btn btn-primary">Enviar</button>  
</div>
```

# Etiquetas Flotantes

Nombre

Notas

País  
Selecciona el país

Nombre

Notas

País  
Selecciona el país

- Al pinchar en la etiqueta, el literal se desplaza hacia arriba para teclear en la parte de abajo.
- Se puede colocar en un input **text**, **textarea** y **select**.
- Clases: **form-floating** a un div:
  - Dentro de esta capa se añade uno de los 3 controles y por debajo un label for con el id al control anterior.



# Ejemplo

```
<div class="form-floating mb-3">  
  <textarea id="notas" name="notas" class="form-control" placeholder="Notas"></textarea>  
  <label for="notas">Notas</label>  
</div>
```

```
<div class="form-floating mb-3">  
  <select id="pais" name="pais" class="form-select">  
    <option selected>Selecciona el país</option>  
    <option value="mexico">México</option>  
    <option value="España">España</option>  
    <option value="colombia">Colombia</option>  
  </select>  
  <label for="pais">País</label>  
</div>
```

# Formularios Responsive

- Los formularios se engloban dentro de un col:
  - div.container
    - div.row
      - div.col
        - form
- Los controles que están dentro del formulario hay que organizarlos en filas y columnas.
- Si lo queremos hacer responsive es importante añadir el margen siempre a las columnas. Tener en cuenta que al reducir la pantalla, seguramente se va a colocar una sola columna ocupando toda la fila y habrá que separarla de la columna que se coloque debajo. Utilizar mb-3 (margin-bottom)

# Formularios Responsive

- **form**

- `div class="col-12 col-md-6 mb-3"`
  - Por defecto la columna ocupa las 12 columnas de ancho, al ampliar la pantalla a tamaño medio (tabletas) se coloca en tamaño 6 (a partir del tamaño medio:lg, xl, xxl, etc.)
- Cada control del form debería llevar su etiqueta.
  - Rellenar los atributos **id** y **name** de cada control.
- Clases:
  - Para las etiquetas: **form-label**
  - Para los controles: **form-control**
  - Los radiobutton en un div cada uno con la clase **form-check**
    - Dentro de la clase:
      - `input type="radio"` con la clase: **form-check-input**
      - La etiqueta: **form-check-label**

# Formularios Responsive

- Para la parte de aceptar los términos, el check que colocamos al final con el botón de enviar el form, se pueden ajustar al mismo tamaño y centrar utilizando **flexbox**.
- Se colocan en una nueva fila, una col (d-flex y justify-content-center)
  - **div.row**
    - **div.col** class="col-12 **d-flex justify-content-center**
      - **div**: La capa extra para agrupar el checkbox y el botón

# Ejemplo

```
<div class="row">
  <div class="col-12 d-flex justify-content-center">
    <div>
      <div class="form-check mb-3">
        <input type="checkbox" class="form-check-input" name="terminos" id="terminos">
        <label for="terminos" class="form-check-label">Acepto los términos y condiciones</label>
      </div>    <!-- Para que abarque lo mismo que el check -->
      <button type="submit" class="col-12 btn btn-primary">Enviar</button>
    </div>
  </div>
</div>
```

# Clases de validación para formularios

- En los campos del form se pueden añadir clases para indicar que un valor es correcto o incorrecto.
  - Incluso con un texto aclaratorio debajo del input:
  - Clases: **is-valid** / **is-invalid**
  - Para los texto informativos: **valid-feedback** / **invalid-feedback**

Nombre

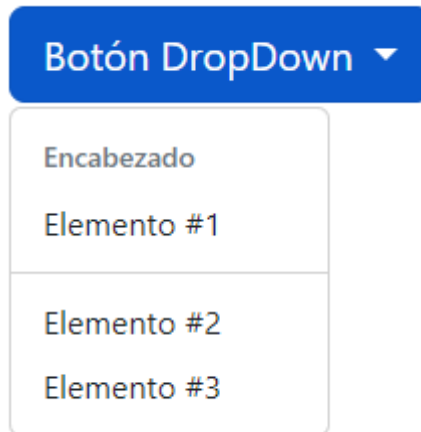
Correcto!

Correo

Introduzca un correo válido

# Menú de navegación (Dropdown)



# DropDown

- Distintas posibilidades de dropdown. Botón que despliega un menú.
- Se puede cambiar la dirección hacia donde se despliega el menú:
  - Dropdown: Hacia abajo
  - Dropup: Hacia arriba
  - DropStart: a la izquierda
  - DropEnd: a la derecha
- El menú se puede hacer oscuro.
- Se puede hacer con un botón o con un enlace
- Se puede poner un botón adicional para separa el botón del otro botón con la flecha.
- Puede hacerse grande o pequeño
- Se pueden añadir encabezado o divisores a la parte del menú
- Si queremos tener varios dropdown juntos utilizar grupos de botones.



# Dropdown con botón

```
<div class="dropdown">  
  <button  
    class="btn btn-primary dropdown-toggle"  
    type="button"  
    id="dropdown-1"  
    data-bs-toggle="dropdown"  
    aria-expanded="false"  
  >Botón DropDown</button>
```

# Dropdown con enlace

```
<div class="dropdown">  
  <a  
    href="#"  
    class="btn btn-primary dropdown-toggle"  
    type="button"  
    id="dropdown-2"  
    data-bs-toggle="dropdown"  
    aria-expanded="false"  
  >Enlace DropDown</a>
```

# La parte del menú

- El menú se implementa con una lista desordenada (ul) y cada li se añade un enlace con href="#"

