



# Introducción a la programación web- Bootstrap



# Introducción a Bootstrap

Bootstrap es un framework de CSS que facilita y estandariza el desarrollo de sitios web.

Está diseñado siguiendo la estrategia mobile first, por lo tanto todos los tamaños y dimensiones están pensadas para los dispositivos móviles. Para tamaños más grandes lo que se hace es adaptar el contenido.

El sistema de grillas de Bootstrap se basa en la disposición del contenido de nuestra web dentro de grillas flexibles, las cuales se van a ir adaptando de forma automática dependiendo del tamaño de la pantalla.

El contenido se distribuye en **12 columnas** y entre cada columna existe una **calle/espaciado de 24px**.



## Sistema de grillas





## Sistema de grillas

Las columnas se agrupan dentro de filas identificadas con la clase **row**. Estas filas se deben colocar dentro de una etiqueta contenedora identificada con la clase **container** (se va adaptando dejando un margen distinto en cada resolución) o **container-fluid** (ocupa el 100% del ancho en todas las resoluciones).

El tamaño de las columnas se especificará con clases css que Bootstrap define para cada tamaño de pantalla, por ejemplo **.col-XX**, donde **XX** es el tamaño de la columna, que podrá tomar valores entre **1 y 12**.



## Sistema de grillas

En la siguiente tabla se muestra un resumen del sistema de grillas de Bootstrap, su comportamiento según el tamaño del dispositivo y las clases CSS que nos permiten controlarlo.

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px

## Sistema de grillas

Comportamiento del container según cada corte de pantalla de bootstrap:

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

# Sistema de grillas

## Ejemplo

```
<div class="container">
  <div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-md-4">.col-md-4</div>
  </div>
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4">.col-md-4</div>
  </div>
  <div class="row">
    <div class="col-md-6">.col-md-6</div>
    <div class="col-md-6">.col-md-6</div>
  </div>
</div>
```

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

Dado que las columnas se han especificado únicamente mediante las clases **.col-md-\*** esto pasará sólo en las pantallas **≥768px**. En pantallas más pequeñas ocuparán el ancho total y se mostrarán una debajo de la otra.



## Columnas de ancho automático

A partir de la versión 4 de Bootstrap podemos utilizar las columnas de ancho automático, es decir, indicar únicamente el prefijo de clase de columna que queremos y el sistema calculará automáticamente su ancho.

Para esto podremos usar la clase **".col"**, sin número de columnas ni tamaño de pantalla.





## Columnas de ancho automático

```
<div class="container">
  <div class="row">
    <div class="col">1 of 2</div>
    <div class="col">2 of 2</div>
  </div>
  <div class="row">
    <div class="col">1 of 3</div>
    <div class="col">2 of 3</div>
    <div class="col">3 of 3</div>
  </div>
</div>
```

1 of 2	2 of 2	
1 of 3	2 of 3	3 of 3

El número de columnas del ejemplo se mantendrá igual para todos los tamaños de pantalla, adaptando el ancho de las columnas para cada uno de ellos.

A esta clase **podemos añadir el sufijo para establecer el tamaño de pantalla**, por lo que disponemos de las clases `.col`, `.col-sm`, `.col-md`, `.col-lg`, `.col-xl`.



## Columnas de ancho automático

Estas columnas de ancho automático se pueden mezclar en una misma fila con las columnas de ancho específico.

```
<div class="container">
  <div class="row">
    <div class="col">1 of 3</div>
    <div class="col-6">2 of 3 (wider)</div>
    <div class="col">3 of 3</div>
  </div>
  <div class="row">
    <div class="col-5">1 of 3 (wider)</div>
    <div class="col">2 of 3</div>
    <div class="col">3 of 3</div>
  </div>
</div>
```

1 of 3	2 of 3 (wider)	3 of 3
1 of 3 (wider)	2 of 3	3 of 3



## Columnas de ancho variable

Estas columnas ocupan el ancho justo que se necesite según el contenido de la columna.

Para utilizarlas disponemos de las clases **".col-\*-auto"**, donde \* puede ser cualquiera de los sufijos de tamaño de pantalla que hemos visto antes **sm, md, lg, xl, xxl o ningún sufijo (.col-auto)** para indicar todos los tamaños.



