

Presentación: Tercera parte de Bootstrap

Estaremos viendo que: data-toggle es un atributo dentro de Bootstrap JS que liga el elemento al tipo de funcionamiento que muestra, por ejemplo:

```
data-toggle="modal"  
data-toggle="collapse"  
data-toggle="dropdown"  
data-toggle="tab"
```

Dropdowns: Elementos desplegables

Descripción: Esta construido en la librería Popper, por lo cual es importante meter el archivo popper.min.js antes del Bootstrap de JavaScript, o usar bootstrap.bundle.min.js el cual contiene a la librería Popper.

Los menús desplegables de Bootstrap son de uso genérico, esto quiere decir que nos permiten incluir en estos inputs y control de cuestionarios, por lo que otras características como role y atributos aria se deben incluir manualmente.

En principio funcionan haciendo clic sobre la opción de desplegar el menú y se puede navegar por los ítems con las flechas del teclado y cerrar con la tecla ESC.

Un solo botón: Un solo .btn puede convertirse en un elemento dropdown, partimos del ejemplo donde tenemos un .btn que nos indica un menú de colores.

```
<button type="button" class="btn btn-outline-info" id="Dibujar">Dibujar</button>  
<button type="button" class="btn btn-outline-info" id="Borrar">Borrar</button>  
<button type="button" class="btn btn-outline-info">Menú Colores</button>
```

Y cambia el ultimo botón a:

```
</div>  
<button type="button" class="btn btn-outline-info" id="Dibujar">Dibujar</button>  
<button type="button" class="btn btn-outline-info" id="Borrar">Borrar</button>  
<div class="dropdown">  
  <button type="button" class="btn btn-outline-info btn-secondary dropdown-toggle" id="menuColores"  
    data-bs-toggle="dropdown" aria-expanded="false">Menú Colores</button>  
  
  <ul class="dropdown-menu" aria-labelledby="menuColores">  
    <li><a class="dropdown-item" href="#">Color 1</a></li>  
    <li><a class="dropdown-item" href="#">Color 2</a></li>  
    <li><a class="dropdown-item" href="#">Color 3</a></li>  
  </ul>  
</div>  
<p id="parrafo1" class="font-monospace fw-bold fs-5"></p>
```

Si tenemos anclas: en caso de tener anclas lo que hacemos es lo siguiente:

And with `<a>` elements:

Dropdown link ▾

Copy

```
<div class="dropdown">
  <a class="btn btn-secondary dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-b:
    Dropdown link
  </a>

  <ul class="dropdown-menu" aria-labelledby="dropdownMenuLink">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>
```

Botones divididos: Son botones más completos que muestran el nombre del menú y una pequeña división que activa el despliegue. Es el que mostramos en el código.

Se pueden tener variantes:

- de color,
- tamaño,
- color negro de los items de la lista,
- incluir en dropdown en la barra de navegacion,
- direcciones de despliegue,
- dropdowns sin formato interactivo,
- Divisores en el despliegue,
- activar e inactivar items,
- especificar la localizacion del despliegue,
- especificar el cierre del despliegue.

Sobre su uso: Se puede usar por dos métodos desde los atributos en las etiquetas o desde JavaScript, pero siempre será necesario el uso de `data-bs-toggle="dropdown"` en un link o un botón.

Via data attributes

Add `data-bs-toggle="dropdown"` to a link or button to toggle a dropdown.

```
<div class="dropdown">
  <button id="dLabel" type="button" data-bs-toggle="dropdown" aria-expanded="false">
    Dropdown trigger
  </button>
  <ul class="dropdown-menu" aria-labelledby="dLabel">
    ...
  </ul>
</div>
```

Via JavaScript

Call the dropdowns via JavaScript:

```
var dropdownElementList = [].slice.call(document.querySelectorAll('.dropdown-toggle'))
var dropdownList = dropdownElementList.map(function (dropdownToggleEl) {
  return new bootstrap.Dropdown(dropdownToggleEl)
})
```

`data-bs-toggle="dropdown"` still required

Regardless of whether you call your dropdown via JavaScript or instead use the data-api, `data-bs-toggle="dropdown"` is always required to be present on the dropdown's trigger element.

