**Curso de Bootstrap 4.**

* **Bootstrap 4.**

Contenido

[Glosario 2](#_Toc532223689)

[Links útiles 3](#_Toc532223690)

[Capítulo 1. Introducción 4](#_Toc532223691)

[Capítulo 2. Estilos básicos. 4](#_Toc532223692)

[Capítulo 3. Entendiendo Boostrap Layout 7](#_Toc532223693)

[Capítulo 4. Usando componentes Navs y Navbar 20](#_Toc532223694)

[Capítulo 5. Overview de elemento de estilos 27](#_Toc532223695)

# Glosario

* CDN: Content Delivery Network
* .Container = se puede usar uno o varios, pero nunca uno dentro del otro.

# Links útiles

# Capítulo 1. Introducción

Instalación

* **CSS y JS**

Descargando todos los CSS y JS

* **CDN (Content Delivery Network)**

Se guarda en algún servidor y se accede a él. Cuando alguien visita un sitio que use el framework con CDN, queda guardado en el caché, y si visita otra página con el mismo CDN no necesita volverlo a descargarlo, haciéndolo más rápido.

* **Source Files**

Todos los archivos como documentación o archivos SASS (incluyendo obviamente los CSS y JS)

* **Package managers**

Como NMP, Rubygems, Composer o Nuget.

Crear una plantilla

* Se necesita los archivos o CDN
* jQuery
* Popper.js

(Primero jquery, después Popper y por último Bootstrap cuando se está cargando)

# Capítulo 2. Estilos básicos.

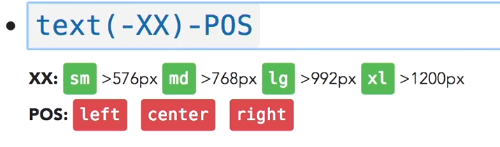
Tipografia

* ***Utiliza Reboot.css*** = Ayuda normalizar los estilos para que se parezcan en todos los navegadores. Usa REMS (unidad de tamaño relativa)
* 1 REM == 16px.
* Los estilos son más fáciles de *override* que en versiones anteriores.
* Se debe evitar **MARGIN-TOP** lo más posible ya que Bootstrap solo añade en el BOTTOM.
* **INHERIT** es una propiedad que utiliza Boostrap cada vez que puede.
* ***Border-box sizing****.* Hace más fácil calcular el tamaño de un elemento. Viene por *default*.
* Trata de usar fuentes nativas, para cada equipo.
* Existen clases llamadas: *h1,h2,h3.* En algunos casos puede querer usarse sin reemplazar la etiqueta
* También hay clases llamadas: *display-1, display-2*. Estos son más grandes que los *headings*.

Utilidades para tipografia

**Clases:**

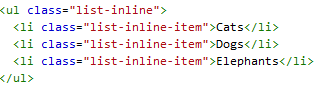
* *Text-justify:* Justifica los textos
* *Text-*nowrap: Permite que el texto no tenga limite. Es decir, que hace un *overflow* al elemento padre.
* *Text-(-XX)-POS*
* Text-sm-center
* Text-md-center
* Text-left
* Text-right



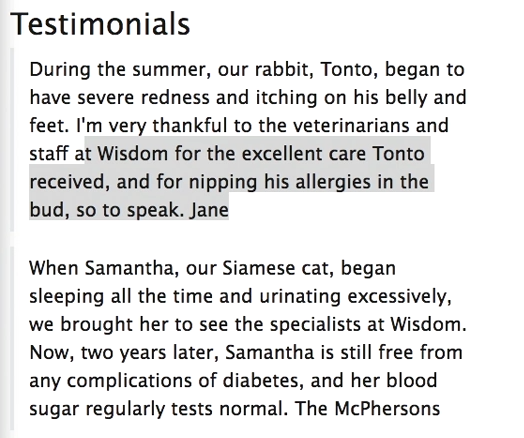
* *Text-lowercase; text-uppercase; text-capitalize.*
* *Font-weight-bold; Font-weight-normal; Font-italic.*

Blockquotes y listas

* *List-unstyled:* No poner los *bullets* de una lista
* *List-inline:* Se usa para hacer una lista en una misma linea. Se puede usar para *Breadcrumbs.*

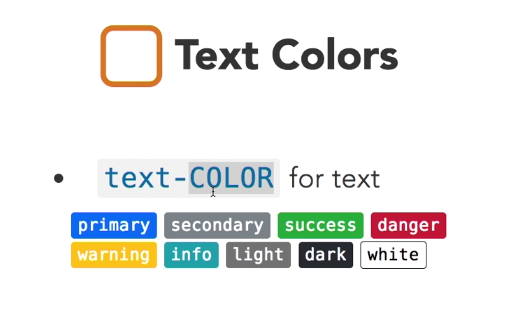


* La clase de *blockquote* le da un estilo para diferenciar cada cita con la barra vertical.



Colores de Bootstrap

Colores contextuales ya que tienen en realidad un significado:



Trabajar con imágenes o figuras

* *Img-fluid:* Añadir imágenes responsive
* *Rounded:* Provee unas esquinas redondeadas ligeramente o se puede indicar la dirección



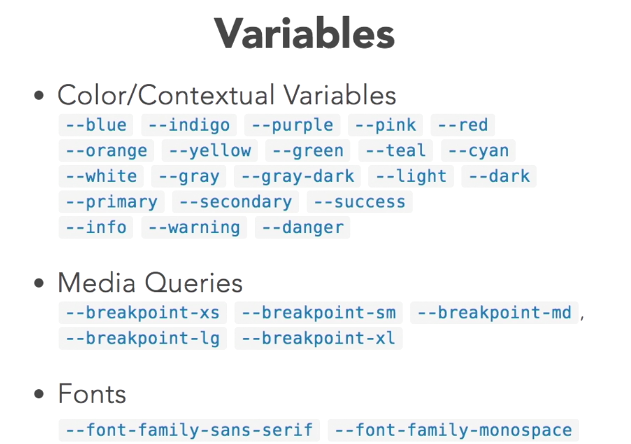
* *Img-thumbnail*

Alinear Imágenes: Son consideradas in-line. Se pueden hacer *float-left* o *float-right, text-center*

