**Introduction to Bootstrap 4**

Bootstrap es un toolkit css que permite hacer páginas responsivas fácilmente y además darles estilos. Provee también de componentes propios con funcionalidad añadida. Extensiones para Visual Studio Code: Highlight matching tag + live server.

Para hacer una página con bootstrap se parte de una starter-template. Podemos crear una carpeta en local, y dentro crear un archivo index.html. Después pegamos el contenido html de la starter-template. Para abrirlo con VScode arrastramos la carpeta al editor.

**Shift+alt+F** formateo automático.

Se incluye en el head el link al css y al final del body los includes de los js: jquery, popper y bootstrap en este orden.

Descargando el paquete compilado: bootstrap grid, bootstrap reboot (normalizar css entre distintos navegadores) o bootstrap. Copiamos solo bootstrap css (que incluye a los otros dos) y lo pegamos en la carpeta del proyecto. Sustituimos los cdn del css y del js, tanto bootstrap como popper js. JQuery lo dejamos estar como cdn.

Con la extensión live server, cada vez que guardas con control S se refresca el nuevo contenido en el navegador, sin tener que recargar la página.

Main components of Bootstrap

Los componentes principales de bootstrap son:

1. .container
2. .row
3. .column

Notar que .**row**y .**column**siempre se aplican a **divs**. Pero .**container**class se puede asociar a otros elementos también como **header** o **footer**. Rows van dentro de containers y columns van dentro de rows.

VSCode tiene autocomplete de código. Escribiendo **div.container** e **INTRO** escribe

<div class="container"></div>

Creamos dentro de container un div row y dentro uno con la clase column. El container lleva un tamaño en función del tamaño de la pantalla. Si la pantalla cambia, cambia max-width del container.

Los container pueden ser fluid (ocupa el 100% del ancho de la pantalla).

<div class="container red white">

    <div class="row">

        <div class="column">Hello there</div>

    </div>

</div>

<div class="container-fluid green white">

    <div class="row">

        <div class="column">Hello there</div>

    </div>

</div>

6. Grid and columns

Columnas de grid: partimos de index.html 'vacío', solo contiene los includes y el vínculo con el css, además del markup de body.

Las columnas van dentro de una row y siempre se deben adaptar a 12, pero no necesariamente llegar a sumar 12. Lo único que no tiene sentido es que sumen más de 12 y pretender que encajen en una sola fila.

Para una anchura de pantalla se establece la cuantía menor de 12 que debe ocupar una columna:

div.col-sm-4 (small) (1/3 del ancho)

Si no pones pantalla (sm) y simplemente pones class col, ocupa toda la fila.

Si pones varios col sin tamaño establecido se reparten el espacio. **Poniendo incluso 18 cols sin tamaño no las organiza en filas de 12, sino que mete las 18 en una única fila**, con el mismo ancho.

El la doc de bootstrap, sección grid layout, hay tabla con tamaños contemplados:

1. sin sufijo (extra small) <575
2. -sm- (small) <768
3. -md- (medium) <992
4. -lg- (large) <1200
5. -xl- (extra large) >1200

Cuando marcamos una col podemos especificar la porción de la fila que queremos establecer para esa columna en cada tamaño de pantalla.

Podemos poner col col-sm-6, que vale por: para pantallas extra small la columna ocupa todo el ancho, pero para pantallas small ocupa la mitad.

Tienen un margen lateral de 15px. medianil o gutter. A veces, si el elemento empieda línea hay que anular ese margen izquierdo para que quede enrasado.

El tamaño especificado para un tipo de tamaño es válido para todos los tamaños iguales o mayores, a no ser que se sobrescriba la propiedad para tamaños mayores.

Si a las columnas se les pone un cardinal, todo lo que pase de 12 se encabalga en la línea siguiente, aún si se trata de un container fluid (que ocupa todo el ancho de la pantalla).

Si no se pone cardinal da igual el tamaño de la pantalla, se ponen las columnas intentando encajar todas. Sería bueno saber qué pasa cuando en las columnas empieza a haber elementos de tamaño fijo.

Ejemplo: queremos 4 columnas en small(>=576) y más, pero 2 columnas en large (>=992) y full en menores que small.

Esto da una columna por fila en col, 4 por fila en sm y md, y dos columnas por fila en lg y xl-

div.container

div.row

div.col-sm-3 col-lg-6

div.col-sm-3 col-lg-6

div.col-sm-3 col-lg-6

div.col-sm-3 col-lg-6

Como observación, no es que la suma de cardinales mayor que 12 para una fila carezca de sentido, sino que cuando es así, el contenido se acomoda en sucesivas filas conforme a los números.