Eventos: es importante decir que siempre se pueden agregar otros eventos a la lista de eventos de los botones de menú desplegables. [Ejemplo sencillo en el código](#).

```
let myDropdown = document.getElementById('myDropdown')

myDropdown.addEventListener('show.bs.dropdown', function () {
  // acción a realizar.
})
```

NavBar: Barras de navegación

Descripción: Es una cabecera de navegación responsiva que soporta complementos que colapsan, navegación, marcas (como logotipos de empresas) y más.

- Para que sea responsiva requiere del empaque `.navbar-expand{-sm | ... | -xxl}` previamente vistas. Pero por defecto son responsivas.
- Por default NavBar viene con un tenedor de clase Fluido por lo que ocupa todo el espacio horizontal de la pantalla, debemos cambiar el contenedor para ajustar.
- Conviene usar las especificaciones de espacio y Flex del Bootstrap para ajustar dentro de la NavBar.
- Usar la etiqueta explícita de `<nav>` del HTML, si se ocupa una etiqueta genérica como `div`, se debe agregar `role="navigation"`.

Contenido que soporta: A continuación vemos la descripción de la página.

- `.navbar-brand` for your company, product, or project name.
- `.navbar-nav` for a full-height and lightweight navigation (including support for dropdowns).
- `.navbar-toggler` for use with our collapse plugin and other [navigation toggling](#) behaviors.
- Flex and spacing utilities for any form controls and actions.
- `.navbar-text` for adding vertically centered strings of text.
- `.collapse.navbar-collapse` for grouping and hiding navbar contents by a parent breakpoint.
- Add an optional `.navbar-scroll` to set a `max-height` and [scroll expanded navbar content](#).

`.navbar-brand` con texto e imagen (logotipo):

```
<nav class="navbar navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">
      
      Bootstrap
    </a>
  </div>
</nav>
```

`.navbar-nav`, `.navbar-toggler`: Como se organizan los links en la barra de navegación y se hacen colapsables de manera responsiva.

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
      target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-
      label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
      <div class="navbar-nav">
        <a class="nav-link active" aria-current="page" href="#">Home</a>
        <a class="nav-link" href="#">Features</a>
        <a class="nav-link" href="#">Pricing</a>
        <a class="nav-link disabled">Disabled</a>
      </div>
    </div>
  </div>
</nav>

```

.collapse.navbar.collapse: Una navbar aún más completa pues muestra, aparte de lo anterior, una opción de menú desplegable. [En el código se muestra.](#)

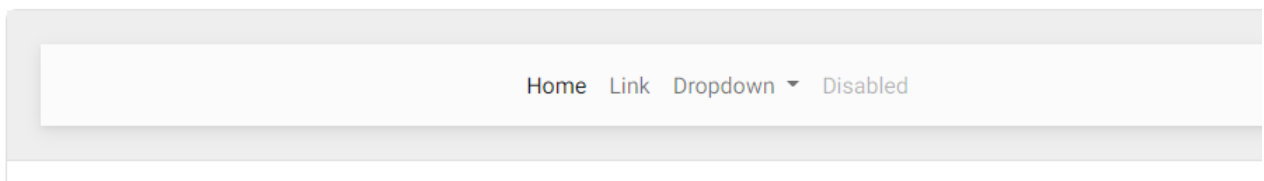
Una opción más completa sería incluir un espacio de formulario como una casilla de búsqueda.

Se pueden modificar (tener variantes):

- Colores, tamaños. [en el código se ve](#)
- Contenedores
- En donde se fija el contenedor dentro del Nav, donde posicionar texto en el Nav. Ver ejemplo [igual está en el código.](#)

Centered

Remove `.ms-auto` or `.me-auto` class and add `.justify-content-center` next to the `.navbar-collapse` to center the content.



- Scrolling cuando colapsa.
- Toggler, en qué parte colapsa la información de la barra cuando se hace botón.
- Colapsar contenido externo en un botón, de entrada aparece colapsado.
- Offcanvas (es otro elemento, información en un costado).

Modal: ventanas emergentes en la página.

Descripción: Es una ventana que se utiliza para agregar contenido, información, diálogos en una página. Se abre esta ventana SOBRE todo el contenido de la página por lo que el scroll de la página pasa a segundo plano y la ventana tendrá su propio scroll. La podemos activar con un botón. Tenemos las siguientes características:

- Por default dando clic en la parte fuera del modal este se cierra, pero se puede modificar: `data-bs-backdrop="static" data-bs-keyboard="false"`
- Solo se puede usar una ventana modal a la vez.
- El elemento modal tiene posición definida fixed, lo que debemos tomar en cuenta por si elementos en la parte superior interfieren con el modal.
- El atributo autofocus de HTML no tiene efecto en los Modal. Se debe usar JS para que los modal adquieran tales atributos.
- Lo que se comento sobre el scroll, en este caso podemos meter un scroll propio del modal al agregar `.modal-dialog-scrollable` al `.modal-dialog`. Y para centrarlo verticalmente se agrega `modal-dialog-centered` igualmente al `modal-dialog`.

Ejemplo con botón incluido: Podemos ver que se llama al modal a través del ID del div que tiene el Modal, notar como se llama la clase y notar el botón X predeterminado para cerrar btn-close. En el código usamos otro botón para disparar el modal.