```
<ul class="dropdown-menu" aria-labelledby="dropdown-1">  
  <li><a href="#" class="dropdown-item">Elemento #1</a></li>  
  <li><a href="#" class="dropdown-item">Elemento #2</a></li>  
  <li><a href="#" class="dropdown-item">Elemento #3</a></li>  
</ul>
```

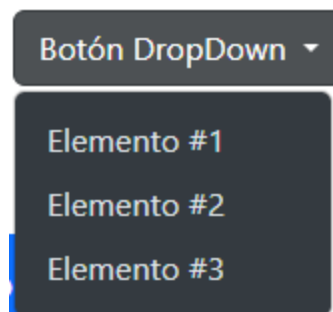
- Las etiquetas aria como siempre para que los lectores puedan leer el contenido.

# DropDown Split

- Se puede separar el botón con la flecha de otro botón con la etiqueta del menú.
- Para agrupar los dos botones necesitamos una capa con la clase: **btn-group**
- Se añade un primer botón: **button** con las clases **btn** y **btn-color** (con el color que queramos) y la etiqueta que da nombre al menú.
  - `<Button class="btn btn-...">Menú</Button>`
- El siguiente botón será el que muestra la flecha. A parte del contenido ya indicado en el dropdown con botón, tenemos que añadir la clase: **dropdown-toggle-Split**
- El contenido de este botón será vacío, salvo una etiqueta span.
  - `<span class="visually-hidden">Mostrar / Ocultar menú</span>`

# DropDown Dark

- Podemos tener el botón con color **dark** y además el contenido del menú.
- Partiendo del dropdown normal le añadimos a la clase del botón: **btn-dark** y a la lista que representa el menú se añade la clase: **dropdown-menu-dark**



# DropDown large / DropDown small

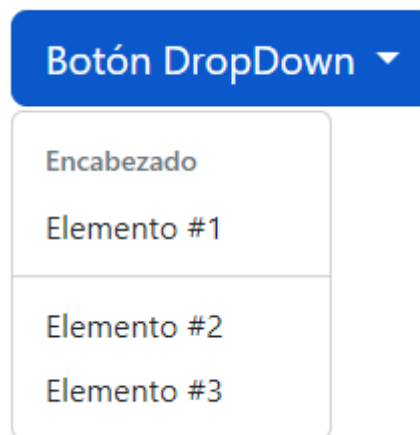
- Igual que otros controles el dropdown proporciona estilos para hacerlo más pequeño o más grande.
- Para ello si partimos del dropdown inicial en la clase del botón se añade la clase:
  - **btn-lg** Para tamaño grande
  - **btn-sm** Para pequeño

# DropDown otras direcciones

- Se puede abrir el menú hacia arriba (**up**), izquierda (**start**) o hacia la derecha (**end**)
- Disponemos de la clases (para colocar en la capa que agrupa el botón y el menú (representado por la lista (**ul**))):
  - **dropup / dropstart / dropend**
- Dentro del menú, en la etiqueta ul el atributo **aria-labelledby** se añade: **dropup, dropstart o dropend** respectivamente.

# El menú con encabezados y divisores

- En la parte del menú se pueden añadir otros componentes aparte de los enlaces.
- Podemos colocar un encabezado como la primera opción para mostrar un título **<h5>**, **<h6>** ...
- Las opciones del menú se puede separar mediante divisores (**hr**)





# Ejemplo

```
<ul class="dropdown-menu" aria-labelledby="dropdown">  
  <li><h6 class="dropdown-header">Encabezado</h6></li>  
  <li><a href="" class="dropdown-item">Elemento #1</a></li>  
  <li><hr class="dropdown-divider"></li>  
  <li><a href="" class="dropdown-item">Elemento #2</a></li>  
  <li><a href="" class="dropdown-item">Elemento #3</a></li>  
</ul>
```

# Grupos de Listas

Elem 1	11
Elem 2	22
Elem 3	33
<input type="checkbox"/> CheckBox1	
<input type="checkbox"/> CheckBox2	

**Titulo de mi artículo** Hace 1 día

Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia, accusantium.

[Ver mas](#)

**Titulo de mi artículo** Hace 1 día

Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia, accusantium.

[Ver mas](#)

**Titulo de mi artículo** Hace 1 día

Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia, accusantium.

[Ver mas](#)

# Grupos de listas

- Bootstrap también tiene estilos para las **listas**.
- A la etiqueta **ul** se le añade la clase: **list-group**
- Cada item va con la clase: **list-group-item**
- Se pueden completar con las clase **active** (para marcar el item que está activo).
- O para deshabilitar un elemento, la clase: **disabled**

# Ejemplo

```
<ul class="list-group">  
  <li class="list-group-item">Elemento#1</li>  
  <li class="list-group-item active">Elemento#2</li>  
  <li class="list-group-item disabled">Elemento#3</li>  
  <li class="list-group-item">Elemento#4</li>  
  <li class="list-group-item">Elemento#5</li>  
</ul>
```

•

# Listas con enlaces

- Se crea la etiqueta **ul** con la clase: **list-group**
- En vez de añadir elementos **li**, se añaden enlaces directamente.
- Los enlaces deben de llevar las clases:
  - **list-group-item**
  - **list-group-item-action**

# Lista sin bordes

- Por defecto, la lista siempre añade **bordes**.
- Se pueden quitar añadiendo la clase: **list-group-flush**
- `ul class="list-group list-group-flush"`
- Dentro de la lista se mantienen las misma etiquetas que antes.

# Lista numerada

- La lista aunque sea ul se puede añadir una clase para numerar los elementos.
- `<ul class="list-group list-group-numbered">`
- Los elementos mantienen las mismas etiquetas / clases que antes.

# Listas con colores

- A los ítems de la lista li aparte de la clase: **list-group-item** se pueden añadir estas clases (los colores son los de siempre).
  - list-group-item-primary
  - list-group-item-secondary
  - list-group-item-success
  - list-group-item-danger
  - list-group-item-warning
  - list-group-item-info
  - list-group-item-light
  - list-group-item-dark
- Si estamos con enlaces hay que mantener la clase: **list-group-item-action**



# Listas con insignias (badge)

- La insignia se añade de un elemento **li** en una etiqueta **span**
- Para alinear el texto a la izquierda y el badge a la derecha se utiliza **flexbox**
  - `<ul class="list-group">`
  - `<li class="list-group-item d-flex justify-content-between">`
  - Elem 1
  - `<span class="badge bg-primary align-items-center">11</span></li>`
  - `</ul>`

# Lista con checkbox

- También se pueden añadir **checkbox** dentro de los elementos de la lista.
- En este caso en vez de utilizar elementos li se añade una label.
  - `<label class="list-group-item">`
  - `<input type="checkbox" name="check1" id="check1" class="form-check-input me-1">`
  - `CheckBox1`
  - `</label>`
- me-1: margin-end (a la derecha)

# Lista personalizada