# Columnas de ancho variable

## Ejemplo

```
<div class="container">
  <div class="row">
    <div class="col col-lg-2">1 of 3</div>
    <div class="col col-md-auto">Variable width content</div>
    <div class="col col-lg-2">3 of 3</div>
  </div>
  <div class="row">
    <div class="col">1 of 3</div>
    <div class="col col-md-auto">Variable width content</div>
    <div class="col col-lg-2">3 of 3</div>
  </div>
</div>
```

1 of 3	Variable width content	3 of 3
--------	------------------------	--------

1 of 3	Variable width content	3 of 3
--------	------------------------	--------



## Anidamiento de columnas

Para anidar columnas solamente **tenemos que crear una nueva fila** (un nuevo **row**) dentro de una columna, y esta nueva fila podremos subdividirla usando también hasta 12 columnas.

## Anidamiento de columnas

```
<div class="row">  
  <div class="col col-sm-9">  
    Level 1: .col-sm-9  
    <div class="row">  
      <div class="col-8 col-sm-6">Level 2:  
.col-8 .col-sm-6</div>  
      <div class="col-4 col-sm-6">Level 2:  
.col-4 .col-sm-6</div>  
    </div>  
  </div>  
</div>
```

Level 1: .col-sm-9	
Level 2: .col-8 .col-sm-6	Level 2: .col-4 .col-sm-6



## Desplazamiento de columnas

Es posible mover o desplazar una columna hacia la derecha, añadiendo un offset inicial mediante las clases: **.offset-\***.

Por ejemplo `.offset-4` creará un espacio a la izquierda de la columna de tamaño 4 (como si se creara una columna oculta de tipo `.col-4`).

**También podemos especificar el offset según el tamaño de pantalla.**

## Desplazamiento de columnas

```
<div class="row">
  <div class="col col-md-4">.col-md-4</div>
  <div class="col col-md-4 offset-md-4">.col-md-4
.offset-md-4</div>
</div>
<div class="row">
  <div class="col col-md-3 offset-md-3">.col-md-3
.offset-md-3</div>
  <div class="col col-md-3 offset-md-3">.col-md-3
.offset-md-3</div>
</div>
<div class="row">
  <div class="col col-md-6 offset-md-3">.col-md-6
.offset-md-3</div>
</div>
```



Si en algún caso necesitamos **eliminar el offset** podemos utilizar el **tamaño cero (0)**. Por ejemplo, si especificamos un offset de 2 para tamaños pequeños y no queremos que dicho offset se aplique para pantallas medianas ni grandes tendríamos que hacer:

```
<div class="col col-sm-5 offset-sm-2 col-md-7 offset-
md-0">...</div>
```





## Márgenes

También existen los márgenes para crear espacios variables entre columnas, para un margen automático del lado izquierdo se utiliza la clase **.ms-auto**, **.me-auto** para un margen del lado derecho y **.mx-auto** para un margen de ambos lados. En este último caso estaríamos centrando la columna.

# Márgenes

```
<div class="row">
  <div class="col col-md-4">.col-md-4</div>
  <div class="col col-md-4 ms-auto">.col-md-4
.ms-auto</div>
</div>
<div class="row">
  <div class="col col-md-3 ms-md-auto">.col-md-3
.ms-md-auto</div>
  <div class="col col-md-3 ms-md-auto">.col-md-3
.ms-md-auto</div>
</div>
<div class="row">
  <div class="col-auto me-auto">.col-auto
.me-auto</div>
  <div class="col-auto">.col-auto</div>
</div>
```

.col-md-4

.col-md-3 .ms-md-  
auto

.col-auto .me-auto

.col-md-4 .ms-auto

.col-md-3 .ms-md-  
auto

.col-auto



## Alineación vertical

Para indicar la misma alineación para todos los elementos de una fila tendremos que utilizar la clase **".align-items-\*"**, donde "\*" podrá ser **"start"** (al principio o pegada a la parte superior de la fila), **"center"** (alineación centrada en vertical) o **"end"** (alineación pegada al final o a la parte inferior de la fila).

# Alineación vertical

```
<div class="container">
  <div class="row align-items-start">
    <div class="col">One of three columns</div>
    <div class="col">One of three columns</div>
    <div class="col">One of three columns</div>
  </div>
  <div class="row align-items-center">
    <div class="col">One of three columns</div>
    <div class="col">One of three columns</div>
    <div class="col">One of three columns</div>
  </div>
  <div class="row align-items-end">
    <div class="col">One of three columns</div>
    <div class="col">One of three columns</div>
    <div class="col">One of three columns</div>
  </div>
</div>
```

One of three columns	One of three columns	One of three columns
One of three columns	One of three columns	One of three columns
One of three columns	One of three columns	One of three columns