<div class="container mt15">

    <div class="row">

        <div class="col-sm-3 col-lg-6 white red"> 1 col or 4 cols or 2 cols (1)</div>

        <div class="col-sm-3 col-lg-6 white green">1 col or 4 cols or 2 cols (2)</div>

        <div class="col-sm-3 col-lg-6 white blue">1 col or 4 cols or 2 cols (3)</div>

        <div class="col-sm-3 col-lg-6 white black">1 col or 4 cols or 2 cols (4)</div>

    </div>

</div>

Si queremos imponer, por ejemplo, 4 columnas independientemente del ancho de la pantalla, lo hacemos con col-3:

<div class="container mt15">

    <div class="row">

        <div class="col-3 white red">4 cols (1)</div>

        <div class="col-3 white green">4 cols (2)</div>

        <div class="col-3 white blue">4 cols (3)</div>

        <div class="col-3 white black">4 cols (4)</div>

    </div>

</div>

Las column tiene que estar dentro de una row, pero no tiene por qué sumar 12. Una row puede corresponderse realmente con varias filas.

Es el nombre de la clase contenedora para divs de class column, pero la relación no es estricta.

7. Grid and columns

Para anidar contenido metemos rows dentro de columns y aplicamos los mismos principios.

Section 3: Project 1

Creamos carpeta nueva myproject1, con carpetas css y js conteniendo bootstrap.css y bootstrap.bundle.js

Copiamos el navbar de uno de los ejemplos de la página de bootstrap y lo vamos acomodando a nuesto proyecto.

elemento html **nav**

class **navbar**(mandatory)

**navbar-expand-md** significa que a partir de tamaño md se ve el menú, sino aparece colapsado como en móvil.

**navbar-dark** ajusta el color de las letras para un **bg-dark**, y este ajusta un fondo oscuro para la barra.

**fixed-top** es para que esté siempre visible en el top de la página, aunque el resto haga scroll.

Dentro del navbar, el div con la ul es el menú. En el del ejemplo hay un link deshabilitado y un drop down, que vamos a quitar para seguir el ejercicio, pues buscamos crear una página como project1.

El **ul**tiene clase **navbar-nav** y los li tienen clase **nav-item**. Los elementos son a href y tienen clase **nav-link** las clases **aria-controls**, **aria-expanded** **aria-label**... son para screen readers.

En el ul, el primer elemento es Home, y tiene la clase active además de **nav-item**. Dentro del a hay un span con clase **sr-only** (screen reader only)

Vamos a poner un Jumbotron, que es como un cartel grande en la cabecera de la página.

Página Bootstrap, Components, Jumbotron. div class jumbotron:

<div class="container">

    <div class="jumbotron">

        <h1 class="display-3">Hello, people!</h1>

        <p>Welcome to my portfolio website!</p>

    </div>

</div>

10. Project 1 Jumbotron

La clase display va de 4 a 1, siendo el 1 el más grande: **display-3**

Usamos <p class="**lead**"> para el comentario bajo el título, pues resalta más.

Para dar el margen en el top del jumbotron, como en la referencia:

utility class para ello, aplicada al container **mt-5** (va del 1 al 5 siendo 5 el mayor).

**Ojo**! si el navbar tiene la clase **fixed-top** flota sobre el contenido, de modo que el margen mt-5 será respecto al tope superior del documento. Si se le quita la clase fixed-top al navbar este se convierte en un elemento 'que ocupa espacio' y entonces el margen de mt-5 es respecto al navbar, como deseamos.

También existen las utility classes **mb-[1-5]**, **ml-[1-5]**, **mr-[1-5]** para bottom, left y right respectivamente.

Hay más clases de utilidad interesantes que se pueden ver en Utilites: Border, Spacing...

11. Project 1 About Me

La sección About Me tiene poca chicha:

Apenas un container con text-center, una row, una col y las clase display para los tamaños de los elementos h1 y h3.

<div class="container text-center">

    <div class="row">

        <div class="col">

            <h1 class="display-2">About Me </h1>

            <p class="font-weight-bold">My name is Daniel Carregal and I am a freelance photographer.< /p>

                    <h3 class="display-4"> Check out my work below </h3>

        </div>

    </div>

</div>

Doc under Layout Content Typography -> Display classes

12. Project 1 Work Section

display-[1-4] siendo 1 el más grande, hace que los h1-h6 se vean iguales con el mismo display-x. La clase de display sobrescribe los tamaños de los elementos h.

Nuevo container con row y tres cols: como queremos 3 cols, son col-lg-4. Dentro de cada columna metemos un div.card

Dentro de cada div.card metemos una img.card-img-top

Tras el img.card-img-top metemos un div.card-body

card - card-body(<div>) - card-title(<h>) - card-text(<p>)

a.btn btn-dark btn-sm sin referencia a card

al container le ponemos my-5 (arriba y abajo, o sea, mt y mb a la vez).

<div class="col-lg-4">

    <div class="card"> <img src="./assets/animals.jpg" alt="animals" class="card-img-top">

      <div class="card-body">

        <h5 class="card-title">Animal photos</h5>

        <p class="card-text">Some animal photos</p>

<a type="button" href="#" class="btn btn-dark btn-sm">Check it out!</a>

      </div>

    </div>

</div>

13. Project 1 Footer

<footer class="container-fluid bg-dark text-light py-4">

    <div class="container">

        <div class="row">

            <div class="col-sm-3">

                <nav class="navbar navbar-expand-lg navbar-dark bg-dark pl-0"> <a class="navbar-brand" href="#">Daniel

                        Carregal</a> </nav>

            </div>

            <div class="col-sm-9">

                <p class="float-sm-right pt-3">&copy;Daniel Carregal - All Rights Reserved </p>

            </div>

        </div>

    </div>

</footer>

14. Project 1 Responsive

Para hacerlo **responsive**: se ve mal el espacio cuando se colapsan por anchura. También necesitamos que cuando se pasa a una card por columna, el texto esté centrado.

