

PROGRAMACIÓN WEB

teoría/práctica



INTRODUCCIÓN A FRAMEWORKS DE FRONTEND: BOOTSTRAP



Frameworks de frontend

- ▶ Recordemos el módulo en el que hablábamos de "Frontend y Backend" (ver currícula Introducción al concepto de Frontend y Backend)
- ▶ Mencionábamos que el frontend incluye a todas las tecnologías que se utilizan para construir lo que el usuario verá cuando ingresa a un sitio web.
- ▶ Entre esas tecnologías teníamos: HTML, CSS y Javascript.
- ▶ Ahora...que es un framework? La traducción nos indica que es un "marco de trabajo". Entonces podemos decir que un framework de frontend, es un marco de trabajo para las tecnologías de frontend.
- ▶ Que incluye este marco de trabajo? Bueno, incluye como hacer determinadas cosas sin necesidad de crear algo desde cero... algo así como: para que reinventar la rueda, si ya está inventada?



Frameworks de frontend

- ▶ Un marco de trabajo de frontend nos facilita la escritura de código, permitiéndonos utilizar código previamente creado y estructurado para una mejor organización general.
- ▶ Dentro de la categoría frameworks de frontend podemos encontrar muchas opciones, todo depende en que aspecto nos queremos enfocar.
- ▶ Algunos frameworks más conocidos son:



Frameworks de frontend: Bootstrap

- ▶ Nosotras trabajaremos con BOOTSTRAP.
- ▶ Bootstrap es el framework más popular de HTML, CSS y JAVASCRIPT para desarrollar de forma "responsive" proyectos web.
- ▶ Que significa **responsive**? Significa que es adaptable a cualquier dispositivo! Por ejemplo, un sitio web es responsive si se ve correctamente y se ajusta a: una tablet, una notebook, un celu, una compu o cualquier otro Dispositivo.
- ▶ Esta currícula está basada en la guía oficial de Bootstrap que la podés Encontrar acá:

<http://getbootstrap.com/css/>



Bootstrap

- ▶ Fué creado por el equipo de programadores y diseñadores de Twitter, lanzado en el 2011
- ▶ Es un framework utilizado en la gran mayoría de sitios web actuales
- ▶ Es gratuito y lo podemos obtener ingresando a: <http://getbootstrap.com/>
- ▶ Utiliza componentes HTML, CSS y JAVASCRIPT definidos que luego veremos en detalle
- ▶ Los "creadores" de Bootstrap lo actualizan frecuentemente, para que todos podamos tener acceso a un framework cada vez más potente y con más elementos que podamos utilizar



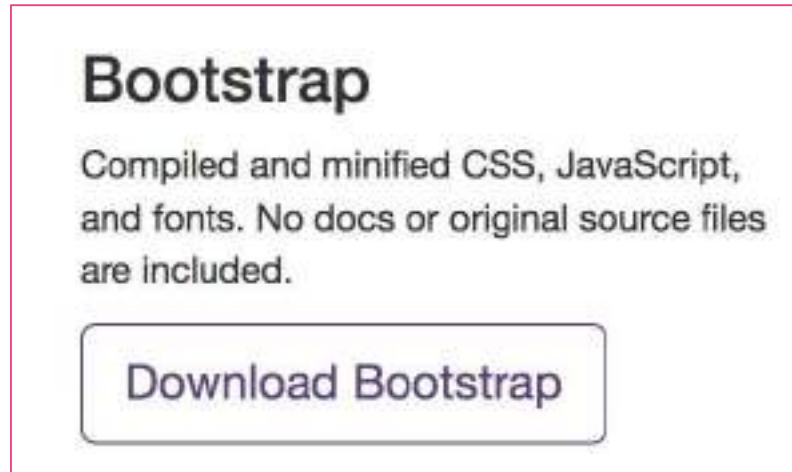
Bootstrap

- ▶ Ejemplos de sitios web que utilizan Bootstrap:
 - ▶ <https://www.lyft.com/>
 - ▶ <http://www.vogue.mx/?international>
- ▶ Para poder utilizar Bootstrap, deberíamos contar con conocimientos mínimos de HTML, CSS y JAVASCRIPT.
- ▶ También ya tendríamos que tener instalado un editor de texto (recordemos que Atom es uno de los recomendados)
- ▶ En esta currícula aprenderemos a:
 - ▶ Cómo utilizar el framework
 - ▶ Estructura de bootstrap
 - ▶ Elementos