# Alineación vertical

Si queremos indicar por separado la **alineación vertical de cada una de las columnas de una fila**, tendremos que usar la clase CSS **.align-self-\***, donde "\*" podrá adoptar los mismos valores: **start, center o end**.

```
<div class="container">
  <div class="row">
    <div class="col align-self-start">One of three
  columns</div>
    <div class="col align-self-center">One of three
  columns</div>
    <div class="col align-self-end">One of three
  columns</div>
  </div>
</div>
```



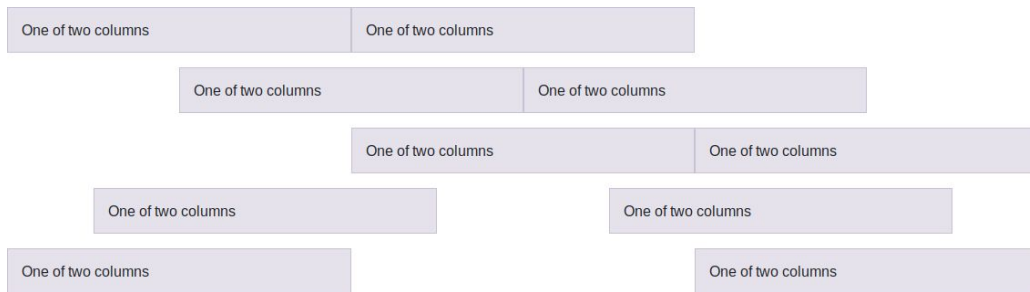


## Alineación horizontal

Para esto disponemos de la clase **".justify-content-\*"**, donde **"\*"** podrá ser **"start"** (izquierda), **"center"** (centrado), **"end"** (derecha), **"around"** (añadirá el mismo espacio a ambos lados de la columna) y **"between"** (añade espacio entre las columnas).

# Alineación horizontal

```
<div class="container">
  <div class="row justify-content-start">
    <div class="col-4">One of two columns</div>
    <div class="col-4">One of two columns</div>
  </div>
  <div class="row justify-content-center">
    <div class="col-4">One of two columns</div>
    <div class="col-4">One of two columns</div>
  </div>
  <div class="row justify-content-end">
    <div class="col-4">One of two columns</div>
    <div class="col-4">One of two columns</div>
  </div>
  <div class="row justify-content-around">
    <div class="col-4">One of two columns</div>
    <div class="col-4">One of two columns</div>
  </div>
  <div class="row justify-content-between">
    <div class="col-4">One of two columns</div>
    <div class="col-4">One of two columns</div>
  </div>
</div>
```



**En caso de que lo necesitemos podremos añadir también el tamaño de pantalla a las distintas clases de alineación.**



## Como incluir Bootstrap

Usando el servicio de BootstrapCDN, puedes incluir Bootstrap en tus proyectos añadiendo la siguiente línea en el head de todas tus páginas:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
```





## Como incluir Bootstrap

Y los siguientes JS al final del archivo antes del cierre de la etiqueta body:

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCNj/IqJ95wo16oMtfS KbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script>
```

### Importante

Es conveniente siempre chequear la última versión de todos los CDN en la web oficial: <https://getbootstrap.com/>



## Como incluir Bootstrap

Por otro parte, es importante que se encuentre la etiqueta **<meta>** en el head para que los contenidos se escalen correctamente en los distintos dispositivos:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```



# Media queries



## Media queries

En la mayoría de los casos gracias a todas las clases que provee Bootstrap nos será suficiente para componer nuestra web.

Sin embargo, en algunas situaciones es posible que queramos modificar dicho comportamiento, por ejemplo para aplicar determinados estilos CSS (colores, alineación interna, etc.) que cambien según el tamaño de pantalla.

En estos casos será necesario que creemos nuestra propia media query para aplicar los estilos deseados.



## Media queries

Una media query se define de la forma:

```
@media (min-width: TAMAÑO-EN-PÍXELES) {  
/* Los estilos aquí contenidos solo se aplicarán a partir del tamaño de pantalla  
indicado */  
}
```

En este caso, los estilos que estén dentro de esta media query se aplicarán solo a partir del tamaño en píxeles indicado. Además del tamaño mínimo, **también podemos indicar el tamaño máximo o el rango de tamaño** en el que se aplicarán los estilos, de la forma:

```
@media (max-width: TAMAÑO-EN-PÍXELES) {  
/* Estos estilos solo se aplicarán hasta el tamaño indicado */  
}  
@media (min-width: TAMAÑO-EN-PÍXELES) and (max-width: TAMAÑO-EN-PÍXELES) {  
/* Solo se aplicarán entre los tamaños indicados */  
}
```