Tambén necesitamos que colapse bien el footer. Podemos aplicar un mb-3 (margin bottom) pero no queremos que se aplique en todos los casos, sino cuando se vean solo una columna.

Entonces aplicamos un margen de 0 en los casos en los que no queremos que salga, que es en los que pintamos más de una columna :)

**div.col-md-4 mb-3 mb-md-0** así solo se aplica si es menor de md, caso en el que se pinta una sola col.

La misma lógica aplicamos para que el texto se vea centrado si hay una columna, y a la izquierda si se ven 3:

<div class="row text-center text-lg-left"> **bootstrap utility classes**

En cuanto al footer, lo que está causando el problema es el float-right.

Para resolverlo se aplica solo al tamaño para el que no falla:  **sm** en adelante.

Project 2 16. Project 2 Navigation

Segundo proyecto, sobre página a favor del consumo de fruta. Tiene un navbar, tiene modales con vídeos de youtube, tiene un formulario, tiene footer, tiene cards, contenedores con imágenes flotando a izquierda y derecha, jumbotrons, dropdowns, etc.

Volvemos a la página de bootstrap y cogemos en la sección de ejemplos el código para el navbar. Le quitamos el formulario del final.

Cambiamos las clases **.navbar-dark** y **.bg-dark** por **navbar-light** y **bg-light** (utility classes para text colors y background colors). Eliminamos la clase  fixed-top, para que el contenido empiece a apilarse bajo el navbar en cuando a colocación.

Para darle negrita a todos los textos del nav, ponemos la clase **font-weight-bold** en el elemento nav.

Para eliminar el highlight en el enlace About Us, dado que estamos en el home, le quitamos la clase active.

17. Project 2 Carousel

Para añadir algo de espacio entre el encabezado del dropdown y la flecha, hay que colocar 'algo' que ocupe ese espacio tras el texto del encabezado. El autor mete un salto de línea, yo he probado con un espacio y funciona. Pero también he metido un elemento inline al que se le puede fijar el ancho, como lo es <img width="5px"/>.

Slider, es un componente de bootstrap que se llama **Carousel** así que vamos a la sección de ejemplos de bootstrap y lo cogemos.

Lo vamos a meter dentro de un container, ya que no es full width, y dentro de una row y col.

En el carousel hay una **ol** con los **indicators** (los segmentos en el inferior de la vista que sirven como enlace a cada una de las vistas sobre las que el carousel hace scroll). Luego está un **div.carousel-inner** con cada una de esas vistas. Después los **a href** con las flechas a izq y der para avanzar o retroceder manualmente. Se puede ajustar el tiempo de visualización de cada vista, así como otras facetas del control.

Uno de los elementos de **div.carousel-inner** tiene la clase **active**.

Las clases aplicadas a las imagenes: **d-block** (display-block) y **w-100** (ancho del 100%).

Añadimos captions a los elementos, vemos en la documentación:

<div class="carousel-caption d-none d-md-block">

    <h5>First slide label</h5>

    <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>

</div>

Pero no vamos a utilizar el h y el p, solo un h.

Les da color a las caption con las clases **text-warning**, **text-danger** y  en el **h2**.

Si añadimos más *.carousel-item* y no añadimos más elementos a **ol.carousel-indicators**, los elementos adicionales no serán alcanzables mediante los indicadores, pero si con los botones de next y previous.

18. Project 2 Newest arrivals

Siguiente sección, una linea horizontal, no tiene ancho total con lo que va en un div.container. Es oscura así que **hr.bg-dark**. Tiene espacio por arriba así que hr.mt-5.

Encabezado 'Newest arrivals', en un h3 (da igual el tipo de h, pues al aplicarle la clase display-4, esta marca el tamaño).

Hay tres columnas con el borde elíptico, pero queremos una si la pantalla es menor de md. Las hacemos **img.img-fluid** para que tengan **100% de ancho y altura auto**.

Bootstrap utility class para el border: **border border-warning rounded-circle**.

**->TIP:** Si no se van a crear columnas no hay necesidad de crear fila. O sea, si solo se va a tener contenido que ocupe la totalidad del ancho de la parrilla, se mete en un container pero no en una row.

Dentro del mismo container, pero fuera de la row, creamos un div normal, dándole las clases **text-center**.

Dentro simplemente colocamos un **a.btn.btn-info.btn-lg** (boton info un poco más grande): See All

De esta forma está centrado en el contenedor.

El siguiente fragmento, un banner verde en el centro de la pantalla, no lo metemos en un container solo para ver que no es estrictamente necesario hacer eso.