Descarga de Bootstrap

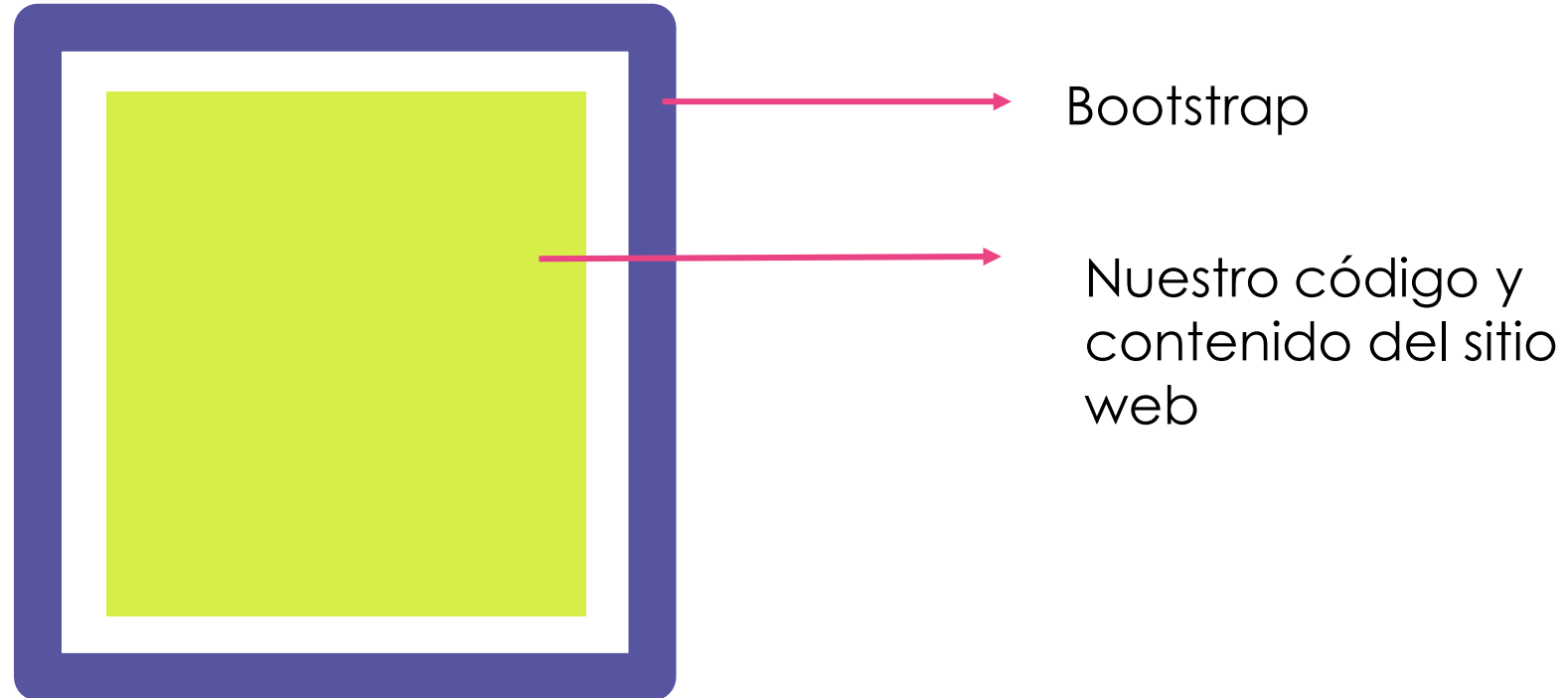
- ▶ Para descargar ingresamos a: <http://getbootstrap.com/getting-started/>
- ▶ Luego hacemos click en esta parte:
- ▶ Guardamos y descomprimos el .zip en una carpeta dedicada a nuestro proyecto
- ▶ Adentro de este .zip vamos a encontrarnos con:



Nombre	
▶	css
▶	fonts
▶	js

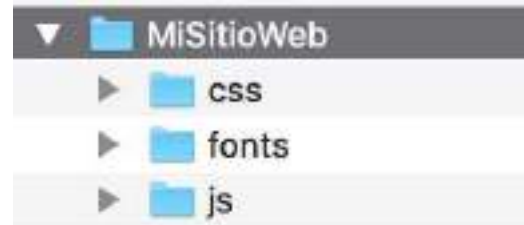
Utilizando Bootstrap

- ▶ Esas 3 carpetas que contiene el .zip: CSS, JS y fonts... es *el framework en si*. Esos archivos son los que nos "proveen" del marco de trabajo



Utilizando Bootstrap

- ▶ Lo que tenemos que hacer es que esas 3 carpetas que contenía el .zip sean parte de nuestro proyecto.
- ▶ Por ejemplo, si en nuestro proyectos tenemos una carpeta que se llama "Mi sitio web", dentro de ésta podríamos ubicar las carpetas de la siguiente forma:



- ▶ Una vez que tenemos cada cosa en su lugar, viene la parte de crear nuestro index.html . Siempre utilizamos el nombre "index" para el archivo HTML principal :)



Utilizando Bootstrap

- ▶ Si ya tenemos nuestro archivo index.html entonces vamos a escribir cierta etiqueta html para "llamar" a Bootstrap y poder utilizarlo en nuestra página web

```
<!-- Bootstrap -->
```

```
<link href="css/bootstrap.min.css" rel="stylesheet">
```

- ▶ Como podemos ver en la etiqueta <link> estamos llamando al archivo que está dentro de la carpeta css (siguiendo el ejemplo, dentro de nuestra carpeta MiSitioWeb).
- ▶ Esto es necesario para que nuestros elementos de HTML que incorporemos puedan ser elementos de Bootstrap (tener su diseño, ser responsive y adoptar un comportamiento predefinido)

jQuery



- ▶ Algo particular que tendríamos que tener en cuenta es el uso de jQuery.
- ▶ Técnicamente, jQuery es una librería de Javascript, también se le llama framework de frontend, pero que es realmente?
- ▶ jQuery es una forma de hacer las mismas acciones que Javascript pero de forma aún más simplificada
- ▶ jQuery deriva de Javascript por lo que ambos pueden convivir en un mismo archivo. Esto quiere decir que en un archivo .js puedo escribir con Javascript y también combinarlo con jQuery. O crear simplemente un archivo .js y solo utilizar jQuery.
- ▶ En estas instancias no vamos a aprender a utilizar jQuery, pero si es importante que sepamos que existe, que sepamos que Bootstrap lo utiliza y que debemos agregar una etiqueta html en nuestro index llamando al lenguaje.