```
<ul class="list-group">
  <a href="#" class="list-group-item list-group-item-action">
    <div class="d-flex justify-content-between">
      <h5>Titulo de mi artículo</h5>
      <small>Hace 1 día</small>
    </div>
    <p class="mb-1">Lorem ipsum dolor</p>
    <small class="text-muted">Ver mas</small>
  </a>
</ul>
```

<b>Titulo de mi artículo</b> Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia, accusantium. Ver mas	Hace 1 día
<b>Titulo de mi artículo</b> Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia, accusantium. Ver mas	Hace 1 día
<b>Titulo de mi artículo</b> Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia, accusantium. Ver mas	Hace 1 día

# Ventanas Modales



# Ventanas modales

- Las ventanas modales las podemos abrir a partir de un botón.
- Se añade un **button** con las clases **btn**, **btn-color** ... y necesitamos añadir los atributos:
  - **data-bs-toggle = “modal”** y **data-bs-target=“#id\_de\_modal”**
- Para construir la **ventana modal**:
  - Una **capa** con la clase **modal** y puede tener también **fade** (para mostrarse progresivamente).
  - Tiene que tener el **id** (que coincida con el atributo **data-bs-target** del button)
  - **tabindex=-1** y luego atributos **aria-hidden=true** y **aria-labelledby=“label-modal-1”**

# Ventanas modales

- Estructura principal de div:
  - **div.modal**
    - **div.modal-dialog**
      - **div.modal-content**
        - **div.modal-header**
          - Encabezado con la clase: **modal-title**
          - Button para cerrar la ventana con atributos:
            - **data-bs-dismiss="modal"**
            - **aria-label="cerrar"**
        - **div.modal-body**
          - Contenido del cuerpo.
        - **div.modal-footer**
          - Normalmente con botones:
            - Button type=button class="btn" **data-bs-dismiss="modal"** → para cerrar
            - Button type=button class="btn" → Para otros cometidos

# Ventanas modales

- A la estructura anterior se puede completar:
  - Haciendo la ventana modal
  - Y centrando en el navegador.
- Por defecto la ventana se cierra al pinchar en cualquier sitio, se puede convertir en modal con:
  - **div.modal data-bs-backdrop="static"**
    - Se puede centrar: **div.modal-dialog modal-dialog-centered**

# Ventanas modales

- Se puede implementar una ventana con un **scroll** en caso de que el texto que contenga sea muy amplio.
- Para simularlo: encerramos el texto que va dentro de **modal-body** en una capa con height:1000px.
- En la caja del dialogo se puede añadir la clase:
  - **modal-dialog-scrollable**
  - El scroll solo afecta al texto, la parte del título y los botones queda fija.



# Nav y Tabs

[Inicio](#) [Blog](#) [Proximamente](#) [Contacto](#)

# Navs y Tabs

- Los **navs** se utilizan para menús de navegación, pueden ser **horizontales** y **verticales**.
- Los horizontales se pueden centrar a la izq (**start**), centro y derecha (**end**).
- Se puede deshabilitar algún enlace con la clase: **disable**
- Se construyen a partir de una lista **ul** y dentro de los **ítems** hay que colocar **enlaces**.
  - Los enlaces irán con la clase: **nav-link**

# Ejemplo

- Centrado a la izquierda:

```
<ul class="nav">  
  <li><a href="#" class="nav-link">Inicio</a></li>  
  <li><a href="#" class="nav-link">Blog</a></li>  
  <li><a href="#" class="nav-link disabled" tabindex="-1" aria-disabled="true">Proximamente</a></li>  
  <li><a href="#" class="nav-link">Contacto</a></li>  
</ul>
```

- Para **centrar** el **nav** añadir la clase: **justify-content-center** a la etiqueta ul
- Para alinear a la **derecha**: **justify-content-end**

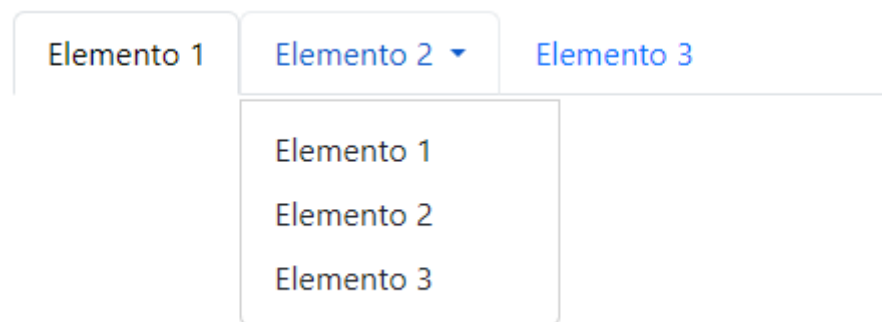
# Alineación vertical

- Añadimos la etiqueta **nav** de **HTML5** con enlaces dentro con la clase **nav-link**.
  - Y a la etiqueta **nav** se añade la clase: **nav** y **flex-column**.

```
<nav class="nav flex-column">  
  <a href="#" class="nav-link">Inicio</a>  
  <a href="#" class="nav-link">Blog</a>  
  <a href="#" class="nav-link">Proximamente</a>  
  <a href="#" class="nav-link">Contacto</a>  
</nav>
```

# Pestañas

- Dentro de una pestaña se puede añadir un dropdown como se indica en la foto de abajo.
- Se añaden clases para indicar cual es la pestaña activa.
- Se pueden añadir estilos para rellenar el contenido de la pestaña.



# Pestañas

- Se construye a partir de una lista **ul** con la clase **nav** (como antes) pero se añade la clase: **nav-tabs**.
- Dentro de la lista los ítems **li** tienen que tener la clase: **nav-item** y los **enlaces** con la clase: **nav-link**.
- Se puede indicar la pestaña activa con la clase **active**.

# Ejemplo