Creamos un **div.w-50.mx-auto.py-5.text-center.text.white** : 50% ancho de pantalla, bordes automáticos (lo centra), padding 5 unidades, text centrado y blanco.

Creamos dentro un encabezado display-3 (tamaño medio).

Por último, una línea verde. Va en **container.text-center** y es **hr.bg-success** (para ser verde).

<div class="w-50 bg-success p-5 my-5 mx-auto text-center text-white">

    <h3 class="display-3">Fruit is good!</h3>

</div>

<div class="container">

    <hr class="bg-success">

</div>

19. Project 2 Watermelons and Grapes

Para poner el contenido de las dos columnas que tienen texto e imagen (columna izq tiene texto a la iz e imagen a la derecha, y columna derecha) se hace uso de la parrilla.

Se definen las dos columnas para tamaños mayores que small. El p tiene **class lead** para tener la letra un poco más grande.

Se define una relación 7-5 para texto-imagen.

<div class="col-md-6">

    <div class="row">

        <div class="col-sm-7">

            <h3 class="text-success">Watermelons</h3>

            <p class="lead">Lorem ipsum dolor sit amet, consectetur adipiscing et, sed do eiusmod tempor incididunt ut

                labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi

                ut aliquip ex ea commodo consequat. </p>

        </div>

        <div class="col-sm-5"> <img src="./assets/watermelon.png" alt="watermelon"> </div>

    </div>

</div>

20. Project 2 Modals Section

Si en una parrilla se definen tres columnas y las dos primeras tienen medida, por ejemplo: col-sm-3 y col-sm-6, la última no hace falta darle medida, pues ocupará el ancho restante.

En un div col podemos querer tener una imagen a la derecha del div. Técnica offsetting: añadir columnas vacías a la izquierda. Se hace mediante clases css:

**col-md-3 offset-md-4** ocuparía desde el **slot 5 al 8** de la parrilla.

<div class="col-md-1 offset-md-2"><img src="./assets/banana.png" alt="banana"></div>

21. Project 2 Adding Modals

Buscamos los modales en la sección de componentes de la doc de bootstrap. Se componen de un **div con class modal** y con partes en **modal-content**, **modal-header**, **modal-body** y **modal-footer**.

Hay un botón con **data-toggle="modal"** y **data-target="#id\_del\_div"**. Hay un **div** con ese **id** que tiene el **layout del modal**.

En el div, dejamos el esqueleto con modal-header, body y footer.

Header: h5.modal-title

<div class="modal fade" id="bananaModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"

    aria-hidden="true">

    <div class="modal-dialog" role="document">

        <div class="modal-content">

            <div class="modal-header">

                <h3 class="text-warning">Bananas</h3>

            </div>

            <div class="modal-body"> <iframe src="http://www.youtube.com/embed/dP15zlyra3c?html5=1"></iframe> </div>

            <div class="modal-footer"> <button type="button" class="btn btn-secondary"

                    data-dismiss="modal">Close</button> </div>

        </div>

    </div>

</div>

Para poner un video Utilities - embed. Copiamos el ejemplo

<div class="embed-responsive embed-responsive-16by9"> <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/zpOULjyy-n8?rel=0" allowfullscreen></iframe></div>

Para poner los botones que abren los modales creamos una fila con dos columnas col-md-6

Hay que poner data-target con el id del div de cada modal.

Estilo del botón: amarillo con texto blanco y bold-> **text-white font-weight-bold bg-warning** (por qué no btn-warning?)

Hace **gordo** el botón dándole **padding px-5.**

Para hacer más grande el modal, al div.modal-dialog le damos clase **modal-lg**

22. Project 2 Footer

Para el footer, **container-fluid** (ocupa todo el ancho de la pantalla), y con bg-dark.

Row, y tres columnas 3, 3 y 6. md

Copiamos el nav y nos quedamos con la parte de brand.

Social icons font awesome website cdn. Copiamos el enlace. ~~Lo ponemos en el header debajo de bootstrap css [OLD].~~

Ahora para incluir **font-awesome** se hace con un elemento **<script>**

<script src="https://kit.fontawesome.com/dd900b9281.js" crossorigin="anonymous"></script>

Columnas para los iconos. Facebook linkedIn y twitter. El color a los iconos se lo da con las clases de Bootstrap: primary es azul, info es turquesa, success es verde. Les damos a los botones padding superior de 5.

<div class="col-md-3 pt-5"><a href="#" class="btn btn-primary btn-sm"><i class="fab fa-facebook-square fa-3x"></i></a></div>

El formulario lo mete en la columna final md-6. **form.form-inline**.

**Input**: cada elemento en bootstrap tiene que tener la class **form-control**.

Tenemos elemento **name** y elemento Email con los **type="text"** y **"email"**.

Para hacer los inputs del form más pequeños, **form-control form-control-sm**

Para los iconos de font awesome se utilizan **elementos i dentro de enlaces type button**:

***The <i> tag defines a part of text in an alternate voice or mood. The content of the <i> tag is usually displayed in italic***.