Si se quiere que sea un elemento block: *mx-auto.*

Variables CSS

* Es un nueva caracteristica en CSS.
* No soy soportados en muchos navegadores aún.
* Var () and :root
* Sirve para crear nuevos elementos



Para crear o redifinir una variable se usa :root

<style>

:root{

--pink: #C4226F;

}

</style>

Y para usar:

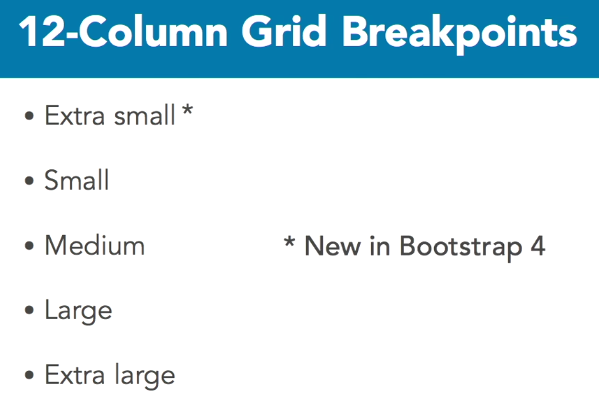
<h2 style="color: var(--pink);">Testimonials</h2>

# Capítulo 3. Entendiendo Boostrap Layout

El Grid es lo más importante de Boostrap. Es un framework que permite crear un *layout* a partir de su grid.

Container: pueden ser ancho de responsive fixed que cambian en ciertos *breakpoints.* También puede ser *fluid-containers* que toman el 100%

Bootstrap usa un grid de 12 columnas:



Sobre Boostrap Grid

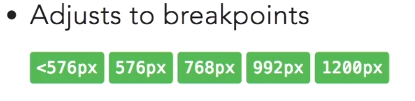
El Grid:

* Responsivo 12-columnas
* Usa Flexbox
* Estructura
  + Containers
  + Rows/columns (primero se crean las *rows* y dentro de estas, las columnas.

Contenedores (containers)

Pueden ser:

* Regular containers: centran el contenido y cambian en ciertos puntos de grid.

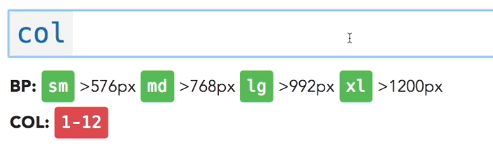


* Fluid containers: los cuales son siempre el ancho total del *viewport* del equipo.

Se usa porque se obtiene un padding de 15 pixeles en cada lado para asegurarse de que trabaje bien con fondos (*backgrounds*) y otros elementos.

Uso de Columnas

* Grid de 12 columnas
* Las columnas pueden span (es decir pueden tomar 2, 3, o hasta los 12 espacios del grid)

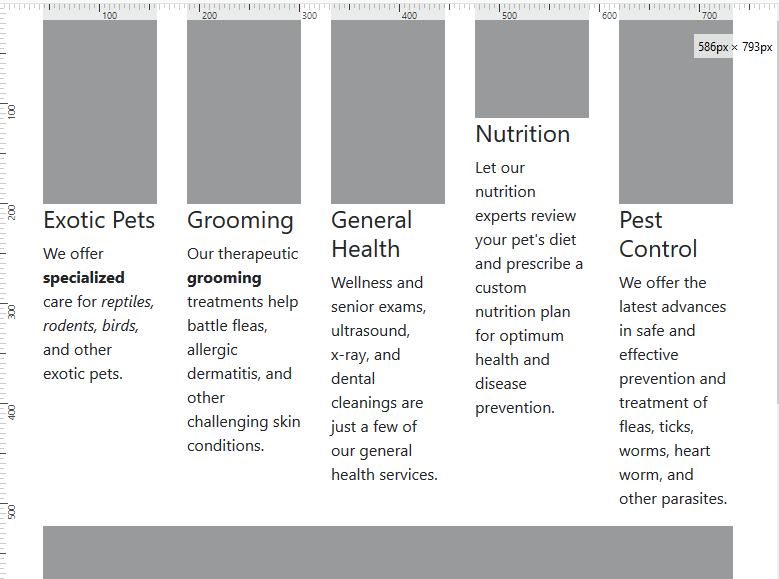


* + Si se usa solo ***col*** bootstrap intentará cuadrar eso en el layout.
  + Si se usa un breakpoint, ***col-sm***, significa que la columna tomará el ancho entero del navegador hasta que alcance el breakpoint de ***sm*** el cual es a 576px o mayor. Así sera hasta el próximo breakpoint.
  + Si se usa el número, ***col-md-6***. Eso indica que la columna tomará todo el ancho del navegador hasta el medium breakpoint, 768 px, el cual en este punto la columna tomara 6 de las 12 unidades.