jQuery

- ▶ En el archivo index.html vamos a tener que agregar las siguientes líneas para que todos los componentes de Bootstrap puedan funcionar correctamente:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>  
<script src="js/bootstrap.min.js"></script>
```

- ▶ Recordemos que el atributo de HTML "src" indica la fuente desde donde se está obteniendo el recurso. En este caso el recurso es un archivo de jQuery que nos permite contar con esta librería para que algunos elementos de Bootstrap puedan funcionar.
- ▶ Otra opción es descargar el archivo jquery.min.js y ubicarlo en la carpeta de nuestro proyecto...

Index.html

- ▶ Resumiendo, nuestro index.html debería quedar así:
- ▶ Las etiquetas <meta> contienen información que hace posible que nuestra página sea vista desde cualquier dispositivo
- ▶ Notemos que dentro de <body> llamamos a los archivos jQuery, esto es porque necesitamos primero que se cargue la página primero y luego jQuery

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7
8     <title>Utilizando Bootstrap</title>
9
10    <!-- Bootstrap -->
11    <link href="css/bootstrap.min.css" rel="stylesheet">
12
13    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
14    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
15    <!--[if lt IE 9]>
16      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
17      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
18    <![endif]-->
19  </head>
20  <body>
21    <h1>hola Mundo!</h1>
22
23    <!-- llamada a jQuery -->
24    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
25    <script src="js/bootstrap.min.js"></script>
26  </body>
27 </html>
```



Contenedores (containers)

- ▶ Bootstrap considera que cada elemento de nuestra web debe estar insertado dentro de un contenedor...
- ▶ Que es un contenedor? No es más que un elemento HTML `<div>` que contiene una clase "container" de CSS y esa clase es adaptable según el dispositivo desde donde se observe la página.

- ▶ Uso de containers:

```
<div class="container-fluid">  
  ...  
</div>
```

```
<div class="container">  
  ...  
</div>
```

- ▶ Donde la clase "container-fluid" es para cuando queremos que el contenedor ocupe todo el ancho del dispositivo por el cual se mire la página



Grillas (Grids)

- ▶ Una de las ventajas que posee Bootstrap es que tiene clases de CSS predefinidas para que funcionen como una especie de "grilla" para poder ordenar y organizar la página y el contenido
- ▶ Las filas deben estar siempre incluidas dentro de un container. La clase para crear una fila es ".row"

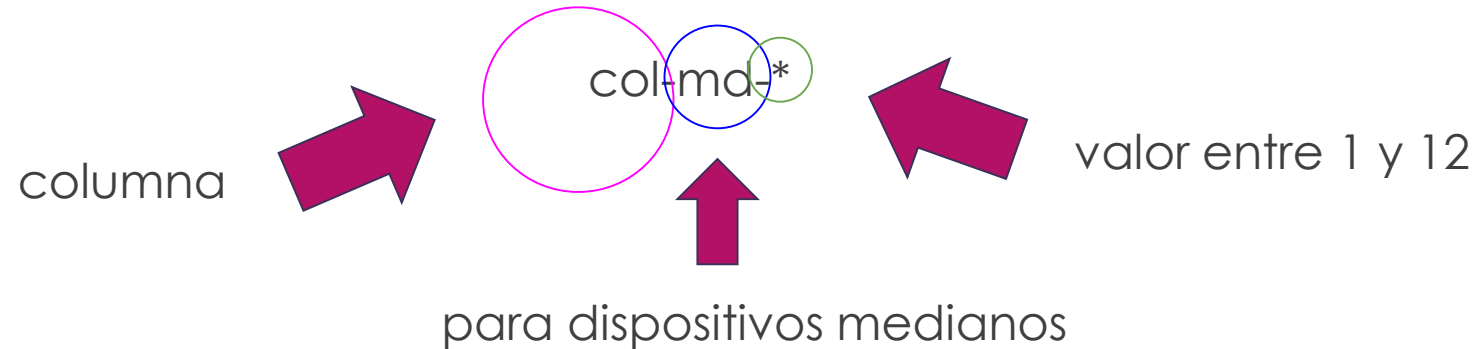
```
<div class="container-fluid">  
  <div class="row">  
    ...  
  </div>  
</div>
```

- ▶ La grilla se centra en filas y columnas de distintos tamaños. Las columnas se definen a partir de las filas...por ejemplo esta fila tiene dos columnas:

```
<div class="row">  
  <div class="col-md-8">.col-md-8</div>  
  <div class="col-md-4">.col-md-4</div>  
</div>
```

Grillas (Grids)

- ▶ Que significa la clase de CSS que usamos recién, "col-md-8"?