<a href="#" class="btn btn-primary btn-sm"><i class="fab fa-facebook-square fa-3x"></i></a>

23. Project 2 Responsive

Para ajustar paddings o margins en funcion del ancho de la pantalla:

<h3 class="display-4 text-center pt-2 pt-lg-4">Newest arrivals</h3>

<div class="col-md-4 mb-5 mb-md-0"> <img src="./assets/mango.jpg" alt="mango"

        class="img-fluid border border-warning rounded-circle"> </div>

Para modificar si algo es visible o no en función del tamaño de la pantalla:

<div class="w-50 bg-success p-5 my-5 mx-auto text-center text-white d-none d-lg-block">

    <h3 class="display-3">Fruit is good!</h3>

</div>

24. Project 2 Responsive 2

Bootstrap utility: display.

**lg**

texto + img | img + texto

**<lg**

1 col lo mismo de antes

**<md**

text-center

img

texto

img

texto

Se plantea el problema con la imagen del melón, que para tamaños de pantalla mayores de sm el div va después del div con el texto, pero para tamaños inferiores debe ir antes. Mi solución consiste en crear dos div, uno antes y uno después, y que se visualice el que yo quiera en función del tamaño de pantalla. Bootstrap ofrece la forma correcta de hacer eso para no tener html duplicado, y se llama **ordering.**

Mi solución (funciona):

<div class="col-lg-6">

    <div class="row">

        <div class="col-sm-5 d-block **d-sm-none**"> <img src="./assets/watermelon.png" alt="watermelon"> </div>

        <div class="col-sm-7">

            <h3 class="text-success">Watermelons</h3>

            <p class="lead">Lorem ipsum dolor sit amet, consetur adipiscing elit, sed do eiusmod tempor incididunt ut

                labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi

                ut aliquip ex ea commodo consequat. </p>

        </div>

        <div class="col-sm-5 d-none **d-sm-block**"> <img src="./assets/watermelon.png" alt="watermelon"> </div>

    </div>

</div>

Con ordering:

<div class="col-lg-6">

    <div class="row">

        <div class="col-sm-7 **order-2 order-sm-1**">

            <h3 class="text-success">Watermelons</h3>

            <p class="lead">Lorem ipsum dolor sit amet, consectur adipiscing elit, sed do eiusmod tempor incididunt ut

                labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi

                ut aliquip ex ea commodo consequat. </p>

        </div>

        <div class="col-sm-5 order-1 order-sm-2"> <img src="./assets/watermelon.png" alt="watermelon"> </div>

    </div>

</div>

mx-auto para centrar el contenido por ejemplo de botones. text-center funciona, pero no es la solución fetén. Se usa text-center para centrar más que mx-auto

**float-left**, **float-right** y **float-none** se pueden aplicar a los distintos tamaños de pantalla.

**w-100, w-50**, son para decir que algo ocupe un cierto porcentaje de la pantalla

Para **centrar**, meter algo en un **div.w-100.text-center** funciona.

<div class="container-fluid bg-dark mt-5 w-100 text-center text-lg-left">

    <div class="row">

        <div class="col-lg-3">

            <nav class="navbar"> <a class="navbar-brand mx-auto mx-lg-0" href="#"><img src="./assets/logo.png"

                        alt="logo"></a> </nav>

        </div>

        <div class="col-lg-3 pt-5"> <a href="#" class="btn btn-primary btn-sm"><i

                    class="fab fa-facebook-square fa-3x"></i></a> <a href="#" class="btn btn-info btn-sm"><i

                    class="fab fa-twitter-square fa-3x"></i></a> <a href="#" class="btn btn-success btn-sm"><i

                    class="fab fa-linkedin fa-3x"></i></a> </div>

        <div class="col-lg-6 pt-4 pb-5 pb-lg-0">

            <form action="" class="form-inline">

                <h4 class="text-success w-100 text-center text-lg-right">Sign up for our newsletter!</h4>

                <div class="w-100 text-center text-lg-right"> <input type="text" name="name"

                        class="form-control form-control-sm" placeholder="Name"> <input type="email" name="email"

                        class="form-control form-control-sm mt-2 mt-sm-0" placeholder="E-mail"> <button

                        class="btn btn-sm btn-success text-white mt-3 mt-sm-0 float-left float-sm-none"

                        type="submit">Sign

                        Up!</button> </div>

            </form>

        </div>

    </div>

</div>

Section 5: 05 Project 3

Barra superior, cabecera de la página: nav lo copia de ejemplos. Le quita el form. y el dropdown., le quita el fixed-top.

bg-danger, font-weight-bold, text-white, cambiar los textos.

Añade el logo. Mete el navbar en un <header>. El resultado visual es el mismo, pero puede ser útil si en la cabecera se quieren meter más cosas.

