**@media + pseudo clases**

## Responsive

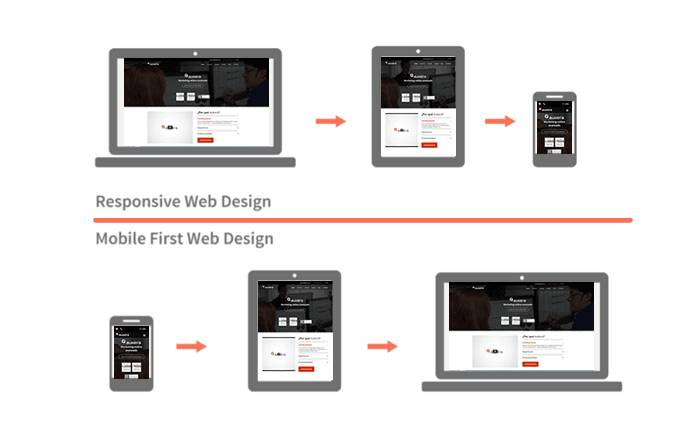
**Vamos a repasar responsive**

Volvemos a tocar el tema, pues para aprovechar Bootstrap, es fundamental conocer y entender muy bien el responsive. Cualquier cambio que se realice altera el diseño en todos los dispositivos.

**Página responsive**

El sistema detecta automáticamente el ancho de la pantalla y, a partir del mismo, adapta todos los elementos de la página, desde el tamaño de letra hasta las imágenes y los menús, ofreciendo al usuario la mejor experiencia posible.

**Recordemos**

****

Media Queries

Recordemos los rangos que define Bootstrap:

* **Pantallas extra pequeñas (móviles) < 576px.**
* **Pantallas pequeñas (\_sm, tablets \_en vertical) ≥ 576px.**
* **Pantallas medianas (md, para tablets en horizontal) ≥ 768px.**
* **Pantallas grandes (lg, tamaño escritorio) ≥ 992px.**
* **Pantallas extra grandes (xl, escritorio grande) ≥ 1200px.**

[Breakpoints en Bootstrap](https://getbootstrap.com/docs/4.1/layout/overview/#responsive-breakpoints)

Ejemplo de Media Queries

Si quisiera que en las pantallas extra pequeñas (xs) el color de fondo que aplica la clase **.miestilo** sea rojo, y para el resto de tamaños sea verde, podrías hacer:

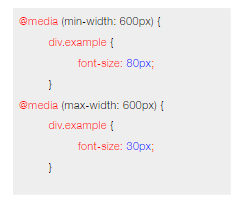
****

Si quisiera variar la alineación del texto que se aplica en una clase, a partir de las pantallas tipo escritorio:



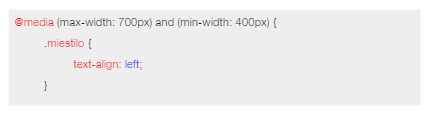
Puedes también modificar el cuerpo de texto, si lo utilizas en px para diferentes pantallas:

**Recuerda que esto no es necesario si utilizas el font-size en “em”.z**



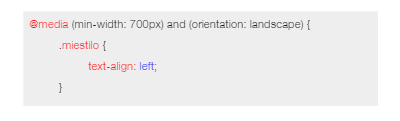
Operador “ADD”

Puedes sumar diferentes indicaciones con las Media Queries, las cuales se utilizan con el operador “add”. En este caso, el estilo que definido se reproducirá en pantallas que van de 400px a 700px:



Orientación

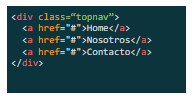
En este caso, sólo se reproducirá el estilo si la ventana tiene un ancho de de 700px o más, y la pantalla está en formato horizontal.



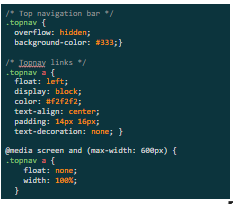
Ejemplo de Media Queries

Modificación de menú de posición horizontal en pantallas grandes, a vertical en pantallas chicas.