- ▶ Se utilizan valores entre 1 y 12 para definir el tamaño de las columnas, lo veremos más adelante con ejemplos.
- ▶ md es una de varias medidas que se utiliza para cuando se achica la página al mismo tamaño del de un dispositivo mediano. Si achicamos la página, la información debe reordenarse para que se ajuste a la pantalla.

Grillas (Grids)

- ▶ Se podrá tener un total de 12 columnas por filas que ocupen todo el ancho de una pantalla, pero que pasa cuando la pantalla por la que estamos viendo la página es más chica? Entonces en vez de mostrarse todas las columnas, van a mostrarse menos y de forma ordenada...

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

- ▶ Tamaño posible de pantalla (xs, sm, md, lg)
- ▶ Estas columnas de las que hablamos no son necesariamente tablas, simplemente simulamos la organización de una tabla para el contenido de mi página (que puede ser texto, imágenes, etc)



Grillas (Grids)

- Por ejemplo, miremos la página de chicasprogramadoras, desde nuestra compu, probablemente se vería así :





Grillas (Grids)

- ▶ Y si achicamos la página desde la compu, o la vemos desde el celu o una tablet? Los elementos se irán reordenando.
- ▶ Esto es porque estarán ubicados (texto, foto, logo, etc) dentro de contenedores, filas y columnas.
- ▶ Observemos la siguiente tabla:

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-





Grillas (Grids)

- ▶ Según la tabla, si queremos que nuestra página se adapte a dispositivos que tengan MENOS DE 768 píxeles de ancho, entonces usaremos: col-xs-*
- ▶ Si queremos que se adapte a dispositivos que sean MAYOR O IGUAL A 768 píxeles, entonces usaremos: col-sm-*
- ▶ Si queremos que se adapte a dispositivos que sean MAYOR O IGUAL A 992 píxeles entonces usaremos: col-md-*
- ▶ Para dispositivos que sean MAYOR O IGUAL A 1200 píxeles, usaremos: col-lg-*
- ▶ Y si queremos que para el menor tamaño de pantalla disponible (col-xs) se muestre una columna de menor tamaño: col-xs-1, o si queremos que ocupe la mitad de la pantalla: col-xs-6, y así sucesivamente

Ejemplo de Grilla

- ▶ En el ejemplo vamos a poder ver como organizamos la información que va a tener nuestra web en columnas, recordando que las filas y columnas no necesariamente implica que sea una tabla
- ▶ Este código vas a tenerlo disponible, solicitá a tu mentora que te pase el link de descarga

```
20 <body>
21
22 <div class="container">
23   <h1>hola Mundo!</h1>
24
25   <div class="row">
26     <div class="col-md-1">col 1</div>
27     <div class="col-md-1">col 2</div>
28     <div class="col-md-1">col 3</div>
29     <div class="col-md-1">col 4</div>
30     <div class="col-md-1">col 5</div>
31     <div class="col-md-1">col 6</div>
32     <div class="col-md-1">col 7</div>
33     <div class="col-md-1">col 8</div>
34     <div class="col-md-1">col 9</div>
35     <div class="col-md-1">col 10</div>
36     <div class="col-md-1">col 11</div>
37     <div class="col-md-1">col 12</div>
38   </div>
39   <div class="row">
40     <div class="col-md-8">col 1</div>
41     <div class="col-md-4">col 2</div>
42   </div>
43   <div class="row">
44     <div class="col-md-4">col 1</div>
45     <div class="col-md-4">col 2</div>
46     <div class="col-md-4">col 3</div>
47   </div>
48   <div class="row">
49     <div class="col-md-6">col 1</div>
50     <div class="col-md-6">col 2</div>
51   </div>
52 </div>
```



Ejemplo de Grilla

► Y el resultado es:

hola Mundo!

col 1
col 1
col 1
col 1

col 2

col 3

col 4

col 5

col 2

col 6

col 7

col 2

col 8

col 9
col 2
col 3

col 10

col 11

col 12

Ejemplo de Grilla

- ▶ Y que pasaría si achicamos la ventana de nuestro navegador simulando tener una pantalla más chica? El resultado sería:
- ▶ Por que se mostraría 1 columna por fila? Porque eso fue lo que escribimos en el código, Para un tamaño mediano (md)
- ▶ Ahora si cambiamos md por xs, y si achicamos la pantalla, se vería así:

hola Mundo!

col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9	col 10	col 11	col 12
col 1								col 2			
col 1				col 2				col 3			
col 1						col 2					

hola Mundo!

col 1
col 2
col 3
col 4
col 5
col 6
col 7
col 8
col 9
col 10
col 11
col 12
col 1
col 2
col 1
col 2
col 3
col 1
col 2



Alineamiento

- Si estamos trabajando con texto dentro de nuestra página, Bootstrap nos provee formas automática para alinearlos. Vayamos probando en nuestro index.html :

```
<p class="text-left">Left aligned text.</p>  
<p class="text-center">Center aligned text.</p>  
<p class="text-right">Right aligned text.</p>  
<p class="text-justify">Justified text.</p>  
<p class="text-nowrap">No wrap text.</p>
```

- Y se vería:

Left aligned text.

Center aligned text.

Right aligned text.

Justified text.

No wrap text.