26. Project 3 Navigation

<header>

    <nav class="navbar navbar-expand-sm text-white navbar-dark bg-danger font-weight-bold"> <a class="navbar-brand"

            href="#"><img src="./assets/logo.png" alt=""></a> <button class="navbar-toggler" type="button"

            data-toggle="collapse" data-target="#navbarsExampleDefault" aria-controls="navbarsExampleDefault"

            aria-expanded="false" aria-label="Toggle navigation"> <span class="navbar-toggler-icon"></span> </button>

        <div class="collapse navbar-collapse" id="navbarsExampleDefault">

            <ul class="navbar-nav mr-auto">

                <li class="nav-item active"> <a class="nav-link" href="#">Home<span class="sr-only">(current)</span></a>

                </li>

                <li class="nav-item"> <a class="nav-link" href="#">Breaking News</a> </li>

                <li class="nav-item"> <a class="nav-link" href="#">Contact</a> </li>

            </ul>

        </div>

    </nav>

</header>

27. Project 3 Jumbotron

**section.jumbotron .text-center .bg-dark, .text-white**

quitarle el borde redondeado: **.rounded-0**

h1 display-3

**p.lead** copiar y pegar el texto.

Botones pegados: dentro de un **div.btn-group**

button.btn.btn.primary

button.btn.btn.success

Espaciado: le da my-2 a los botones.

<section class="jumbotron rounded-0 bg-dark text-white text-center">

    <h1 class="display-3">No. 1 News Website</h1>

    <p class="lead">The best place to catch latest news from around the world.</p>

    <div class="btn-group flex-wrap"> <button class="btn btn-primary my-2 text-white mx-auto">Read the latest

            news</button> <button class="btn btn-success my-2 text-white mx-auto">Sign up tp stay informed</button>

    </div>

</section>

28. Project 3 News section

news section: **div.container div.row div.col-md-6**

similar a la sección con el watermelon y las grapes: img + title + text

Ahora diferente. Media/layout **media object**

dentro de la col un **div.media y dentro un img + un div.media-body**

copiar el texto-> h5 y p Para el control+espacio meterlos dentro de un tag **kbd** (teclado)

Espacio a la izq del texto .mr-3 en la **img.img-thumbnail** para el border de la imagen.

Al media div añadir las clases .**shadow border-top border-dark p-3**

Copiamos el primer div. Cambiamos el texto. pegar la imagen después del **media-body**.

Hay que centrar la imagen verticalmente **img.align-self-center**

<div class="media shadow border-top border-dark p-3"> <img class="img-thumbnail mr-3 d-none d-sm-block"

        src="./assets/media.jpg" alt="media">

    <div class="media-body"> <img class="img-thumbnail mb-3 d-block mx-auto d-sm-none" src="./assets/media.jpg"

            alt="media">

        <h5>News title 1</h5>

        <p>Lorem ipsum dolor sit amet, consectetur adiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

            magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco. laboris nisi ut aliquip ex ea

            commodo consequat. You can usually toggle fullscreen by pressing <kbd>ctrl + enter</kbd></p>

    </div>

</div>

29. Project 3 Our reporters and Most popular news

**div.container-fluid.bg-light div.row div.col-md-6**

Los headings son h4.display-4

Content: tables doc

Bajo el h4 una **table.table** (una tabla de bootstrap siempre tiene que tener la clase table) y otras se pueden usar para estilos, pero table siempre.

**table-striped table-responsive-lg**

Es responsive por debajo de lg

**thead tr th** # first-name last-name more-info

**tbody tr td** 1 Donal Duck N/A N/A N/A N/A

quitarle el border top a la tabla lo hace en cada columna, probar en la fila .**border-top-0**. Nope, **hay que quitarle el borde a cada columna td**

La sección está dentro de un container-fluid, pero **dentro de otro container (boxed).**

mt-5 al container-fluid para el spacing con la section anterior.

Tambien necesita padding  pt-3

<table class="table table-striped table-responsive-lg">

    <thead>

        <tr>

            <th class="border-top-0">#</th>

            <th class="border-top-0">First Name</th>

            <th class="border-top-0">Last Name</th>

            <th class="border-top-0">More info</th>

            <th class="border-top-0">More info</th>

            <th class="border-top-0">More info</th>

            <th class="border-top-0">Last info</th>

        </tr>

    </thead>

La tabla tiene scroll en virtud de responsive.

Siguiente elemento es una lista:

div.col-md-6

h4 display-4

**Components list**

**ul.list-group** <- en bootstrap van asi las listas.

**li.list-group-item**

tienen una interrogación con icono. Espacio bajo el h4 pb-3

**badges**components **span.badge.badge-danger**

**tooltip**docs-> **inicializado con js (ver doc bootrsap)**

y ver el html en la doc dentro del html del span con los attr adicionales:

data-toggle="tooltip" data-placement="right" title="Tooltop on right"

    <ul class="list-group">

        <li class="list-group-item">Sheep spotted flying <span class="badge badge-danger" data-toggle="tooltip"

                data-placement="right" title="Lorem ipsum dolor sit amet, consectetur adipiscing elit.">?</span></li>

        <li class="list-group-item">People can live<span class="badge badge-danger" data-toggle="tooltip"

                data-placement="left" title="Lorem ipsum dolor sit amet, consectetur adipiscing elit.">?</span></li>

        <li class="list-group-item">Aliens are among us <span class="badge badge-danger" data-toggle="tooltip"

                data-placement="right" title="Lorem ipsum dolor sit amet, consectetur adipiscing elit.">?</span></li>

    </ul>

inicialización js de los tooltip:

<script> $(function () { $('[data-toggle="tooltip"]').tooltip() }) </script>

30. Project 3 Recent events

Otra columna **div.col-md-6**. Si sabes lo que haces puedes meter más de 12 *cols* en una fila, vaya que puedes ir apilando sin tener que codificar nuevas div.row.