**HTML**

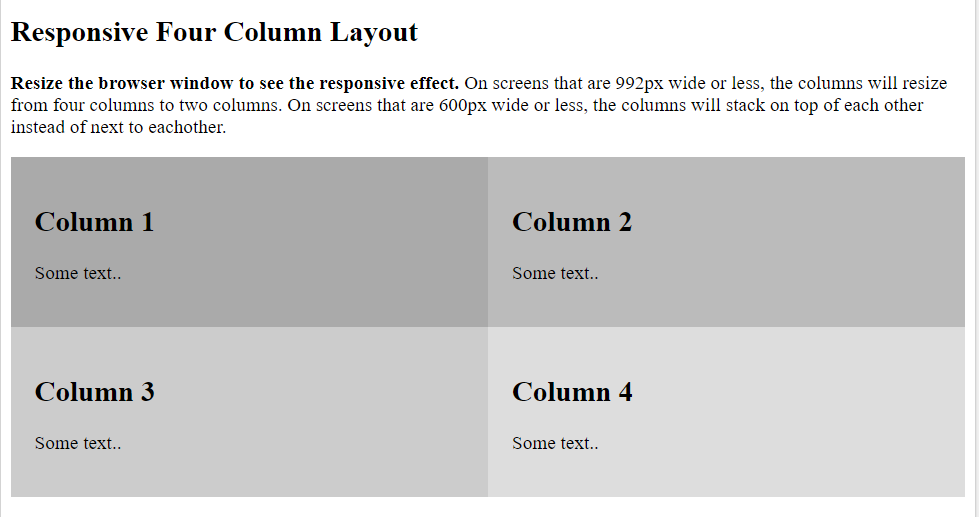


**CSS**

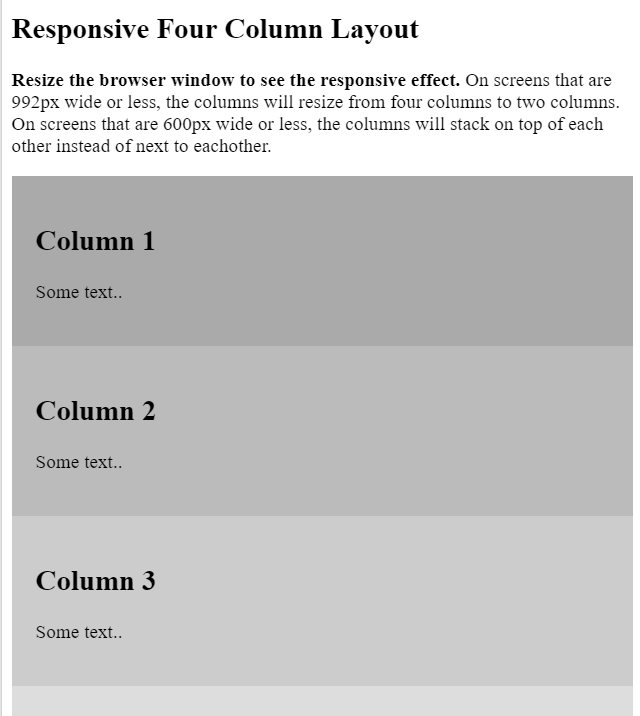


Puedes modificar la distribución de las columnas para que en pantallas pequeñas se vean una abajo de la otra:

**Desktop**

****

**Mobile**



[Ver el código](https://www.w3schools.com/css/tryit.asp?filename=trycss_mediaqueries_ex2)