Tablas

- ▶ Una opción muy potente de Bootstrap es la utilización de tablas. Solamente tendríamos que agregar la clase "table" y la estructura de la tabla en sí.
- ▶ Tenemos variantes para crear distintos tipos de tablas, podemos poner clases como: ".table-bordered", ".table-striped", "table-hover"
- ▶ Podemos utilizar colores para las filas, en general para remarcar algún error, alguna advertencia o simplemente para destacar. Utilizamos las clases:
- ▶ Para que nuestras tablas sean "responsive", solamente agregamos:

```
<div class="table-responsive">  
  <table class="table">  
    ...  
  </table>  
</div>
```

```
<!-- On rows -->  
<tr class="active">...</tr>  
<tr class="success">...</tr>  
<tr class="warning">...</tr>  
<tr class="danger">...</tr>  
<tr class="info">...</tr>  
  
<!-- On cells (`td` or `th`) -->  
<tr>  
  <td class="active">...</td>  
  <td class="success">...</td>  
  <td class="warning">...</td>  
  <td class="danger">...</td>  
  <td class="info">...</td>  
</tr>
```



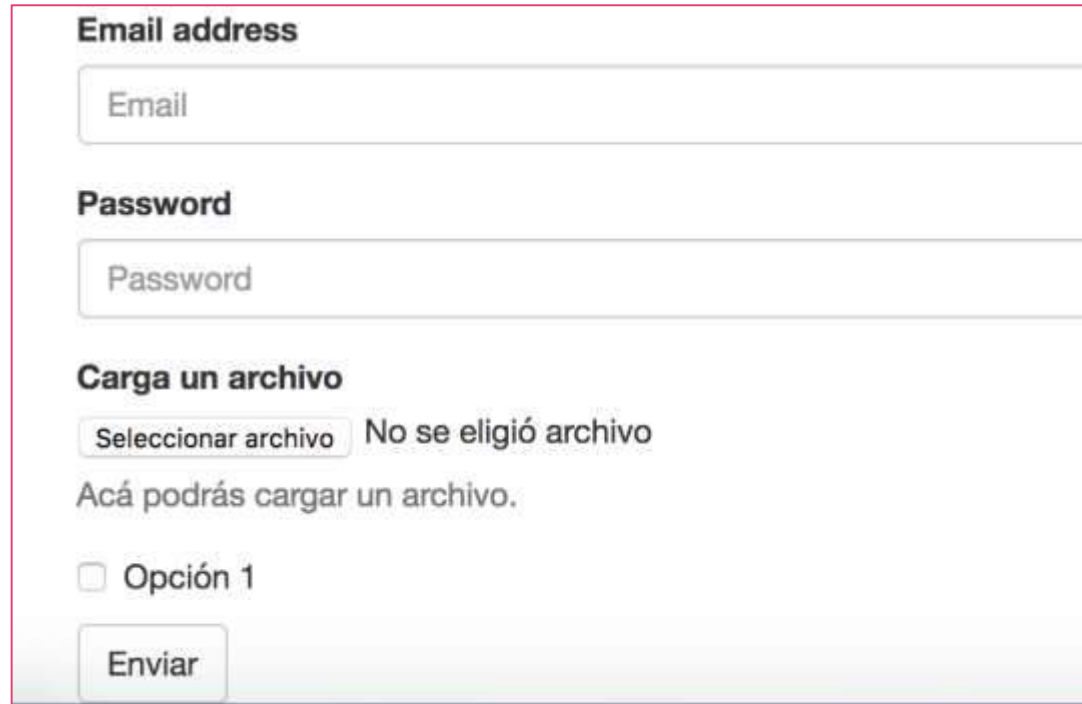
Formularios

- ▶ Recordemos que un formulario se utiliza para que los usuarios manden información a nuestros servidores con la intención de dejar sus datos, hacernos consultas y demás.
- ▶ Prestá atención a las clases que se utilizan:

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" placeholder="Email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Formularios

- ▶ Esas clases permiten que el formulario tenga el siguiente formato:

A screenshot of a web form with a light gray background and rounded corners. It contains three sections: "Email address" with a text input field containing the placeholder "Email"; "Password" with a text input field containing the placeholder "Password"; and "Carga un archivo" which includes a button labeled "Seleccionar archivo", the text "No se eligió archivo", the instruction "Acá podrás cargar un archivo.", a checkbox labeled "Opción 1", and a button labeled "Enviar".

Email address

Password

Carga un archivo

No se eligió archivo

Acá podrás cargar un archivo.

☐ Opción 1

- ▶ También permiten que cuando nos posicionamos en un input (por ejemplo: email, aparezca el efecto celeste que resalta el input.



Validación de Formularios

- ▶ Es muy importante que a medida que el usuario va completando los campos de nuestro formulario, podamos avisarle en tiempo real si ha completado algo de forma incorrecta o si le ha faltado algún dato.
- ▶ Este proceso se llama validación, y Bootstrap es especialista :)
- ▶ En el ejemplo se muestra como van a lucir los campos en caso de éxito, error y advertencia, tanto para inputs como para checkboxes.
- ▶ El uso de estas clases se realiza dinámicamente, es decir a medida que vamos ingresando la información. Utilizar este tipo de formulario es un poco complejo por lo que veremos más adelante como hacer para validar en tiempo real los datos.



Tamaños de input

- Podemos optar por inputs de tamaño estándar, grandes o inclusive pequeños.