Ejemplos Gráficos de Columnas

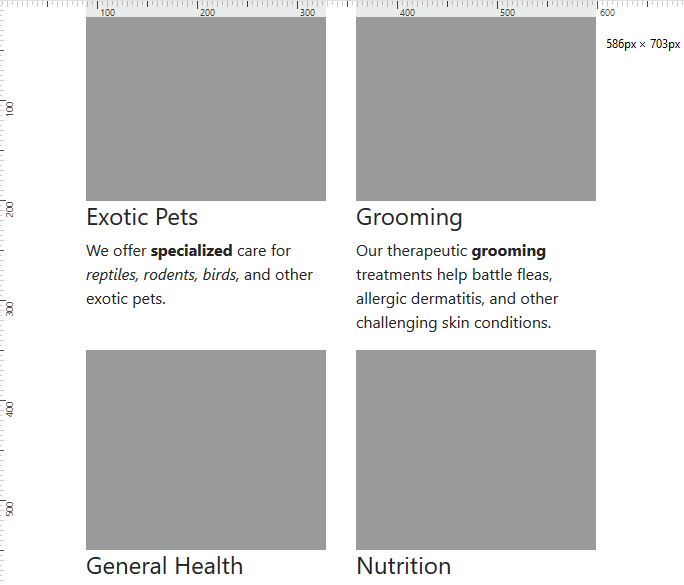
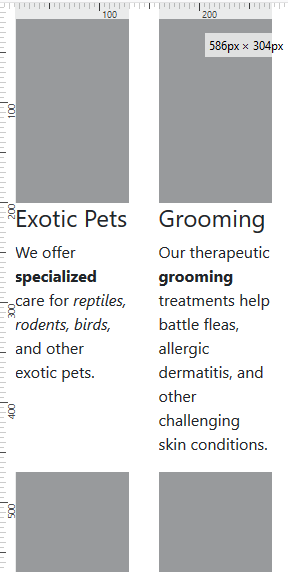
Este es un ejemplo de una página que se encuentra en los ejercicios de la carpeta 03\_03. Para tener en cuenta son 6 sections en el ejemplo.

* Si se usa la clase de *col* únicamente, lo que hará es que *bootstrap* intentara acomodar los elementos como sea por defecto de acuerdo al viewport. En el siguiente ejemplo, se muestra como todos los *section* tienen la clase solo de *col* y se van acomodando a medida que cambia el tamaño del viewport lo más responsive que pueden.

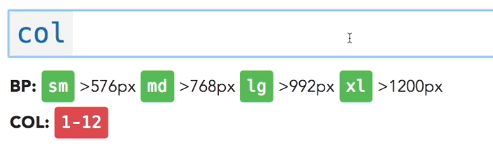


* Por otra parte si se usa la clase de *col-#* (en este caso será 6), este elemento tomará estos espacios en todos los tamaños del viewport.





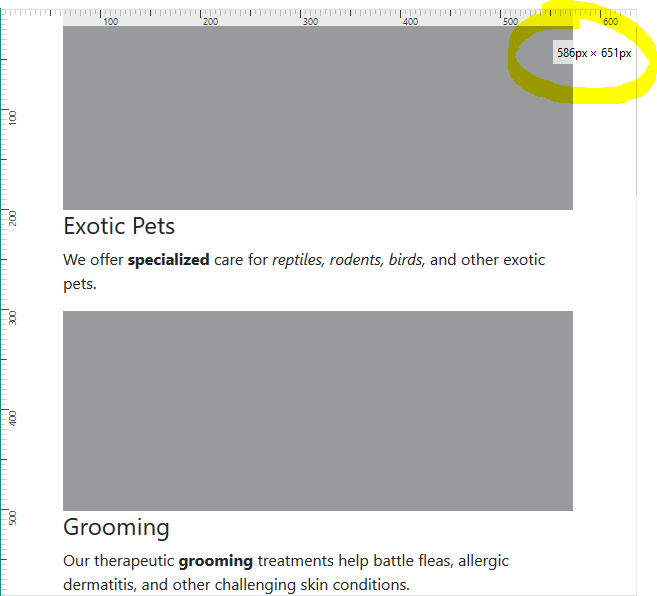
* Por último, si se establece un *breakpoint* las columnas cambiarán de acuerdo a este *breakpoint*.



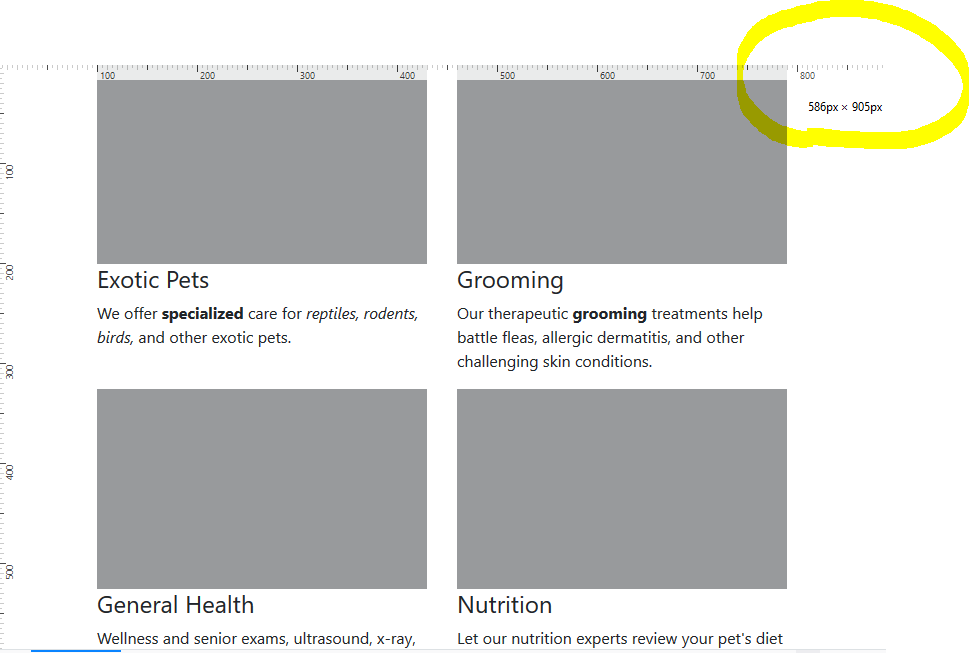
Por ejemplo, si se indica el valor a *col-md-6* esto indicara que cualquier valor menor a **768px** ocupará el total del ancho del *viewport.* Una vez llegue al *breakpoint* de **768px**, cada elemento tomará 6 espacios del grid.