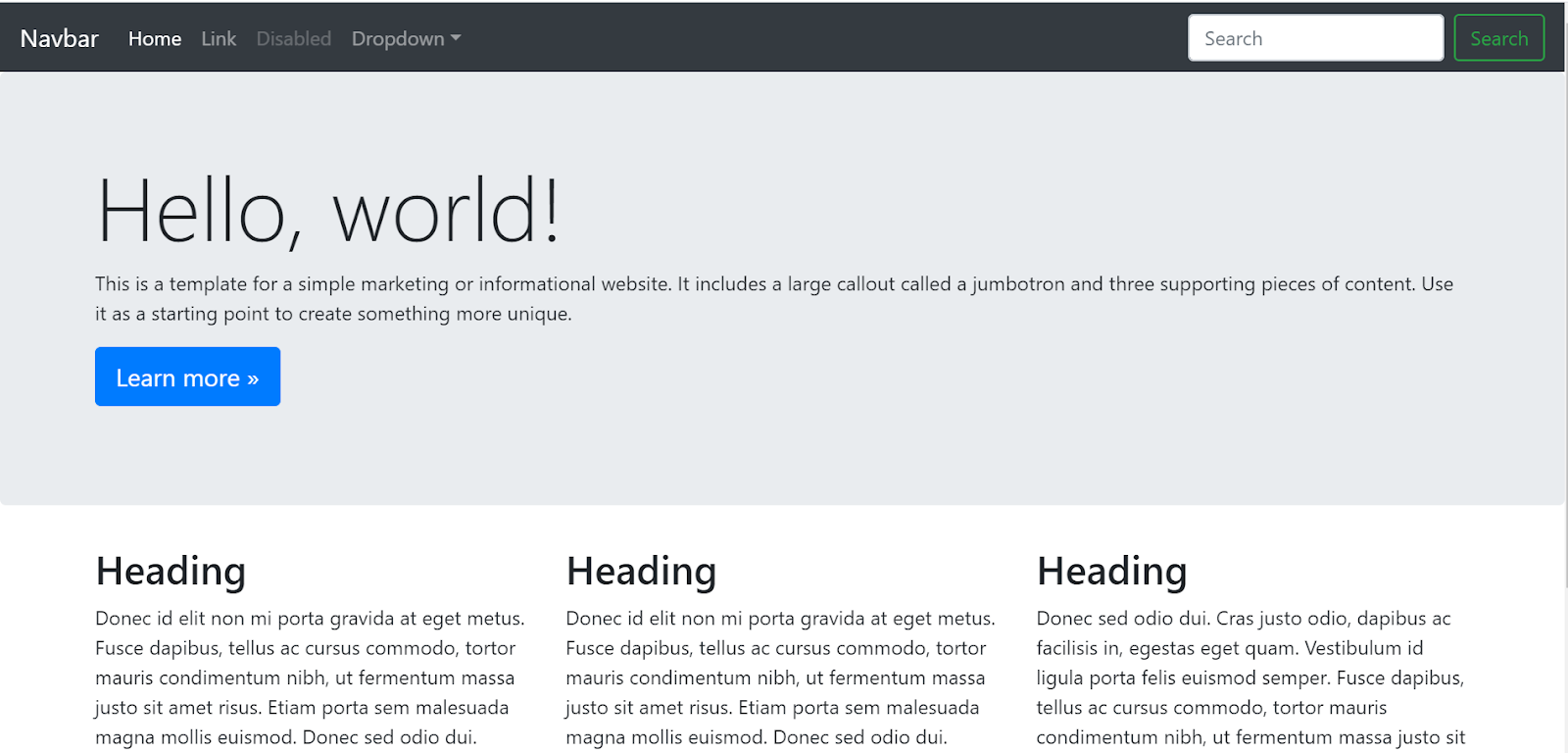
**Puedes llegar a hacer cosas muy avanzadas y personalizar completamente el aspecto de una web según el tamaño del dispositivo.**

* Cambiar el tamaño y la posición de una imagen.
* Modificar la posición de cualquier elemento.
* Cambiar el tamaño de letra, la fuente o su color.
* Aplicar combinaciones de estilos avanzados.