Dentro de la fila, tras las dos primeras columnas, mete un **div.w-100 para separar con py-3**

después col-md-6 con un **acordeón: components - collapse (doc)**

copiamos uno de la pagina de bootstrap

**div.accordion** tiene dentro **div.card** con **div.card-header** y dentro varios **div.collapse** con **.show** o no y attr **data-parent="#id"**

Se va a quitar la cabecera y sustituye el h por un botón.

ese botón tendrá los **data-toggle** y demás atributos necesarios.

<div class="accordion" id="accordionExample">

    <div class="card"> <button class="btn btn-primary text-left text-white font-weight-bold py-3" type="button"

         data-toggle="collapse" data-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne"> Team

                building in the woods </button>

        <div id="collapseOne" class="collapse show" aria-labelledby="headingOne" data-parent="#accordionExample">

            <div class="card-body"> Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry

                    richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch.

</div>

        </div>

    </div>

…

</div>

31. Project 3 Useful info

h display-4

para los tabs vamos a **components navs: ul.nav.nav-tabs role="tablist"**

Hay un ul para las tabs y el **div para el contenido**, que cambia conforme a la tab en la que estás.

El de contenido tiene **tantos div dentro como tabs**. Están **ligados**por el **id**.

**Ul-> li.nav-item a.nav-link.active data-toggle="tab" href="#id\_div\_contenido" role="tab"**

texto de los tab blue bold

<div class="tab-content mt-3" id="myTabContent">

    <div class="tab-pane fade show active" id="about" role="tabpanel" aria-labelledby="home-tab">

        <h3 class="text-success">About Us</h3>

        <p class="lead">Nunc feugiat quis tortor a aliquam. Vivamus ut suscipit nunc. Vestibulum mollis rutrum leo,

            tristique varius enim faucibus a. Fusce luctus ullamcorper purus nec pellentesque.</p>

    </div>

    <div class="tab-pane fade" id="profile" role="tabpanel" aria-labelledby="profile-tab">

        <h3 class="text-success">Company Profile</h3>

        <p class="lead">Nunc feugiat quis tortor a aliquam. Vivamus ut suscipit nunc. Vestibulum mollis rutrum leo,

            tristique varius enim faucibus a. Fusce luctus ullamcorper purus nec pellentesque.</p>

    </div>

32. Project 3 Contact Us

**div.container-fluid div.row div.col-md-6**

**heading h3 display-4** Contact Us

Dentro del form, por cada elemento del formulario hay un **div.form-group**

los input son **.form-control** y tienen placeholder

Para que estén en una línea los tres primeros elementos del form los mete dentro de un div.row (**se hace parrilla dentro del formulario**).

Hace form-group para cada conjunto de elementos que quiere que se muestren juntos. Esto incide en los márgenes entre inputs. Si cada control está en un form-group sale uno en cada línea, sino salen en una línea por form-group.

Después, para poder ajustarlos a tamaños de columna, los mete en una row y en cols.

Para tener un form-group con un checkbox tiene que tener **.form-check**y dentro un **input type="checkbox"** y su **label.form-check-label for="exampleCheck1"**

Le damos padding px-5 a la col-md-6 y pb-5

Para poner el icono de email, dentro de un **input-group** va un **div.input-group-prepend** que tiene dentro un **div.input-group-text.**

Entonces hay que crear un **input-group** y meter dentro el **input-group-prepend** y el**input** que ya teníamos:

<div class="input-group">

        <div class="input-group-prepend"><span class="input-group-text bg-success text-white">@</span></div> <input

            type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"

            placeholder="Enter email">

    </div>

    <form>

        <div class="form-group">

            <div class="row">

                <div class="col-12 col-sm-3"> <select class="form-control" id="exampleFormControlSelect1">

                        <option>Mr.</option>

                        <option>Ms.</option>

                    </select> </div>

                <div class="col-12 col-sm mt-3 mt-sm-0"> <input type="text" class="form-control" placeholder="First name">

                </div>

                <div class="col-12 col-sm mt-3 mt-sm-0"> <input type="text" class="form-control" placeholder="Last name">

                </div>

            </div>

        </div>

        <div class="form-group">

            <div class="row">

                <div class="col-md-6">

                    <div class="input-group">

                        <div class="input-group-prepend"><span class="input-group-text bg-success text-white">@</span></div>

                        <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"

                            placeholder="Enter email">

                    </div> <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone

                        else.</small>

                </div>

                <div class="col-md-6"> <input type="text" class="form-control" placeholder="Phone"> </div>