En esta imagen, el tamaño del ancho está en 651 px, cual ya paso el *breakpoint de sm (small).* Sin embargo, como se menciono, **no cambiará** hasta que pase el *breakpoint de md (medium)*



Por último, una vez pasa el *breakpoint* que definimos, comenzará a realizar los cambios de ahí a cualquier valor mayor de ocupar cada elemento 6 espacios del grid. Claro está que si indicamos más *breakpoints* estos actuarán también.

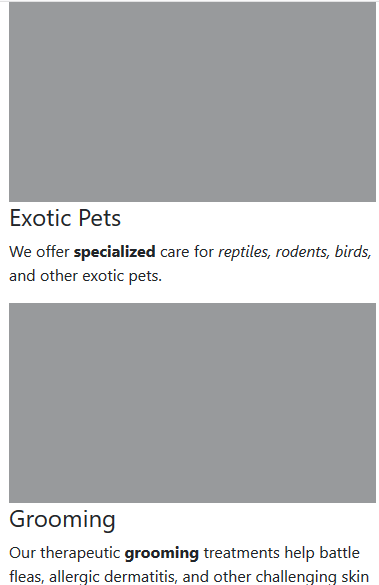


Multiples Breakpoints de columnas

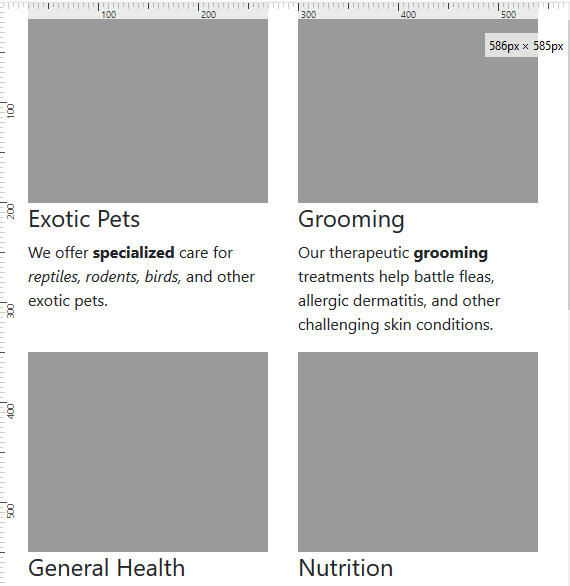
Como se mencionaba anteriormente, también se puede indicar distintos breakpoints en un elemento para asi aprovechar mejor el potencial de Bootstrap. El siguiente ejemplo, es una representación gráfica de todos los breakpoints:



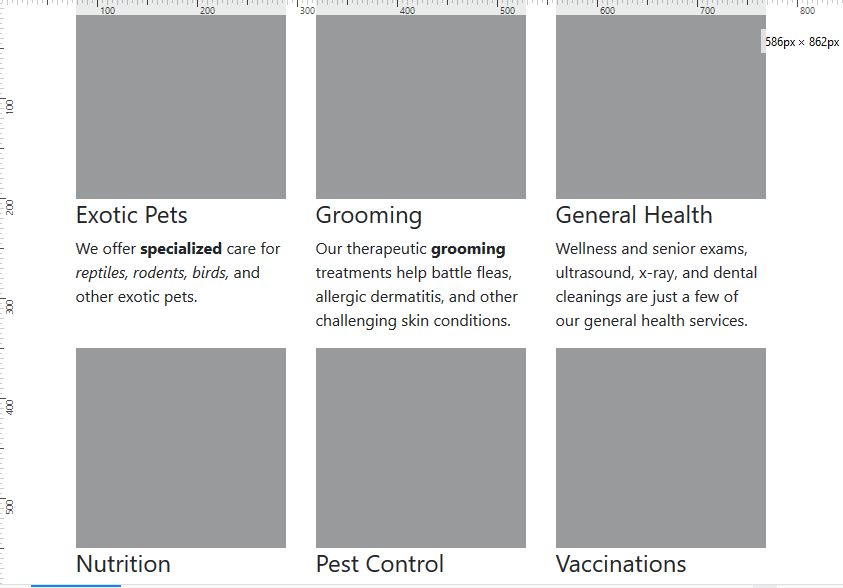
**Breakpoint menor a 576 px (Extra small)**



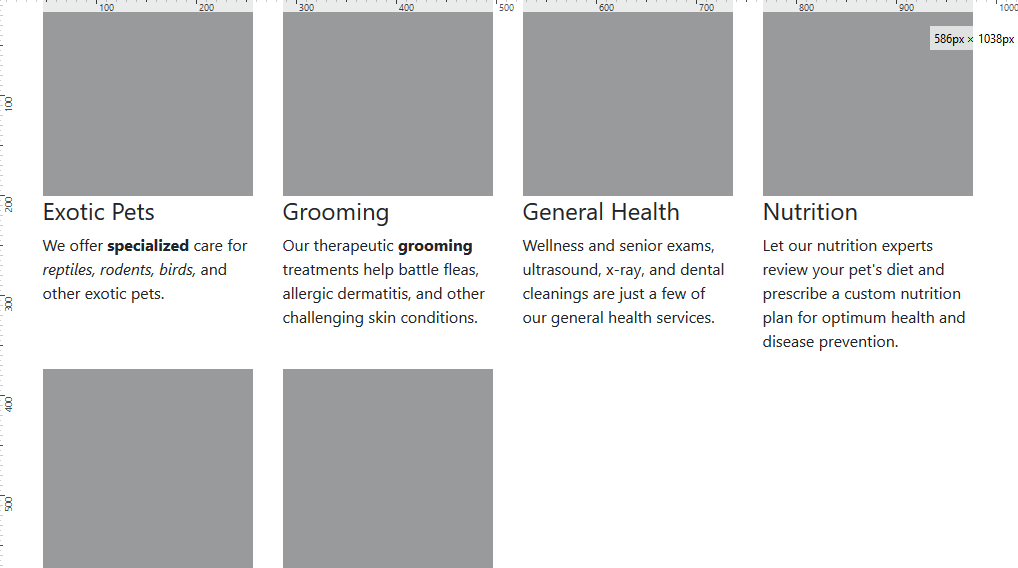
**Breakpoint mayor o igual a 576px (small)**