```
<ul class="nav nav-tabs">  
  <li class="nav-item"><a href="#" class="nav-link active" aria-current="page">Elemento  
  1</a></li>  
  <li class="nav-item"><a href="#" class="nav-link">Elemento 2</a></li>  
  <li class="nav-item"><a href="#" class="nav-link">Elemento 3</a></li>  
</ul>
```

- Se puede rellenar el contenido de la pestaña sustituyendo la clase **nav-tabs** con la clase: **nav-pills**.
- O rellenando todo el ancho, para ello se añade además de las dos anteriores la clase: **nav-fill**.

# Pestañas con dropdown

- A cualquiera de los ítems que se mantienen en la lista se le puede añadir un componente dropdown para tener un menú desplegable:

```
<li class="nav-item dropdown">  
  <a href="#" class="nav-link dropdown-toggle"  
    data-bs-toggle="dropdown" role="button"  
    aria-expanded="false">  
    Elemento 2  
  </a>  
  <ul class="dropdown-menu">  
    <li><a href="#" class="dropdown-item">Elemento 1</a></li>  
    <li><a href="#" class="dropdown-item">Elemento 2</a></li>  
    <li><a href="#" class="dropdown-item">Elemento 3</a></li>  
  </ul>  
</li>
```



# NavBar



# NavBar: Menú página principal

- **Navbar** lo vamos a utilizar para implementar el menú principal de nuestra web.
- Se puede combinar con **dropdown** para añadir opciones.
- Se puede crear un diseño responsive para dispositivos móviles.
- El menú se podrá ocultar o mostrar con un botón.
  - Se puede mostrar el menú horizontal y al ajustar a un dispositivo mas pequeño se coloque en vertical.

# NavBar

- En el caso del **navbar** se suele colocar como primer elemento.
- Colocando el container dentro del nav.
  - Podemos elegir con poner el container-fluid para extender a toda la página.
  - O no poner nada y el menú se colocará a la izquierda.
- Las clases que colocamos a la etiqueta nav
  - **navbar**
  - **nav-bar-expand-lg**: Expande el menú a partir del tamaño large en adelante y oculta el botón del menú.
    - Si se reduce se muestra un botón para el menú.
  - Se puede añadir clases para el estilo, por ejemplo: **navbar-dark** y **bg-dark**

# Estructura de etiquetas

- nav
  - div.container
    - a href class=**navbar-brand**> Título de la aplicación
    - **button** (para el menú del móvil)
    - Se añade el menú de navegación:
    - div.**collapse navbar-collapse**
      - ul.**navbar-nav** (el menú con una lista)
        - Dentro del menú elementos **li** clase: **nav-item**
        - Alguno de los **li** puede tener un **dropdown**
    - **form**: se puede añadir con una caja para buscar y un botón

# Otros estilos

- Se puede poner fija la barra de navegación, de tal forma que si tenemos mucho contenido con scroll vertical no desaparezca la **navbar**.
  - Añadir la clase: **fixed-top** a la clase **nav**.
  - Se puede añadir el contenido debajo del nav con la clase `div.container`.
  - Para simular el scroll, en la etiqueta `body` añadir una etiqueta `style`:
    - `min-height:2000px;padding-top: 56px;`
- Otra opción es colocar la **navbar** en la parte **inferior** con la clase: **fixed-bottom**.
  - Si tenemos un **dropdown** hay que cambiarlo para que se despliegue hacia arriba con la clase: **dropup**

# OffCanvas

titulo



Lorem ipsum dolor sit amet consectetur, adipisicing elit. Cupiditate, blanditiis.

ipsum dolor.

ipsum dolor sit amet consectetur adipisicing elit. Sit, error?

# OffCanvas

- Este nuevo componente nos permite mostrar un panel con contenido cuando hacemos click en un botón o en un enlace.
- El panel se puede mostrar en las 4 direcciones:
  - offcanvas-start: izq.
  - offcanvas-end: der.
  - offcanvas-top: arriba
  - offcanvas-bottom: abajo

# Offcanvas

- Necesitamos el botón / enlace desde donde haremos click para abrir el panel lateral.
- Hay que indicar el id del panel (offcanvas) para poderlo abrir.

<a

```
href="#mi-offcanvas"  
class="btn btn-primary"  
data-bs-toggle="offcanvas"  
role="button"  
aria-controls="mi-offcanvas"  
>
```

Botón

</a>

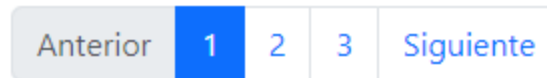


# Estructura

- div.offcanvas
  - div.offcanvas-header
    - h5.offcanvas-title
    - button.btn-close test-reset
  - div.offcanvas-body

```
<div class="offcanvas offcanvas-bottom" tabindex="-1"  
  id="mi-offcanvas" aria-labelledby="mi-offcanvas"  
  data-bs-scroll="true"  
  data-bs-backdrop="true">  
  <div class="offcanvas-header">  
    <h5 class="offcanvas-title">titulo</h5>  
    <button  
      type="button"  
      class="btn-close text-reset"  
      data-bs-dismiss="offcanvas"  
      aria-label="cerrar"  
    ></button>  
  </div>  
  <div class="offcanvas-body">  
    <p>Lorem ipsum dolor sit amet</p>  
  </div>  
</div>
```

# Paginación



# Paginación

- Mostrar una serie de números con enlaces para navegar a una determinada página.
- Se pueden mostrar también los botones de anterior y siguiente.
- Marcar como activa la página en la que nos encontramos.
- O deshabilitar, para no poder pinchar en la página activa

# Paginación

- Para la paginación se utiliza la etiqueta **nav** con una lista desordenada **ul** con la clase **pagination**.
- Dentro de cada item **li** de la lista **ul** hay que indicar la clase **page-item** y añadir enlaces con la clase: **page-link**
- Cada enlace muestra el número de la página donde se encuentra y además se puede añadir otro enlace para el anterior y el siguiente.

# Ejemplo

```
<nav aria-label="paginación de la página">
```

```
  <ul class="pagination">
```

```
    <li class="page-item disabled" aria-disabled="true" tabindex="-1"><a href="#" class="page-link">Anterior</a></li>
```

```
    <li class="page-item active"><span class="page-link" aria-current="page">1</span></li>
```