# Media queries

Los media queries de Bootstrap son:

```
// Small devices (landscape phones, 576px and up) @media  
(min-width: 576px) { ... }
```

```
// Medium devices (tablets, 768px and up)  
@media (min-width: 768px) { ... }
```

```
// Large devices (desktops, 992px and up)  
@media (min-width: 992px) { ... } //
```

```
// X-Large devices (large desktops, 1200px and up) @media  
(min-width: 1200px) { ... }
```

```
// XX-Large devices (larger desktops, 1400px and up)  
@media (min-width: 1400px) { ... }
```

```
// Extra small devices (sm) (portrait phones, less than 576px)  
@media (max-width: 575.98px) { ... }
```

```
// Small devices (md) (landscape phones, less than 768px) @media  
(max-width: 767.98px) { ... }
```

```
// Medium devices (lg) (tablets, less than 992px) @media  
(max-width: 991.98px) { ... }
```

```
// Large devices (xl) (desktops, less than 1200px) @media  
(max-width: 1199.98px) { ... }
```

```
// XX-Large devices (xxl) (desktops, less than 1400px)  
@media (max-width: 1399.98px) { ... }
```



## Media queries

Si por ejemplo queremos que en las pantallas extra pequeñas (md) el color de fondo que aplica la clase **.miestilo** sea rojo y para el resto de tamaños sea verde, podríamos hacer:

```
.miestilo {  
  background-color: green;  
}  
  
@media (max-width: 767.98px) {  
  .miestilo {  
    background-color: red;  
  }  
}
```



# Introducción a Sass





## Introducción a SASS

La principal ventaja de SASS es la posibilidad de convertir los archivos CSS en algo dinámico añadiéndole características que están presentes en los lenguajes de programación tradicionales.

Permite el uso de variables, reglas CSS anidadas, mixins, importación de hojas de estilos y muchas otras características, al tiempo que mantiene la compatibilidad con CSS.



# Introducción a SASS

Existen dos tipos de sintaxis para escribir su código:

**Sintaxis Sass:** esta sintaxis es un poco diferente de la sintaxis de CSS estándar. Por ejemplo, evita colocar puntos y comas al final de los valores de propiedades. Además, las llaves no se usan y en su lugar se realizan indentados.

**Sintaxis SCSS:** Es una sintaxis bastante similar a la sintaxis del propio CSS. Podríamos decir que SCSS es código CSS con algunas cosas extras.



## Introducción a SASS

En la práctica, aunque podría ser más rápido escribir con sintaxis Sass, es menos recomendable, porque te aleja más del propio lenguaje CSS.

Los archivos **SASS** se identifican con la **extensión .sass** mientras que los **SCSS con la extensión .scss**

Nuestro archivo .scss y .sass no lo puede interpretar directamente el navegador, por lo que se tiene que convertir a la extensión CSS.



## Introducción a SASS

Para ello usamos **compiladores como KOALA** (<http://koala-app.com/>)  
Además de transformar el archivo .scss en .css te permite comprimirlo/  
minificarlo para que sea más fácil de leer por el navegadores y mejore la  
velocidad de carga del sitio.

Al comprimirlo lo que se hace es eliminar los saltos de línea y los  
espacios en blanco, innecesarios entre caracteres. Todos estos  
elementos consumen espacio y tiempo y recursos de los navegadores  
web para procesarlos.



# Variables

Permite reutilizar valores que podemos manejar desde un solo sitio de forma sencilla y centralizada. Las variables comienzan con el signo de pesos (\$).

SCSS CSS

```
$brand: #F98355;  
body {  
  color:$brand;  
}
```

```
body {  
  color: #F98355;  
}
```



## Nesting (Anidación)

SASS nos ayuda a simplificar haciendo uso del nesting o anidación. Conseguimos escribir menos, con la indentación del código CSS podemos reflejar mejor la jerarquía.

SCSS CSS

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  li { display: inline-block; }  
}
```

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}
```



## Nesting (Anidación)

Además, podemos referenciar fácilmente los padres de la etiqueta haciendo uso del selector “&”.

SCSS CSS

```
nav {  
  li {  
    display: inline-block;  
    &:hover {  
      background-color: red;  
    }  
  }  
}  
  
#main {  
  color: black;  
  &-sidebar { border: 1px solid; }  
}
```

```
nav li {  
  display: inline-block;  
}  
  
nav li:hover {  
  background-color: red;  
}  
  
#main {  
  color: black;  
}  
#main-sidebar {  
  border: 1px solid;  
}
```



## Referencias

- <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/tutorial-de-sass/>
- <https://uniwebsidad.com/libros/sass>
- <http://koala-app.com/>
- <https://startbootstrap.com/template/scrolling-nav>
- <https://getbootstrap.com/>