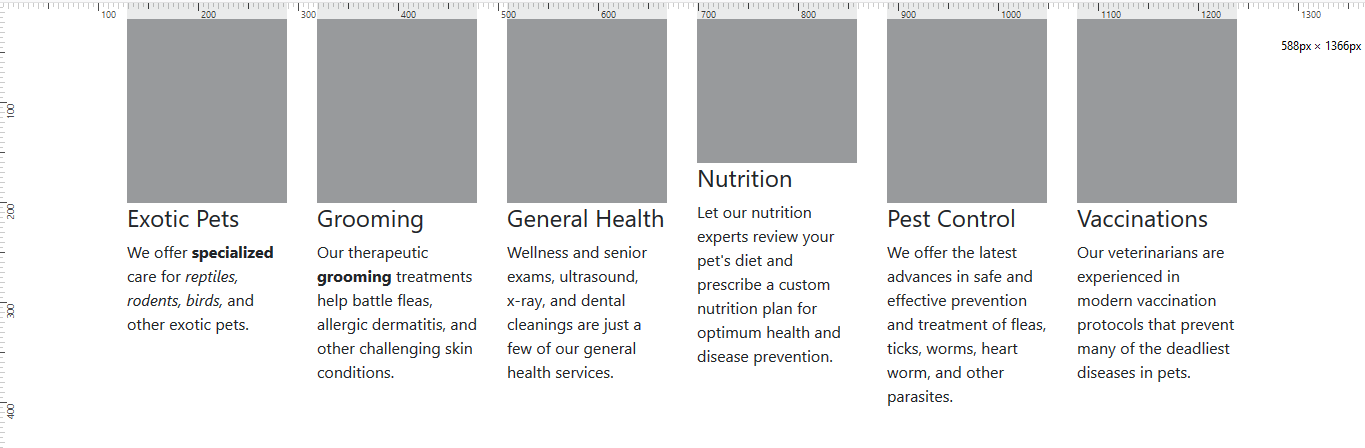
**Brekpoint mayor o igual a 768px (medium)**



**Breakpoint mayor o igual a 992px (large)**



**Breakpoint mayor o igual a 1200px (extra large)**



Offset de columnas

Sirve para dejar espacios vacios del grid. Sirve para simular una columna sin necesidad de crear una vacia.



Aunque el *offset* ya no se debe usar para centrar los elementos como en versiones pasadas (En bootstrap 4 hay mejores formas: mirar alineamientos) es útil igual saber como funciona:



Se puede ver la diferencia con y sin el *offset* aplicado.

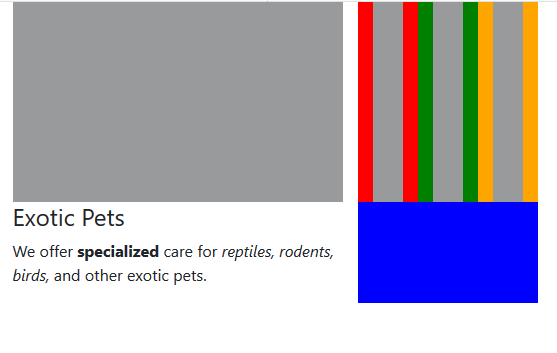


Columnas nesteadas

* Se debe crear una nueva fila *<row>* dentro de una columna, eso crea un grid de 12 columnas
* Se usan las mismas clases.

El color **azul** representa la sección que ocupa 4 espacios. Los otros colores hacen parte de la fila nueva dentro de esta columna y que tomarán cada uno 4 espacios dentro de la columna.

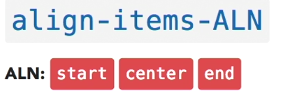




Alineación

**¡Bootstrap 4 usa FLEXBOX!**

* Alineación vertical
  + **Usar en filas**
  + Las clases se definen de la siguiente forma:



* + - Start: Al principio
    - Center: En la mitad
    - End: Al fina
* Alineación individual
  + **Se usa en las columnas**
  + Las clases se definen de la siguiente forma:

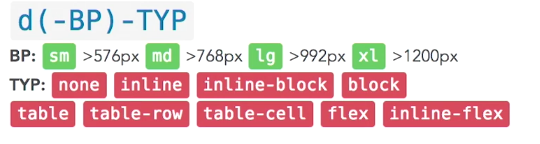


* Alineación horizontal
  + **Se usa en las filas**
  + Se necesita ancho de col
  + Las clases se definen de la siguiente forma:



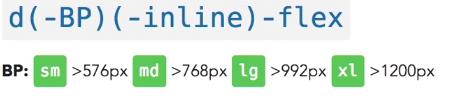
Position/Display

* Position
  + Fixed-top, fixed-bottom, sticky-top (funcionan como CSS)
* Display
  + Imitan CSS
  + Se permite controlar diferentes tipos de display en distintos viewports
  + Se usa de la siguiente manera:



* El container Flex basico

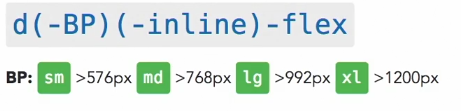
Por defecto flexbox es en *block.*



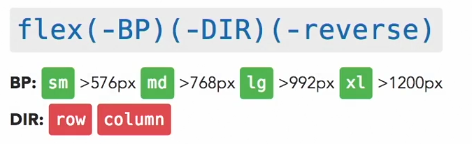
Opciones del contender de Flex

**Archivo 03\_10**

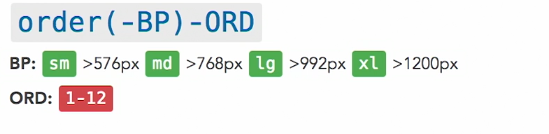
* Clases de items y contenedores
* Son *block* por defecto por eso se puede usar *inline*



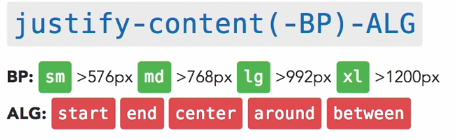
* También existen las clases de dirección. Es para decirle al navegador para decirle que los elementos se muestren como filas o columnas



* En adición, se puede contralar el orden de los elementos.