```
    <li class="page-item"><a href="#" class="page-link">2</a></li>
```

```
    <li class="page-item"><a href="#" class="page-link">3</a></li>
```

```
    <li class="page-item"><a href="#" class="page-link">Siguiente</a></li>
```

```
  </ul>
```

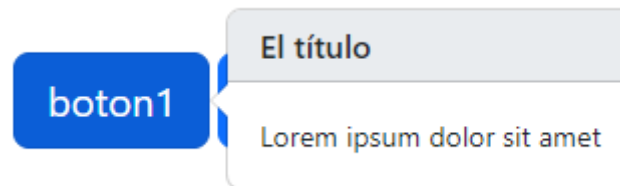
```
</nav>
```

- Las etiquetas aria son necesarias para los lectores de páginas para personas con discapacidades visuales.
- Podemos utilizar las clases: **disabled** y **active** para la página activa y si esta es la primera. El enlace al anterior se debe deshabilitar.
- Anular tabindex para no poder utilizar el tabulador.

# Paginación

- Los enlaces de anterior y siguiente se pueden sustituir por los símbolos: << y >>
- Para ello en el texto del enlace se colocan las entidades: **&laquo;** y **&raquo;**; respectivamente.
- Para colocar el control de paginación más grande utilizamos:
  - pagination **pagination-lg**
- Para más pequeño:
  - pagination **pagination-sm**
- Para centrar el control, añadir la clase: **justify-content-center**
- Para alinear a la derecha: **justify-content-end**

# Popover



# Popover

- Se activan al hacer click sobre el botón.
- Se pueden implementar también con un enlace.
- Hay que inicializarlos mediante **javascript**, si no, no funcionan.

```
<button
  class="btn btn-lg btn-primary"
  data-bs-toggle="popover"
  title="El título"
  data-bs-content="Lorem ipsum dolor sit amet"
  >
  boton1
</button>
```



# Popover

- Por defecto el **popover** se muestra a la derecha del botón.
- El atributo: **data-bs-placement** indica la posición donde se muestra respecto al botón o el enlace que representa el popover:
  - top, right, left, bottom

# Popover

- Código **javascript**.
  - Se coloca en el body debajo de la referencia del fichero de javascript de Bootstrap.

```
<script>
```

```
    var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'))
```

```
    var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {
```

```
        return new bootstrap.Popover(popoverTriggerEl)
```

```
    })
```

```
</script>
```

# Barras de Progreso



# Barras de Progreso

- Se pueden crear fijas o animadas o con un color rayado.
- Necesitamos dos capas con las clases:
  - `div.progress`
    - `div.progress-bar`
- Se indica el atributo **role="progressbar"** y además es necesario indicar el valor actual de progreso, el mínimo y el máximo.
  - Para ello tenemos los atributos:
    - `aria-valuenow`, `aria-valuemin` y `aria-valuemax`
- Se suele utilizar un estilo personalizado `style` con un `width` para indicar el ancho de la barra, debería coincidir con el valor del atributo `aria-valuenow`:
  - `style="width:25%"`
  - También se le puede dar un `height` para indicar la altura de la barra, normalmente en pixels.

# Ejemplo


```
<div class="progress mb-4">  
  <div  
    class="progress-bar"  
    role="progressbar"  
    aria-valuenow="25"  
    aria-valuemin="0"  
    aria-valuemax="100"  
    style="width:25%"  
  >  
    25%  
  </div>  
</div>
```

# Barra de progreso

- Se puede cambiar el color de la barra de progreso con las clases:
  - **bg-success**, **bg-dark**, etc. **bg-color**
- Para indicar una barra con el color rayado y/o animada utilizar las clases (en la capa interior):
  - **progress-bar** **progress-bar-striped** **progress-bar-animated**

# Spinners



 Cargando ...

 Cargando ...

# Spinners

- Los **spinner** los vamos a utilizar para indicar que la página está cargando.
- La vista habitual los  $\frac{3}{4}$  de círculo girando o se pueden incrustar en un botón.
- Además se puede implementar con un círculo que crece.



# Spinners

```
<div class="spinner-border" role="status">  
    <span class="visually-hidden">Cargando ...</span>  
</div>
```

- Es necesario indicar el atributo **role** con el valor status.
- La etiqueta span con la clase: **visually-hidden** es necesario para los lectores de las páginas.

# Spinners

- Se puede definir un spinner pequeño o grande con las clases:
  - spinner-border **spinner-border-sm** → Para pequeño
  - spinner-border **spinner-border-lg** → Para grande
- Además se puede personalizar con el atributo style:
  - **style**="width:4rem;height:4rem"
- El **color** se puede cambiar añadiendo las clases:
  - **text-primary**, **text-dark**, **text-color** junto con la clase spinner-border.

# Spinner en botones

- Para indicar el spinner dentro de un botón se añade una etiqueta **span**.
- Se puede poner el botón en **disabled** para no poder pinchar mientras que estamos esperando la carga.

```
<button class="btn btn-primary" type="button">  
  <span class="spinner-border spinner-border-sm" role="status"></span>  
    Cargando ...  
</button>
```

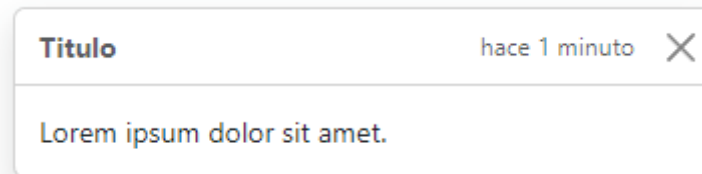
# Spinner grow

- Otro tipo de spinner con un círculo relleno que va creciendo:

```
<div class="spinner-grow text-primary" role="status">  
    <span class="visually-hidden">Cargando ...</span>  
</div>
```

- Además como el anterior se puede personalizar el tamaño.
  - Utilizar el atributo **style** con los valores **height** y **width** con el número.

# Notificaciones (toast)



# Toast

- Dentro de la página vamos a poder mostrar **pequeñas notificaciones**.
- Normalmente se muestran a la **derecha** y en la parte **superior**.
- En este caso no se utiliza el contenedor habitual de Bootstrap se utiliza: **toast-container** y **position-absolute** (se suele poner un padding: p-3)
  - Dentro de este va otro contenedor:
    - **div.toast show** → Tiene que llevar la clase show para que se muestre.
    - Después se reparte el contenido en:
      - **toast-header** Para la cabecera, normalmente se añade un botón para cerrar.
      - **toast-body** Para el body

# Ejemplo