```
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
  Launch demo modal
</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

Toolstips y Popovers: Los Toolstips son pequeños cuadros emergentes que aparece cuando el usuario mueve el puntero del ratón sobre un elemento. Pueden aparecer variantes sobre el lado donde se despliegan y otras características editables con CSS, ver la documentación.

Para agregarlos se pueden usar anclas <a> con las siguientes etiquetas de clase y título:

```
class="tooltip-test" title="Hola soy un tooltip"
```

Por su parte los Popovers son similares a los Toolstips; se trata de un cuadro emergente que aparece cuando el usuario hace clic en un elemento. La diferencia es que el Popover puede tener mucho más contenido. Mostraremos un ejemplo que se incluye en un botón dentro del modal, pero igual podría ir en un ancla.

[Ver ejemplo en el código](#)

IMPORTANTE: Para este caso del popover si hay más especificaciones necesarias para su uso:

- Resultan importantes estas etiquetas dentro del botón o ancla del Popover:

```
• data-bs-toggle="popover"
• title="Titulo del popover"
• data-bs-content="Contenido del popover"
```

- Como requiere la librería Popper se deben meter los siguientes archivos en la página:
[bootstrap.bundle.min.js](#) / [bootstrap.bundle.js](#)
- **Importante:** Se deben inicializar manualmente los Popovers pues por default no vienen inicializados por motivos de rendimiento. Una opción que muestran es activar todos los popovers que se tengan mediante:

```
var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'))
var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {
  return new bootstrap.Popover(popoverTriggerEl)
})

##### Para especificar características de los Popovers en un Container #####
let popover = new bootstrap.Popover(document.querySelector('.example-popover'), {
  container: 'body'
})
```

- Los valores de contenido y titulo vacíos nunca mostrarán un popover. También es importante especificar a los popover, que se meten en un contenedor, en el cuerpo para evitar problemas con grupos de botones, etc. En el recuadro anterior vemos como se puede especificar un popover dentro de un contenedor para modificar propiedades con CSS y JS.
- Puede variar hacia donde se coloca el popover, ejemplo:
[data-bs-placement="bottom"](#) Que se coloca en las etiquetas del botón que tendrá el Popover

- Notamos que los Popover se quedan desplegados hasta que se vuelve a oprimir el botón o ancla, para que se dejen de mostrar cuando se realice otra acción (clic en otro lado) se debe hacer un tratamiento con el focus trigger que implica usar anclas en lugar de botones y agregar dos etiquetas `tabindex="0"`: (por ahora funciona solo con eso).
- `data-bs-trigger="focus"`

Dismiss on next click

Use the `focus` trigger to dismiss popovers on the user's next click of a different element than the toggle element.