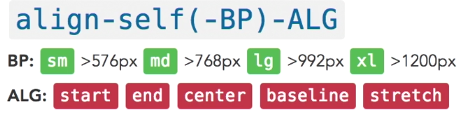
* Además, se puede especificar justificación para los items



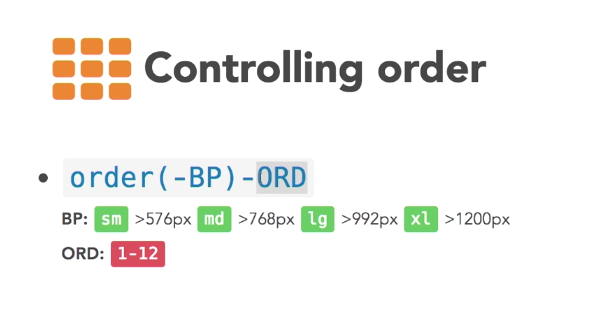
Elementos individuales flex

También se pueden controlar elementos

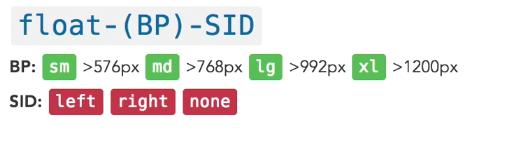
* Se usa al clase *align-self.* Hacen que se emparejen las alineaciones con el elemento



* También se puede especificar el orden de los elementos

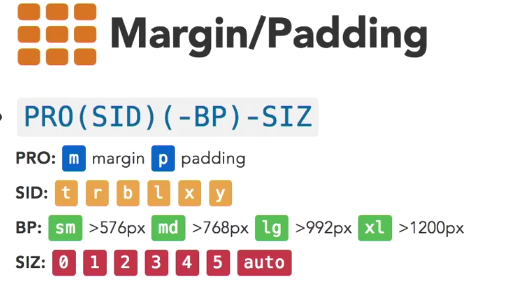


Elementos flotantes



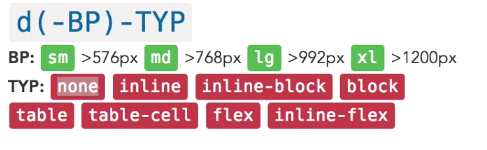
Existe una clase *clearfix* que ayuda a que el contenedor no pierda rastreo de los elementos. Esto se puede hacer en CSS

Margin y padding

**Pro:** propiedad. **Side:** lado. .**Size:** Esta definido en REMS. 3=1 REM

Visibilidad

* Dos clases: Invisible(toma el espacio y lo retiene) – visible
* Display también existe:

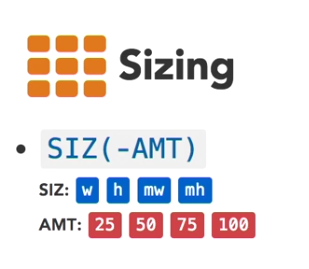


Si se coloca *none* desaparece el elemento.

Utilidades de tamaños

Sirve para controlar el ancho y alto de los elementos

Los elementos **AMT** están en porcentajes.



Bordes



# Capítulo 4. Usando componentes Navs y Navbar

* **Nav:** Son los padres de toda la navegación
* **Tabs y pills:** Ayudan crear contenido que cambian cuando hacen click
* **Navbar:** Se usan para la navegación principal entre páginas

Creación de navs

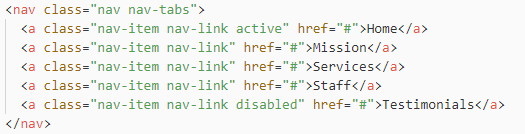
Ha cambiado mucho con la nueva implementación de Flexbox en Boostrap 4.

Para empezar a usar una navegación en Bootstrap, se debe definir las siguientes clases para las etiquetas: **<*ul>, <li> y <a>***





**También se puede hacer de otra manera:** Usando la etiqueta de <nav> y por cada item referenciando un elemento <a> directamente donde se colocaran las clases restantes como se ve:



* Dos clases importantes son ***active y disabled.*** Estos están en el link de *home y testimonials* del ejemplo. El *active* indica el elemento en que se encuentra actualmente y el *disabled* un elemento que no se puede acceder. En las siguientes imágenes se podran apreciar estos.
* Dos tipos de estilos definidos para la barra de navegación son las clases: ***nav-pills y nav-tabs***
  + Nav-pills





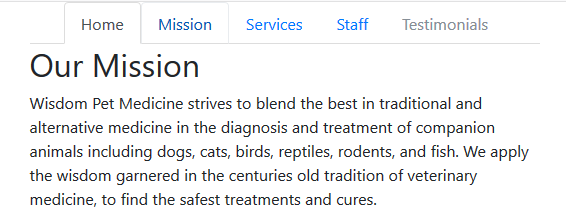
* + Nav-tabs



También se puede realizar alineación para los *tabs o pills* usando las siguientes clases:

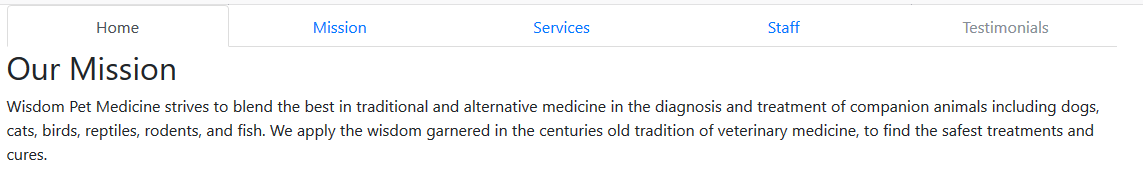
* *Justify-content-center:* Los tabs en este caso quedan en la mitad. Tambien se puede decir *end* para ponerlo a la izquierda



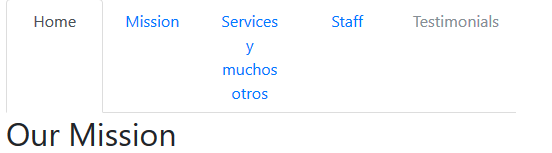


* *Nav-justified:* Pone todo en el mismo tamaño.

<nav class="nav nav-tabs nav-justified">

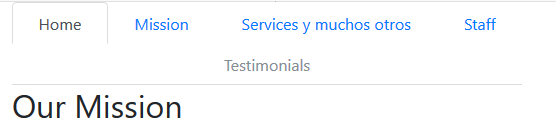


Sin embargo, si tiene un texto amplio, forza al aumentar el tamaño.



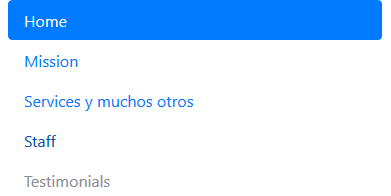
Eso se puede solucionar en el siguiente punto

* *Nav-fill*: Trata de mantener los demás elementos el mismo tamaño mientras le da más espacio al item necesari



* *Flex-column:* Esto sirve para controlar cuando los elementos se ponen unos encima de otros (como para el viewport de un celular) *y* hacerlos más responsive. Se puede especificar el *breakpoint* para que vuelva a una barra normal:

<nav class="nav nav-pills flex-column flex-sm-row">



Componente Navbar

Como se habia mencionado antes, navbar representa la navegación principal entre páginas. El siguiente es el código del ejemplo:

<nav class="navbar bg-dark navbar-dark navbar-expand-sm">

<div class="container">

<div class="navbar-nav">

<a class="nav-item nav-link active" href="#">Home</a>

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

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

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

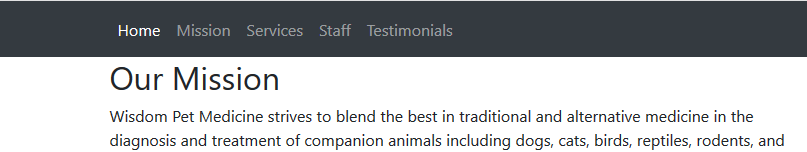
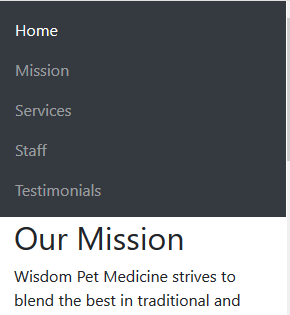
<a class="nav-item nav-link" href="#">Testimonials</a>

</div>

</div>

</nav>

La clase *navbar* debe ser definida dentro de la etiqueta <nav> junto opcionalmente clases de estilo como *bg-dark* que define el *background* de la navegación y el estilo de este *navbar-dark* para las letras y otras cosas. La clase *navbar-expand-sm* es para que la barra deje de ser *responsive* en cierto *breakpoint*



Navbar Brand y Texto

Añadir el logo a la barra o un texto inline dentro de la navegación. Colocandolo antes o despues del resto de la navegación afectará su ubicación. Se puede usar con una etiqueta de div, a, o img.

<nav class="navbar bg-dark navbar-dark navbar-expand-sm">

<div class="container">

<div class="navbar-brand">Wisdom Pet Medicine</div>

<div class="navbar-nav">

<a class="nav-item nav-link active" href="#">Home</a>

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

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

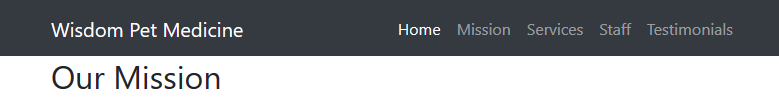
<a class="nav-item nav-link" href="#">Staff</a>

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

</div>

</div>

</nav>



Para una imagen tipo svg:

<img src="images/wisdompetlogo.svg" class="navbar-brand" style="width:40px;"></img>

* Para añadir un texto adicional:

<nav class="navbar bg-dark navbar-dark navbar-expand-sm">

<div class="container">

<img src="images/wisdompetlogo.svg" class="navbar-brand" style="width:40px;"></img>

<div class="navbar-nav">

<a class="nav-item nav-link active" href="#">Home</a>

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

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

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

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

</div>

<span class="navbar-text">Text</span>

</div>

</nav>

Dropdowns:

Sirve para crear listas dentro del menu. Se necesita basicamente las clases de:

* Dropdown: Definida está clase en un div, se colocará adentro el elemento <a> que hará la función de contener el dropdown
* Dropdown-toggle: ira como clase del elemento <a> mencionado.
* Dropdown-menu: Crear un div con esta clase debajo del <a>
* Dropdown-item: Valores de <a> que irán dentro del menu definido:

<nav id="navbar-site" class="navbar navbar-dark bg-dark navbar-expand-sm">

<div class="container">

<div class="navbar-nav">

<a class="nav-item nav-link" href="#mission">Mission</a>

<div class="dropdown">

<a class="nav-item nav-link dropdown-toggle" href="#" data-toggle="dropdown" >Services</a>