```
<div class="toast-container position-absolute p-3" style="right: 0; top:0;">
  <div class="toast show" role="alert" aria-live="assertive" aria-atomic="true">
    <div class="toast-header">
      <strong class="me-auto">Titulo</strong>
      <small>hace 1 minuto</small>
      <button type="button" class="btn-close" aria-label="cerrar"></button>
    </div>
    <div class="toast-body">
      Lorem ipsum dolor sit amet.
    </div>
  </div>
```

- El toast tiene que tener el role de alert → es como si fuera una alerta.
- Las etiquetas **aria** como siempre, son necesarias para **los lectores de las páginas**.

# Toast

- El **toast** se puede implementar solo con el cuerpo:

```
<div class="toast show" role="alert" aria-live="assertive" aria-atomic="true">  
  <div class="toast-body">  
    Lorem ipsum dolor sit amet.  
  </div>  
</div>
```

- Si solo tenemos el body y además queremos añadir el botón de cerrar para colocarlos correctamente necesitamos una capa adicional que los contenga y a esta se le añade la clase **d-flex** → para que entre **flexbox**



# Ejemplo

`<!-- Con el cuerpo y el botón de cerrar -->`

```
<div class="toast show" role="alert" aria-live="assertive" aria-atomic="true">
```

```
  <div class="d-flex">
```

```
    <div class="toast-body">
```

```
      Lorem ipsum dolor sit amet.
```

```
    </div>
```

```
    <button type="button" class="btn-close me-2 m-auto" aria-label="cerrar"></button>
```

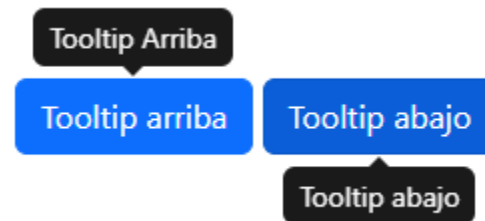
```
  </div>
```

```
</div>
```

# Toast con otros colores

- Para cambian los colores al toast tenemos que añadir más clases a la capa que añade la clase: toast
  - `div.toast show` **bg-primary text-white**
  - Como siempre `bg-color` (background - fondo)
  - `text-color` (para el texto)

# ToolTips



# Tooltips

- Pequeño componente que cuando pasemos el cursor por encima de alguna parte de la web nos muestre una pequeña caja con algo de información.
- Se puede hacer con un **botón**, con un **enlace** o con una **imagen**.
- Los **tooltips** al igual que los **popovers** hay que habilitarlos mediante **javascript**.

# Tooltips

- Código **javascript** para activar los tooltips:

```
<script>
    var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]'))
    var tooltipList = tooltipTriggerList.map(function (tooltipTriggerEl) {
        return new bootstrap.Tooltip(tooltipTriggerEl)
    })
</script>
```

# Tooltips

```
<button  
    class="btn btn-primary"  
    type="button"  
    data-bs-toggle="tooltip"  
    data-bs-placement="top"  
    title="Tooltip Arriba"  
>  
    Tooltip arriba  
</button>
```

- Con el atributo **data-bs-placement** se puede cambiar la posición donde aparece el tooltip, tomando los valores: **top**, **bottom**, **left** y **right**.

# Clases de Utilidades de Bootstrap

- Enlaces coloreados
- Ratios
- Posicionamiento
- Visually-hidden
- Stretched-links (Enlaces alargados)
- Fondos
- Bordes
- Overflow
- Sombras
- Tamaños
- Resumen espaciado

# Enlaces coloreados

- Disponemos de una serie de clases para colorear los enlaces:
  - `<a href="#" class="link-color">Texto</a>`
- Color puede ser:
  - link-primary
  - link-secondary
  - link-success
  - link-danger
  - link-warning
  - link-info
  - link-light
  - link-dark



# Ratios

- Vamos a poder definir contenedores que guardan una proporción que indicaremos.
  - Por ejemplo: 1 x 1, 2 x 3, etc.
- Son **div** donde se les añade la clase **ratio** y se pueden colocar algunos de tipo **ratio-1x1**, **ratio-4x3**, **ratio-16x9**, **ratio-21x9**
  - *Esto puede venir bien para ciertos monitores o TVs*
- Se les puede asignar un color de fondo: **bg-color** (bg-primary)
- Y un color para el texto: **text-color** (text-white)

# Ratios

- Puede venir bien para visualizar videos de you-tube y que sean responsive y que se siga manteniendo la proporción aunque se reduzca el tamaño del dispositivo.

# Posicionamiento

```
<div class="position-relative bg-secondary" style="height:500px;">  
  <div class="caja bg-dark position-absolute top-0 start-0"></div>  
  <div class="caja bg-dark position-absolute top-0 end-0"></div>  
  <div class="caja bg-dark position-absolute top-50 start-50"></div>  
  <div class="caja bg-dark position-absolute bottom-50 end-50 "></div>  
  <div class="caja bg-dark position-absolute bottom-50 start-0"></div>  
  <div class="caja bg-dark position-absolute bottom-0 end-0"></div>  
</div>
```

# Otros posicionamientos

- Se pueden posicionar contenedores:
  - Fijo arriba: clase: **fixed-top**
  - Fijo abajo: **fixed-bottom**
  - Aunque tengamos **scroll** los dos contenedores se mantienen arriba y abajo fijos.
- Class: **sticky**: Es un contenedor que se encuentra en medio de la página y cuando hacemos scroll y llega a la parte alta se mantiene.
  - Se puede quedar pegado al de arriba si le ponemos un style de top: 56px.
  - En este caso se quedaría pegado al contenedor (con la clase: fixed-top)

# visually-hidden

- Clases que permite ocultar ciertos textos para una página normal que van a poder ser leídos por los lectores de pantalla, para que personas con discapacidades puedan escuchar la página.
- **Clases:**
  - visually-hidden
  - visually-hidden-focusable

# Stretched-links

- Con esta clase que le aplicamos a un enlace nos permite extender el área sensible a una tarjeta completa.
- En vez de activarse sólo cuando pasamos el cursor por encima del botón se puede extender a toda la tarjeta.
- La clase es: **stretched-link**

# Ejemplo