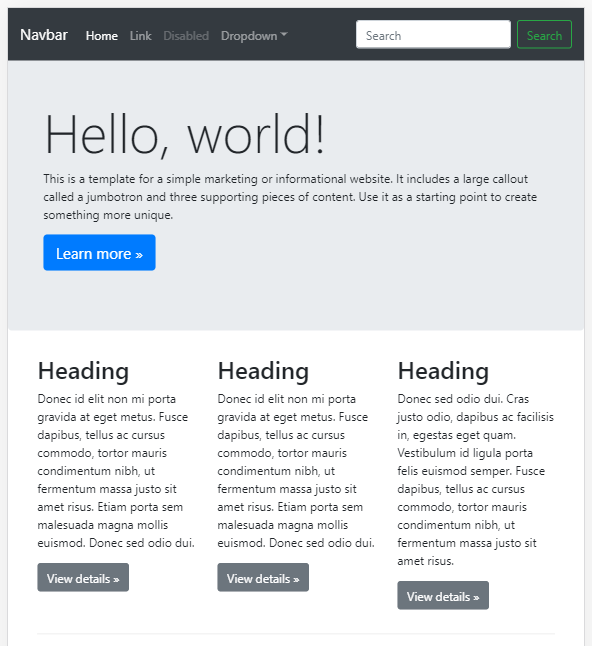
**¡Cualquier cosa que se te ocurra! 🚀**

Ejemplo

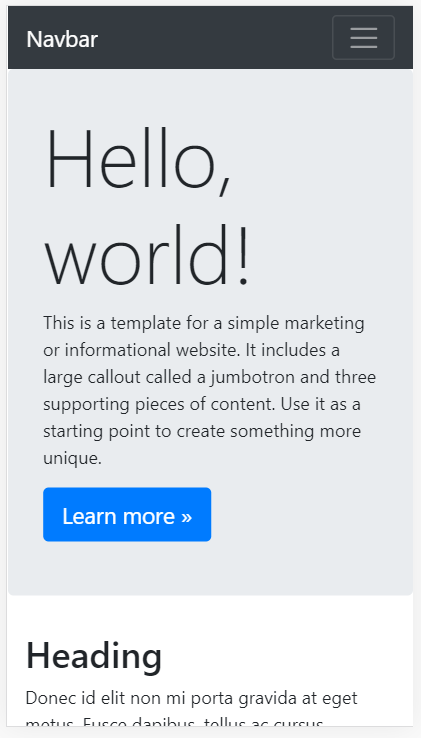
**Desktop**



**Tablet**

****

**Mobile**



Pseudo clases

¿Qué es una pseudo clase? 🤔

Una pseudoclase CSS es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.

Por ejemplo, :hover aplicará un estilo cuando el usuario haga hover sobre el elemento especificado por el selector.

**selector:pseudoclase { propiedad: valor; }**

Las pseudoclases, junto con los pseudoelementos, permiten aplicar un **estilo** a un elemento no solo en relación con el contenido del árbol de documento, sino **también en relación a factores externos** como:

* El historial del navegador (:visited, por ejemplo),
* El estado de su contenido (como :checked en algunos elementos de formulario),
* La posición del ratón (como :hover que permite saber si el ratón está encima de un elemento o no).

Teniendo como referencia…

<div id="abuelo">

<p>Párrafo 1</p>

<p>Párrafo 2</p>

<div class="padre">

<div class="hijo"><a title="Título hijo1">HIJO1</a></div>

<div class="hijo"><a>HIJO2</a></div>

<div class="hijo"><a>HIJO3</a></div>

</div>

<p>Párrafo 3</p>

<p>Párrafo 4</p>

<div class="padre">

<div class="hijo"><a title="Título hijo4">HIJO4</a></div>

<div class="hijo"><a>HIJO5</a></div>

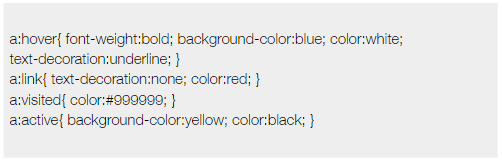
<div class="hijo"><a>HIJO6</a></div>

</div>

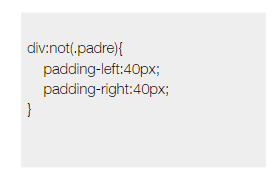
<p>Párrafo 5</p>

<p>Párrafo 6</p>

</div>

**Algunos ejemplos  
Ejemplos más complejos**

Otra pseudo clase muy útil pero poco utilizada, aplica a todos los elementos A, excepto a los incluidos como elementos B. Es decir, podríamos aplicar unos estilos a todos los elementos “div” y evitar que éstos se apliquen a otras capas con un id o class determinado. En el siguiente ejemplo hacemos justo esto, evitando que se apliquen los estilos a las capas “padre“.

****

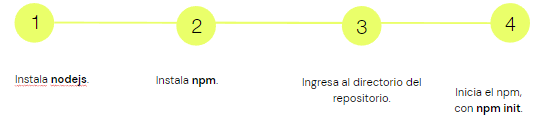
**Otros ejemplos…**

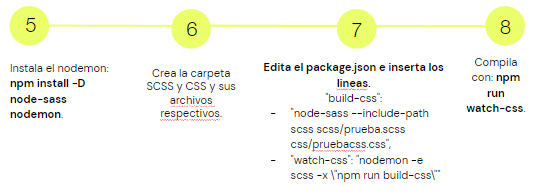
La solución a todos los problemas de maquetación de antaño con las filas y los bloques.

👉 Con :nth-child(N) podremos aplicar sus estilos a todos los elementos hijos cuya posición sea un número “N” respecto a un padre. Este número N no tiene que ser necesariamente un número entero para especificar una posición fija (la posición 2, por ejemplo), ya que también permite insertar fórmulas y palabras específicas.

****

Configurando Node

****

****

**Material ampliado**

[Themes y templates](https://teloexplicocongatitos.com/poster?id=tlecg04) | **TE LO EXPLICO CON GATITOS**

[Íconos Material Design](https://materialdesignicons.com/) | **Material Design Icons**