            </div>

        </div>

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

            <div class="row">

                <div class="col-12 col-sm"> <input type="checkbox" class="form-check-input" id="exampleCheck1"> <label

                        class="form-check-label" for="exampleCheck1">Sign up for our newsletter</label> </div>

                <div class="col mt-2 mt-sm-0 pl-0"> <button type="submit"

                        class="btn btn-primary float-sm-right">Submit</button> </div>

            </div>

        </div>

    </form>

33. Project 3 Contact Information1

Contact information section

**div.col-md-6.bg-dark.text-white**

heading **h1 class="font-weight-normal"** reduce el bold del h1

**table.table tbody tr td**

<div class="col-md-6 bg-dark text-white pt-3 p-sm-5">

    <h1 class="font-weight-normal">Contact Information</h1>

    <table class="table">

        <tbody>

            <tr>

                <td>E-mail</td>

                <td class="font-weight-bold">news@news.com</td>

            </tr>

            <tr>

                <td>Phone</td>

                <td class="font-weight-bold">111-222-333</td>

            </tr>

            <tr>

                <td>Adress</td>

                <td class="font-weight-bold">123 Baker Street</td>

            </tr>

        </tbody>

    </table>

</div>

34. Project 3 Footer

Elemento **footer**de html: **footer.container-fluid bg-secondary**(gris claro)

**div.row div.col-md-4** copiamos **navbar brand.**

**text-light** lo hace gris, no blanco. Para el enlace en el navbar del footer.

Para el menú de la derecha del footer: **div.col-md-8**

**components navs: ul.nav con li's y a formatear**.

padding top 2 y **float-right** a la lista (se revisará para responsive)

pone el primero a active y con **font-weight-bold** (en realidad no tiene estilo asociado a la clase active, con lo que tiene que ser manual).

<footer class="container-fluid bg-secondary">

    <div class="row">

        <div class="col-sm-4">

            <nav class="navbar navbar-expand-sm"> <a class="navbar-brand text-light" href="#">News Website</a> </nav>

        </div>

        <div class="col-sm-8 text-light pt-2">

            <ul class="nav float-sm-right flex-column flex-sm-row">

                <li class="nav-item"> <a class="nav-link active text-light font-weight-bold" href="#">Home</a> </li>

                <li class="nav-item"> <a class="nav-link text-light" href="#">Breaking News</a> </li>

                <li class="nav-item"> <a class="nav-link text-light" href="#">Contact</a> </li>

            </ul>

        </div>

    </div>

</footer>

35. Project 3 Cookies notification

Cookies notificación, **components alerts** copiamos un **alert dismissable**

Pegamos después del footer. Color warning

Quitamos el margen inferior **mb-0 fixed-bottom** (aunque hagamos scroll para arriba se ve en el bottom)

<div class="alert alert-warning mb-0 alert-dismissible fixed-bottom fade show" role="alert"> We use cookies on this

    website. <button type="button" class="close" data-dismiss="alert" aria-label="Close"> <span

            aria-hidden="true">&times;</span> </button> </div>

36. Project 3 Responsive 1

Al header solo cambiarle el mod. de tamaño para convertirse en menu mobile: md -> sm

Al **button-group** le damos **flex-wrap** (de esta forma se divide en dos botones individuales si falta espacio), y para centrarlos **mx-auto**. Los botones ya tenían **my-2** cada uno.

Para los **media objects**, a los que solo les habíamos dejado media-body e img, hay que hacer la técnica de tener más de una imagen y esconderla con **d-none y b-block** para aquellos tamaños donde aplique. Por debajo de **md**tienen que estar a todo ancho.

Al 2º div, **mt-4 mt-lg-0**

Para la imagen del media también aplica **mx-auto** para centrarla.

En el segundo media hace lo mismo.

Our reporters y Most popular news. Ajustar el responsive de la tabla para que se ajuste al modelo. Cambio a una en md.

Recent Events más o menos bien, pero Useful (tabs) mal.

Meter margen encima de useful info **pt-4 pt-md-0**

**ul flex-column flex-sm-row**para que cambie layout de elementos en fila o columna en función del tamaño.

**Form**: Debajo de lg debe ser una col col-lg-6.

Por debajo de sm tiene que ser un campo por línea.

**col-sm-3** para el género, todos los **md** deben ser **sm**.

Está mal en la primera línea first name y last name. Para hacerlos de todo el ancho por debajo de sm: **col-12 col-sm**

Hay que darles algo de espacio cuando están en una linea **mt-3 mt-sm-0**

37. Project 3 Responsive 2

Menos espacio, ya no es px-5 por debajo de sm

boton **float-sm-right** para que esté a la izq en pequeño

**col-12 y col-sm** para checkbox y botón

El botón tiene espacio a la izq porque la col tiene padding.

Se elimina con **pl-0 mt-2 mt-sm-0**

Siguiente contact info: quitar el padding en menor sm **p-sm-5 pt-3**

Footer: below sm las col cogen todo el ancho: **md por sm**

La lista de enlaces tiene **float-sm-right** y **flex-column flex-sm-row** para que se pongan uno encima de otro los enlaces por debajo de sm.