```
<div class="card">  
    
  <div class="card-body">  
    <h5 class="card-title">Titulo de la card</h5>  
    <p class="card-text">Lorem ipsum dolor sit amet.</p>  
    <a href="#" class="btn btn-primary stretched-link">Ver más</a>  
  </div>  
</div>
```

# Fondos

- Disponemos de las clases: **bg-color** para poner el fondo a un contenedor. Y estos se pueden degradar con la clase: **bg-gradient**.
- Disponemos de una clase transparente: **bg-transparent**
  - `<div class="p-3 mb-2 bg-transparent text-dark">.bg-transparent</div>`
- Y los colores para el texto:
  - **text-white**
  - **text-dark**
    - `<div class="p-3 mb-2 bg-primary bg-gradient text-white">.bg-primary</div>`



# Bordes

- Disponemos de varias clases para poner los bordes a los componentes de Bootstrap.
- Incluso con esquinas redondeadas.
- Se pueden indicar colores en los bordes: **border-primary**, ... border-**color**
- Clase principal: **border** (se aplica a todos) y luego se pueden poner por zonas: **border-top**, **border-bottom**, **border-end**, **border-start**.
- Para esquinas redondeadas:
  - **rounded**
  - O por zonas: **rounded-top**, **rounded-bottom**, **rounded-end**, **rounded-start**, **rounded-circle**, **rounded-pill** (para este indicar un width: 200px, tiene que ser rectangular).

# Bordes

- A parte de los colores, tipo de esquinas, y zona donde queremos poner el borde, disponemos también del grosor del borde.
- Los grosores de los bordes van de **border-1** a **border-5** (el mas grueso).

# Overflow

- Para overflow tenemos las siguientes clases:
  - **overflow-auto**: por defecto muestra el scroll
  - **overflow-hidden**: no lo muestra y lo corta
  - **overflow-visible**: El texto si se sale de la caja, lo muestra tal cual.
  - **overflow-scroll**: Muestra las dos barras de scroll
- Estas clases se las colocaremos a un contenedor que es que contiene al texto.

# Ejemplo

```
<style>  
  .caja {  
    max-height: 100px;  
  }  
</style>
```

```
<div class="caja border border-dark border-3 p-2 overflow-auto">
```

```
  <h3>.overflow-auto</h3>
```

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Perferendis quos maxime rem necessitatibus laborum, doloribus corporis id minus quaerat aperiam debitis, eum dolore at vel numquam perspiciatis eveniet molestiae voluptatibus?

```
</div>
```

# Sombras

- Se pueden añadir sombras suavizadas.
- Cuatro posibilidades:
  - Sin sombra: **shadow-none**
  - Sombra pequeña **shadow-sm**
  - Sombra normal **shadow**
  - Sombra larga **shadow-lg**
- Ejemplo:
  - **<div class="shadow-none p-3 mb-5 bg-light">Sin sombra</div>**

# Tamaños

- Clases para controlar el tamaño de nuestros elementos.
  - Se puede indicar con un porcentaje tanto en horizontal como en vertical.
  - Para indicar el tamaño en vertical, el contenedor tiene que tener un alto que se establece con el atributo **style** y **height: XX**
- Para el ancho: **w-25, w-50,w-75,w-100**
- Para el alto: **h-25, h-50, h-75, h-100**
- **No tenemos otros valores ...**

# Ejemplos

```
<div class="col-6">  
  <div class="bg-primary text-white p-2 my-2 w-25">25%</div>  
  <div class="bg-primary text-white p-2 my-2 w-50">50%</div>  
</div>
```

```
<div class="col-6">  
  <div style="height: 200px" class="bg-dark">  
    <div class="bg-primary text-white p-2 mx-2 d-inline-block h-25">25%</div>  
    <div class="bg-primary text-white p-2 mx-2 d-inline-block h-50">50%</div>  
  </div>  
</div>
```

# Resumen Espaciado

- **Espaciado**

- p          Padding
- m          margin
  
- t          top
- b          bottom
- s          start (izquierda)
- e          end (derecha)
  
- y          eje y (arriba y abajo)
- x          eje x (izquierda y derecha)



# Ghyphicons

# Glyphicons

- Es una galería de iconos gratuita en Bootstrap:
  - <https://getbootstrap.com/docs/3.3/components/#glyphicons>
- Indicar la doble clase que se coloca para mostrar el icono.
- **No mezclar con otros componentes.**
  - Las clases de iconos no se pueden combinar directamente con otros componentes.
  - No deben usarse junto con otras clases en el mismo elemento. En su lugar, agregue un anidado `<span>` y aplique las clases de iconos al archivo `<span>`.
- **Solo para uso en elementos vacíos.**
  - Las clases de iconos solo deben usarse en elementos que no contienen contenido de texto y no tienen elementos secundarios

# Iconos accesibles

- Las versiones modernas de tecnologías de asistencia anunciarán contenido generado por CSS, así como caracteres Unicode específicos.
- Para evitar resultados imprevistos y confusos en los lectores de pantalla (particularmente cuando los íconos se usan únicamente como decoración), los ocultamos con el **aria-hidden="true"** atributo.
- Si está utilizando un ícono para transmitir significado (en lugar de solo como un elemento decorativo), asegúrese de que este significado también se transmita a las tecnologías de asistencia; por ejemplo, incluya contenido adicional, oculto visualmente con la clase **.sr-only**.
- Si está creando controles sin otro texto (como uno `<button>` que solo contiene un ícono), siempre debe proporcionar contenido alternativo para identificar el propósito del control, de modo que tenga sentido para los usuarios de tecnologías de asistencia. En este caso, podría agregar un **aria-label** atributo en el propio control.

# Ejemplo