- Ejemplos:

- Y las clases que definen los tamaños:

```
<input class="form-control input-lg" type="text" placeholder=".input-lg">  
<input class="form-control" type="text" placeholder="Default input">  
<input class="form-control input-sm" type="text" placeholder=".input-sm">
```

```
<select class="form-control input-lg">...</select>  
<select class="form-control">...</select>  
<select class="form-control input-sm">...</select>
```



Botones

- Tenemos diseños de botones predefinidos, con efectos:



```
<a class="btn btn-default" href="#" role="button">Link</a>  
<button class="btn btn-default" type="submit">Button</button>  
<input class="btn btn-default" type="button" value="Input">  
<input class="btn btn-default" type="submit" value="Submit">
```

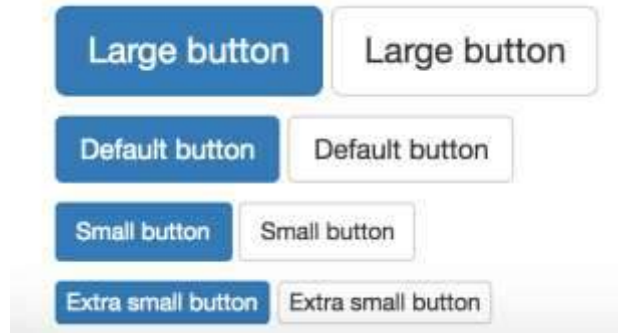
- Con estilos:

```
<button type="button" class="btn btn-default">Default</button>  
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-info">Info</button>  
<button type="button" class="btn btn-warning">Warning</button>  
<button type="button" class="btn btn-danger">Danger</button>  
<button type="button" class="btn btn-link">Link</button>
```



Tamaño botones

- ▶ Así como podemos regular el tamaño de los inputs, podemos cambiar el tamaño a los botones:



```
<p>
  <button type="button" class="btn btn-primary btn-lg">Large button</button>
  <button type="button" class="btn btn-default btn-lg">Large button</button>
</p>
<p>
  <button type="button" class="btn btn-primary">Default button</button>
  <button type="button" class="btn btn-default">Default button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-sm">Small button</button>
  <button type="button" class="btn btn-default btn-sm">Small button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-xs">Extra small button</button>
  <button type="button" class="btn btn-default btn-xs">Extra small button</button>
</p>
```

Imágenes

- ▶ Nuestras imágenes también pueden ser responsive! Para esto solo agregamos la clase a nuestra etiqueta `<img>` ".image-responsive"
- ▶ También podemos hacer que nuestras imágenes sean circulares, rectangulares o cuadradas