Specific markup required for dismiss-on-next-click

For proper cross-browser and cross-platform behavior, you must use the `<a>` tag, *not* the `<button>` tag, and you also must include a `tabindex` attribute.

Dismissible popover

```
<a tabindex="0" class="btn btn-lg btn-danger" role="button" data-bs-toggle="popover" data-bs-trigger="focus" data-b...
```

```
var popover = new bootstrap.Popover(document.querySelector('.popover-dismiss'), {
  trigger: 'focus'
})
```

- Para elementos inactivados igualmente se realiza un tratamiento que viene en la página. Y como siempre es posible modificar estilos, métodos, eventos y opciones.

List Groups

Descripción: Muestra una lista de elementos, por ejemplo el más sencillo es una lista desordenada. Notar la clase de la lista completa y las clases de los elementos de la lista.

An item

A second item

A third item

A fourth item

And a fifth one

```
<ul class="list-group">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
  <li class="list-group-item">A fourth item</li>
  <li class="list-group-item">And a fifth one</li>
</ul>
```


- Mejor hacemos una implementación práctica. Usando botones, donde es importante notar que ***no se usa la clase btn*** y tambien se puede ***realizar con anclas*** pero en ellas no se puede poner un elemento como ***disable***.
- Podemos quitar los bordes y las esquinas de la lista agregando en la clase list-group: `list-group-flush`
- Podemos numerar los ítems de la lista, cambiando la lista `ul` por `ol` y agregando las clases: `list-group list-group-numbered`
- Podemos cambiar el color y estilo de cada ítem con lo siguiente en la clase del item: `list-group-item-success` (mostrar diferencias al colocar esa etiqueta de dos formas en la clase)
- Algo más completo:

<p>List group item heading 3 days ago</p> <p>Some placeholder content in a paragraph.</p> <p>And some small print.</p>
<p>List group item heading 3 days ago</p> <p>Some placeholder content in a paragraph.</p> <p>And some muted small print.</p>
<p>List group item heading 3 days ago</p> <p>Some placeholder content in a paragraph.</p> <p>And some muted small print.</p>

- Checkboxes: Se pueden implementar dentro de una lista, y la parte del input que se palomea se puede agregar con las siguientes etiquetas:
`class="form-check-input me-1" type="checkbox"`

<input type="checkbox"/> First checkbox
<input type="checkbox"/> Second checkbox
<input type="checkbox"/> Third checkbox
<input type="checkbox"/> Fourth checkbox
<input type="checkbox"/> Fifth checkbox


Scrollspy

Descripción: Es una combinación de un scroll con una barra de navegación, donde a medida que el usuario mueve el scroll, la barra de navegación indica donde está posicionado el usuario.


- Lo primero que necesita el scrollspy es una barra entonces la creamos:
- En segundo lugar ya pensamos en el contenido del cual indicaremos su ubicación. Es importante que el elemento donde se encuentra el scrollspy tenga posición relativa y overflow de scroll, esto se indica en el archivo css. Entonces agregamos al body estas características. Y lo que resulta importante es agregar un id a la parte de la navegación que queremos, en este caso lo haremos al h3
- Resulta relevante también agregar en el body un data-bs-target que identifique a la barra de navegación, por cual agregamos un id a la barra de navegación y finalmente se le indica al body que tendrá un data-bs-spy del tipo scroll y finalmente para que no se mueva la barra de la parte superior agregamos la propiedad fixed-top a la clase.

Media Object


Descripción: Es otro componente de Bootstrap que nos permite colocar por un elemento alineado con otro, por ejemplo una imagen de perfil con la descripción en un costado, se alinea la imagen con el texto, y nos permite colocarlo en la izquierda, derecha o anidarlo. Como se ve en el siguiente ejemplo más completo.




John Doe *Posted on February 19, 2016*
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut la




John Doe *Posted on February 20, 2016*
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt



John Doe *Posted on February 21, 2016*
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor



Jane Doe *Posted on February 20, 2016*
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt



Jane Doe *Posted on February 19, 2016*
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut la

IMPORTANTE: Quedo removido del Bootstrap 5

Carrusel