<div class="dropdown-menu" >

<a class="dropdown-item" href="#">Grooming</a>

<a class="dropdown-item" href="#">Hola</a>

</div>

</div>

<a class="nav-item nav-link" href="#staff">Staff</a>

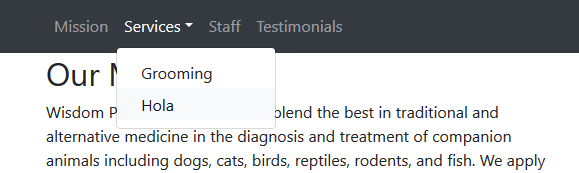
<a class="nav-item nav-link" href="#testimonials">Testimonials</a>

</div>*<!-- navbar-nav -->*

</div>*<!-- container -->*

</nav>

Resultado:



Form elements

Aunque más adelante se explicará en más detalle, se implementará estos elementos para crear un formulario sencillo que hará la función de busqueda dentro de menu.

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

<input class="form-control mr-2" type="text" placeholder="Search">

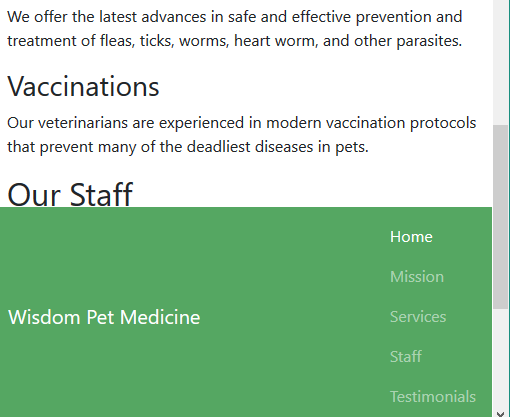
<button class="btn btn-outline-light" type="submit">Go</button>

</form>



Posicionamiento

Para poner el menu en una posición fija del *viewport*.



Contenido colapsable

Generamente usado en un *navbar*. Los puntos a tener en cuenta son:



El menu de *hamburguer* tiene otras clases adicionales que se verán en el ejemplo:

* Como primer medida se añade al clasle de collapse y navbar-collapse dentro de una etiqueta de *div* y añadiendo a su vez un *id* que será referenciado por el botón definido.
* Dentro del boton se debe añadir la clase *navbar-toggler* para indicar que este sera el toggler y por último los data-attribute definidos:

<nav class="navbar navbar-dark bg-dark navbar-expand-sm">

<div class="container">

<a href="#" class="navbar-brand">Wisdom Pet Medicine</a>

<button class="navbar-toggler" type="button"

data-toggle="collapse" data-target="#myTogglerNav"

aria-controls="myTogglerNav" aria-expanded="false"

aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

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

<div class="navbar-nav">

<a class="nav-item nav-link active" href="#">Home</a>

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

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

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

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

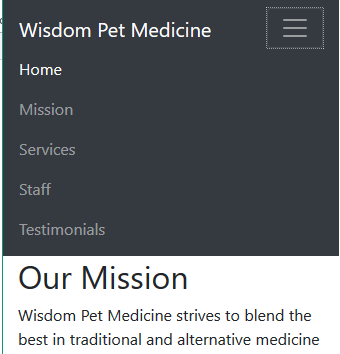
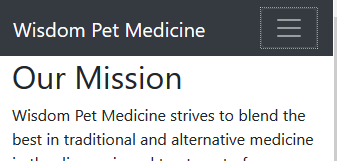
</div>*<!-- navbar -->*

</div>

</div>*<!-- container -->*

</nav>*<!-- nav -->*

**El resultado es:**



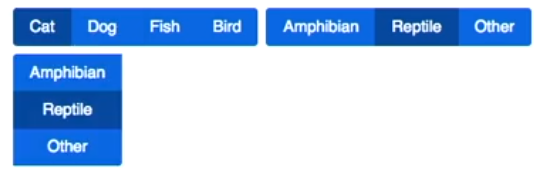
# Capítulo 5. Overview de elemento de estilos

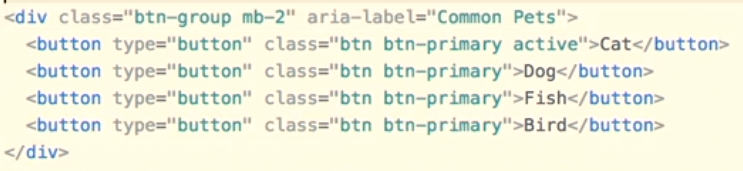
Botones

Se comienza con la clase **btn,** añadiend una clase adicional de tamaño o color. Las opciones para los botones son:



Grupos de botones





Badges

Usualmente van un elemento *span*



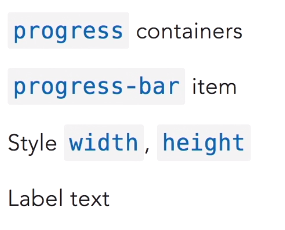
**Ejemplo**:





Progress Bar

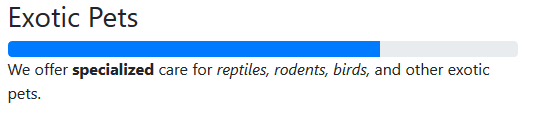
Se necesita:



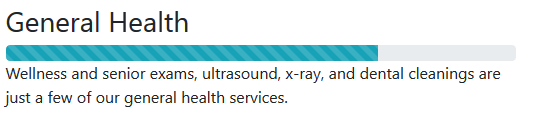
<div class="progress">

<div class="progress-bar" style="width: 73%"></div>

</div>



Usando la clase <div class="progress-bar bg-info progress-bar-striped" style="width: 73%"></div> Se obtiene una barra *striped*



Pueden ser animados usando JS para cargar o movimiento.