```
<button type="button" class="btn btn-default" aria-label="Left Align">  
  <span class="glyphicon glyphicon-align-left" aria-hidden="true"></span>  
</button>
```

```
<button type="button" class="btn btn-default btn-lg">  
  <span class="glyphicon glyphicon-star" aria-hidden="true"></span> Star  
</button>
```

# Ejemplo

```
<div class="alert alert-danger" role="alert">
```

```
  <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
```

```
  <span class="sr-only">Error:</span>
```

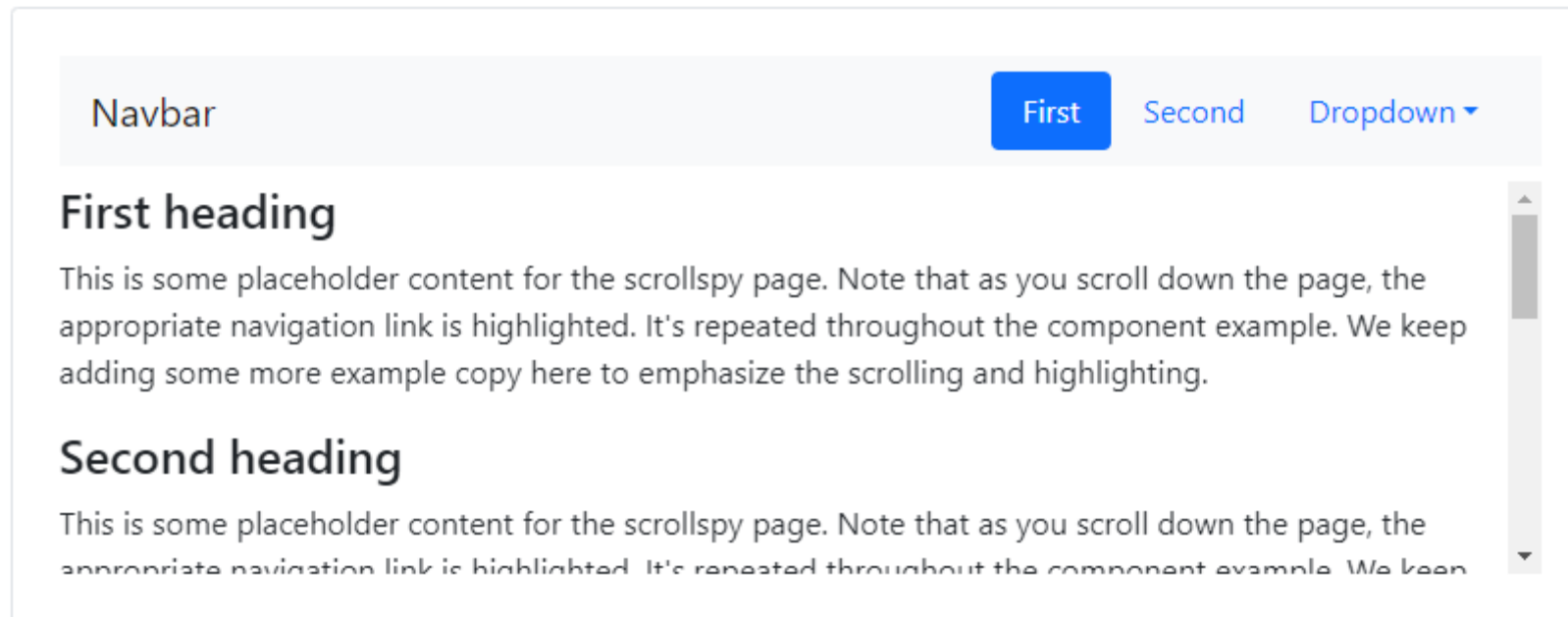
```
  Enter a valid email address
```

```
</div>
```

## Formas de trabajar con los iconos:

- <https://www.youtube.com/watch?v=gH2CImOviqY>

# Scrollby



# Scrollby

- Otro contenedor que se suele colocar sobre el **navbar** y nos permite acceder directamente a contenidos de nuestra página mediante botones de navegación.
  - Al pulsar en estos botones automáticamente navega y si nos movemos por el scroll automáticamente va activando el botón que indica la sección en la que nos encontramos.
  - Podemos colocarlo en la parte superior como el menú de navegación principal.
    - O en el lateral de la página e incluso anidar el contenido.
- <https://getbootstrap.com/docs/5.0/components/scrollspy>

# Scrollby

- Elemento principal **nav**.
  - Los botones se implementan con una lista **ul**
    - Cada item **li** hace referencia a un atributo **#id** que se especifica fuera de la nav en otro contenedor con la clase: **scrollby-example**
- También se puede mediante un **list group**



# Estructura

- ```
<!-- The scrollable area -->
<body data-spy="scroll" data-target=".navbar" data-offset="50">

<!-- The navbar - The <a> elements are used to jump to a section in the
scrollable area -->
<nav class="navbar navbar-inverse navbar-fixed-top">
  ...
  <ul class="nav navbar-nav">
    <li><a href="#section1">Section 1</a></li>
  ...
</nav>

<!-- Section 1 -->
<div id="section1">
  <h1>Section 1</h1>
  <p>Try to scroll this page and look at the navigation bar while scrolling!</p>
</div>
...
</body>
```

# Scrollby Vertical

- `<body data-spy="scroll" data-target="#myScrollspy" data-offset="20">`  
    `<div class="container">`  
        `<div class="row">`  
            `<nav class="col-sm-3" id="myScrollspy">`  
                `<ul class="nav nav-pills nav-stacked">`  
                    `<li><a href="#section1">Section 1</a></li>`  
                    `</ul>`  
                `</nav>`  
                `<div class="col-sm-9">`  
                    `<div id="section1">`  
                        `<h1>Section 1</h1>`  
                        `<p>Try to scroll this page and look at the navigation list while scrolling!</p>`  
                    `</div>`  
                `</div>`  
        `</div>`  
    `</div>`  
    `</body>`

# Visual Studio Code

# Atajos con VSCode

- Generar varios encabezados:
- `h$*6{Encabezado H$} tab`
  - Generar 6 encabezados y el \$ se va sustituyendo por 1 ... 6.