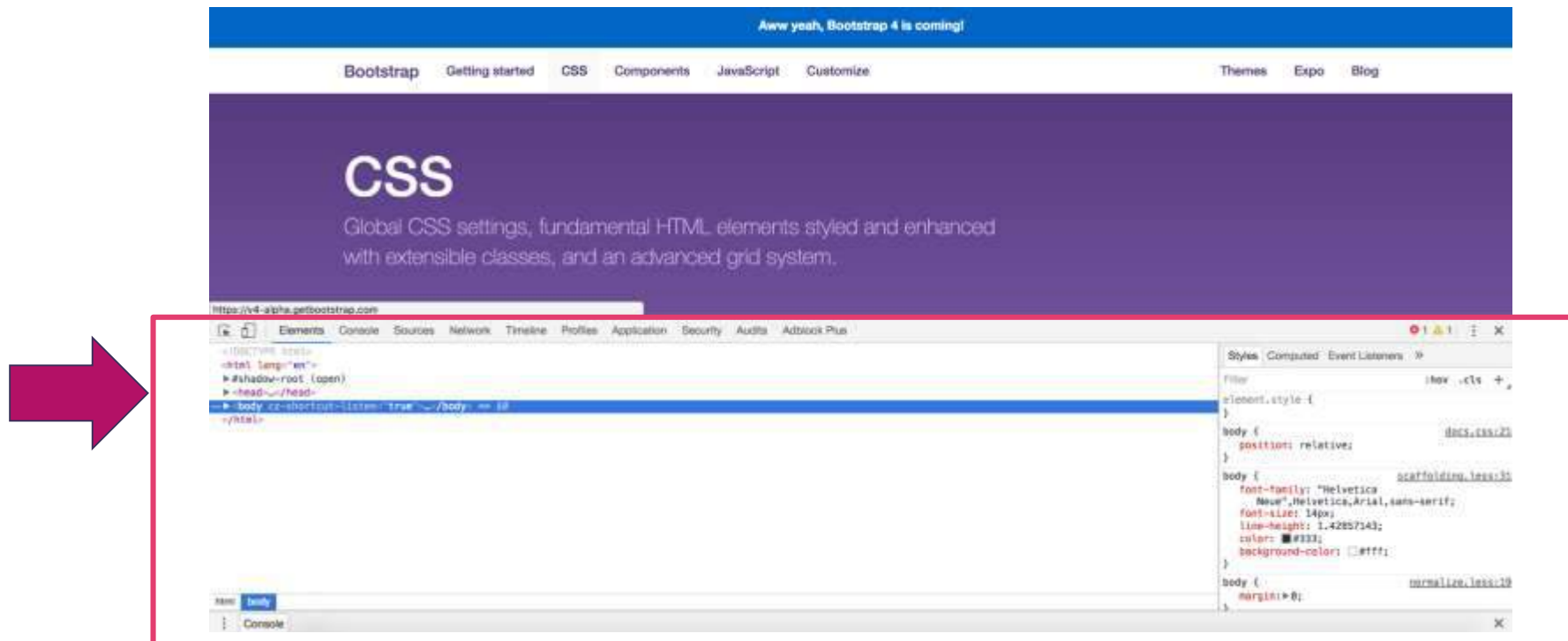
```
  
  

```



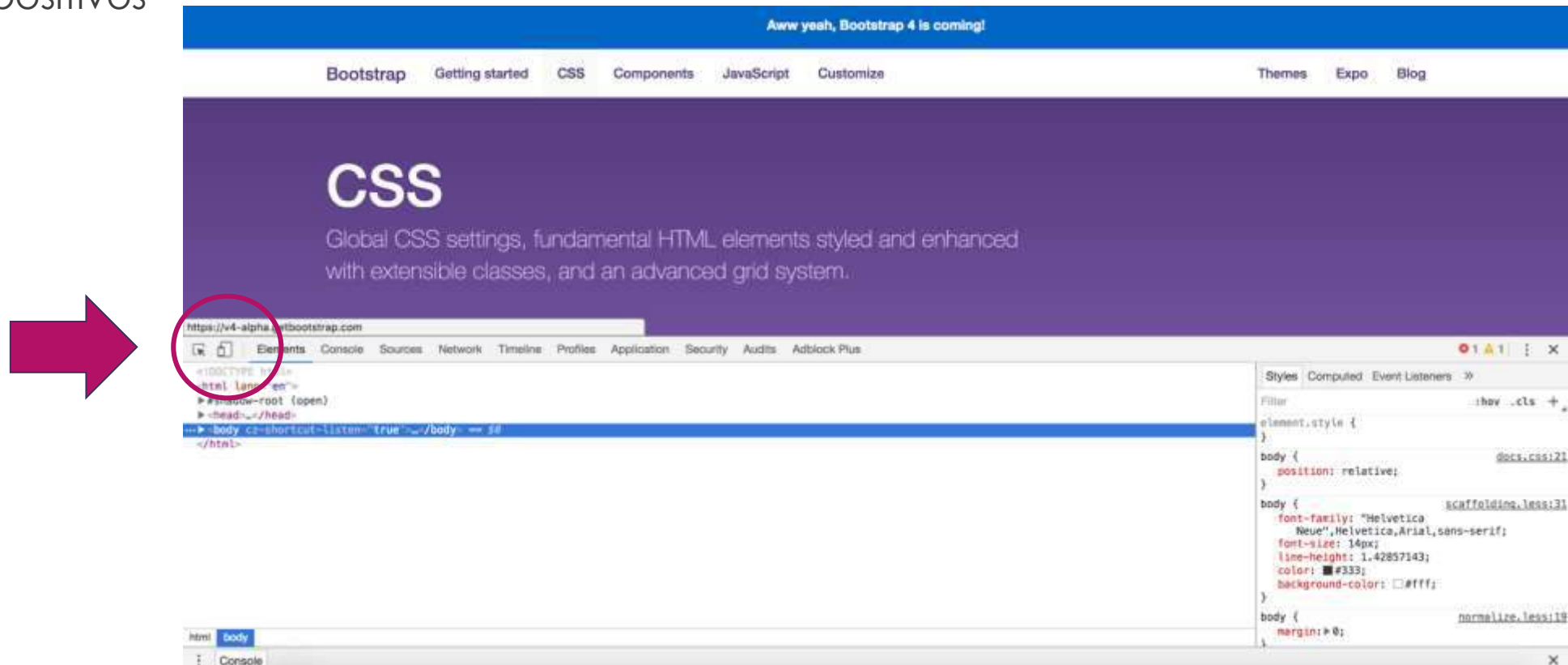
Responsive

- ▶ Tenemos varias opciones a la hora de probar el código que vamos ingresando.
- ▶ Una opción es desde tu navegador preferido, apretar F12 O bien, sobre la misma pantalla click derecho, inspeccionar.
- ▶ Nos debería aparecer una ventana en la parte inferior o en el costado de la página:



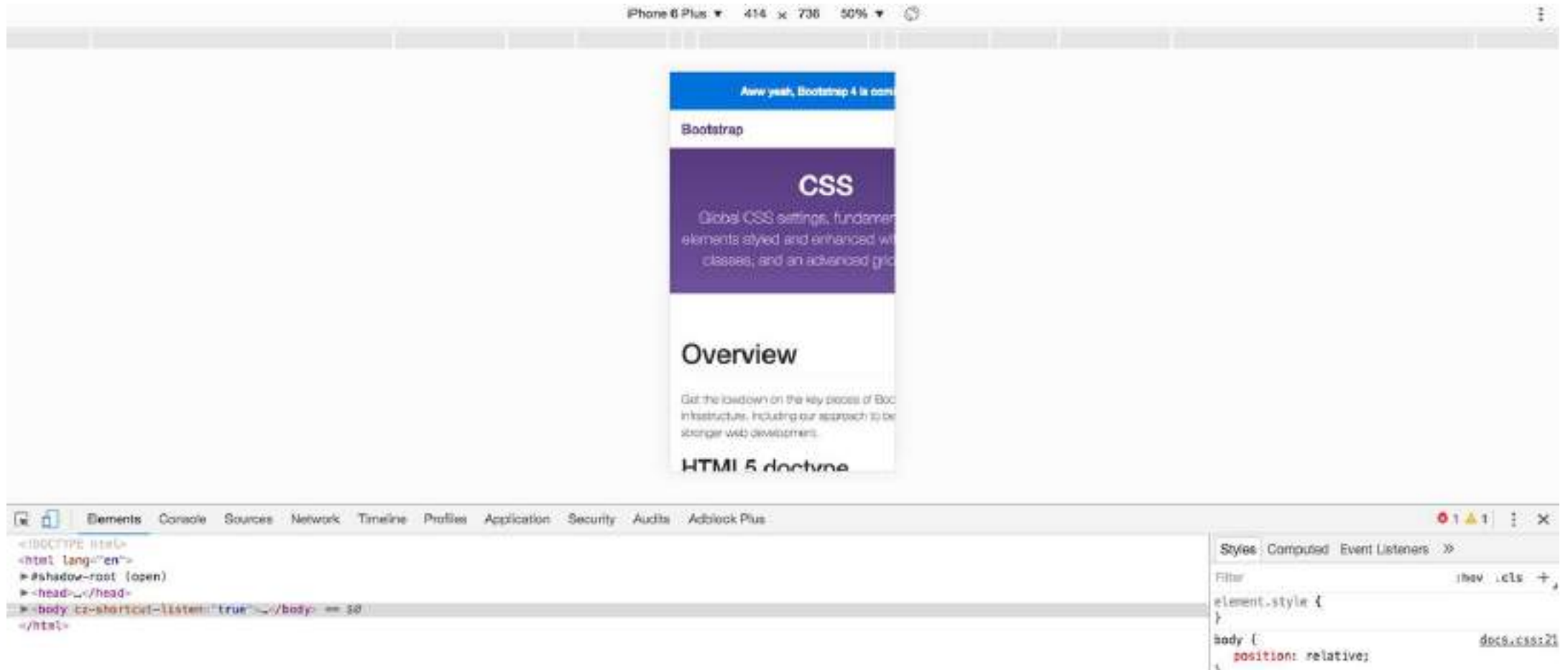
Responsive

- El inspeccionador es una herramienta que nos provee el navegador que nos ayuda a programar. Más adelante veremos en detalle de que se trata, por ahora solo nos interesa encontrar el ícono de un celular, que nos va a permitir visualizar nuestra web según diferentes dispositivos



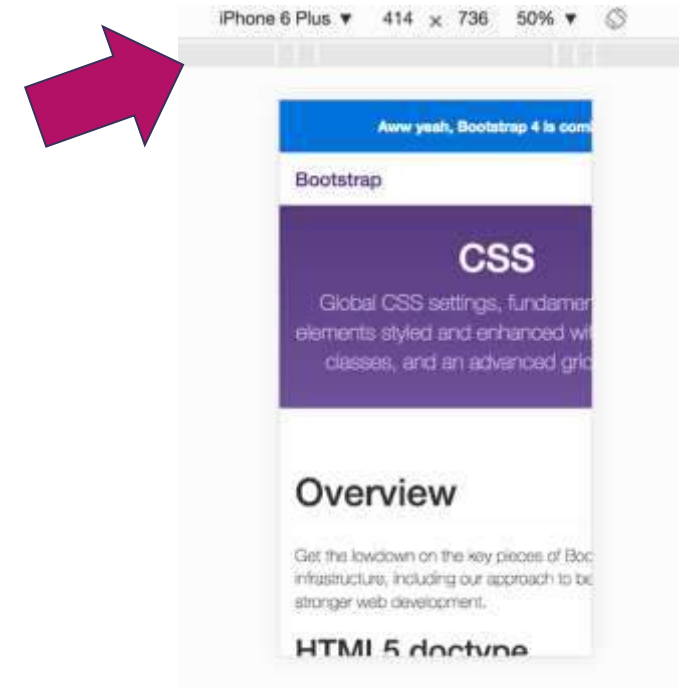
Responsive

- Hacemos click en el celular y ahora nuestra página se vería así:



Responsive

- ▶ En la parte superior elegimos el dispositivo, para probar como se vería nuestra web:
- ▶ También podemos elegir rotar el teléfono
- ▶ Esta es una opción para probar y ver si estamos Utilizando Bootstrap de la manera correcta
- ▶ De lo contrario podemos optar por sitios como: www.responsinator.com que únicamente servirían Si nuestro sitio ya se encuentra online.





Desafío

- ▶ Bootstrap tiene aún más opciones que podemos explorar, recordá que podés ingresar a: <http://getbootstrap.com/css/> para aprender a fondo esta herramienta.
- ▶ El desafío ahora es replicar lo que dimos en el tutorial, ir fijándote en la web oficial de Bootstrap e ir explorando las distintas opciones que tenés para tu página.
- ▶ Modificando el tamaño de la ventana de tu navegador podés ir probando como va quedando y sino utilizando el inspeccionador de código que te provee tu navegador

Buena suerte!



¿Preguntas?



MUCHAS GRACIAS