



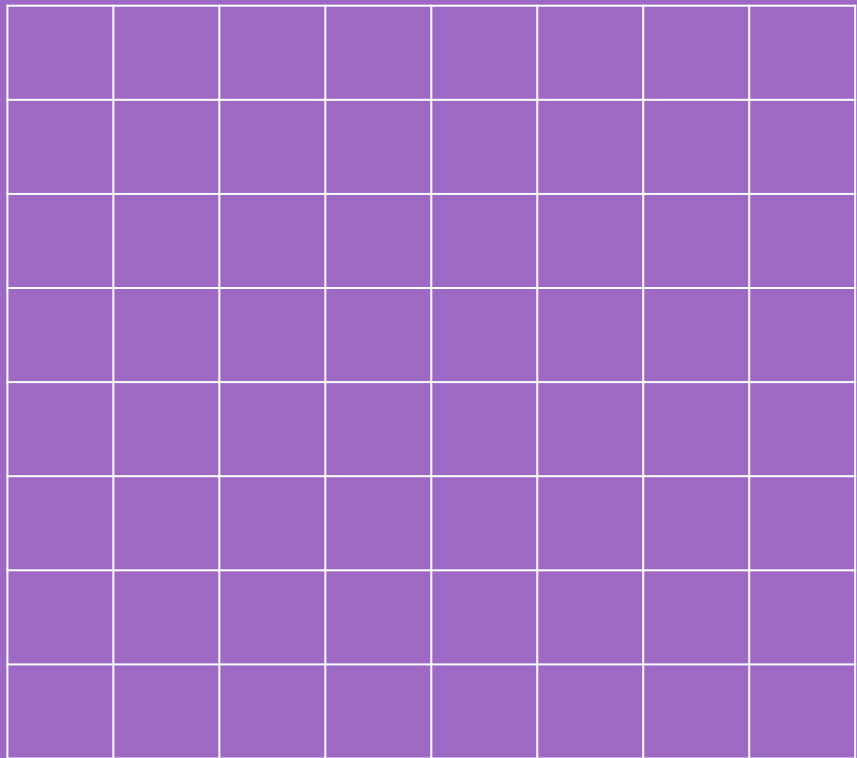
Argentina
programa
4.0

Media Queries

“Introducción a la Programación Web”



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) { ... }
```

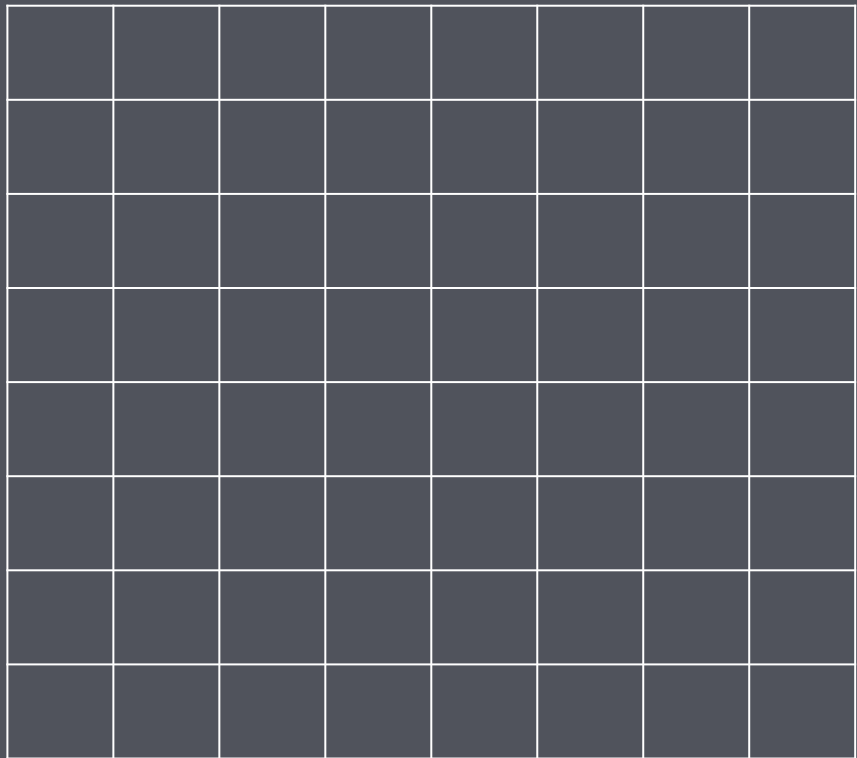
Media queries

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;  
  }  
}
```



Componentes Bootstrap



Botones

Mediante la clase **.btn** podemos aplicar estilo a los elementos tipo button. Esta clase la podemos combinar con otras clases que ya trae Bootstrap: **.btn-primary**, **.btn-secondary**, **.btn-success**, **.btn-danger**, **.btn-warning**, **.btn-info**, **.btn-light**, **.btn-dark** o **.btn-link** para crear botones para diferentes estados o acciones en nuestros formularios:

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>
```



Modales

Un modal es una capa que se superpone a las demás quedando en primer plano de nuestra web y dejando en un segundo plano el resto de la página.

Si quisieramos que el modal se abriera, por ejemplo, al hacer clic sobre un enlace, sólo tendríamos que insertarle los atributos data-toggle y data-target en la propia etiqueta

```
<!-- Button trigger modal -->  
<button type="button" class="btn btn-primary" data-toggle="modal"  
data-target="#exampleModal"> Launch demo modal </button>
```

El atributo “data-target” es el encargado de contener el identificador del DIV del modal.

Modales

Luego tenemos el contenido del propio modal:

```
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel"
aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close"> <span
aria-hidden="true">&times;</span> </button>
      </div>
      <div class="modal-body"> ... </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
</div>
```

El ID del modal debe coincidir con el data-target del botón que llamará a este modal.

Ocultando elementos

Para ocultar elementos, simplemente se debe usar la clase **.d-none** o una de las clases **.d-{sm,md,lg,xl,xxl}-none** para cualquier variación de pantalla.

Para mostrar un elemento se utiliza la clase **.d-block** o una de las clases **.d-{sm,md,lg,xl,xxl}-block** para cualquier variación de pantalla.

Ocultando elementos

Screen size	Class
Hidden on all	<code>.d-none</code>
Hidden only on xs	<code>.d-none .d-sm-block</code>
Hidden only on sm	<code>.d-sm-none .d-md-block</code>
Hidden only on md	<code>.d-md-none .d-lg-block</code>
Hidden only on lg	<code>.d-lg-none .d-xl-block</code>
Hidden only on xl	<code>.d-xl-none</code>
Hidden only on xxl	<code>.d-xxl-none .d-xxl-block</code>
Visible on all	<code>.d-block</code>
Visible only on xs	<code>.d-block .d-sm-none</code>
Visible only on sm	<code>.d-none .d-sm-block .d-md-none</code>
Visible only on md	<code>.d-none .d-md-block .d-lg-none</code>
Visible only on lg	<code>.d-none .d-lg-block .d-xl-none</code>
Visible only on xl	<code>.d-none .d-xl-block .d-xxl-none</code>
Visible only on xxl	<code>.d-none .d-xxl-block</code>

Librería de íconos

Bootstrap tiene su propia fuente de íconos, para insertarla debemos poner su enlace CDN en el head de nuestro html.

```
<link rel="stylesheet"  
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">
```

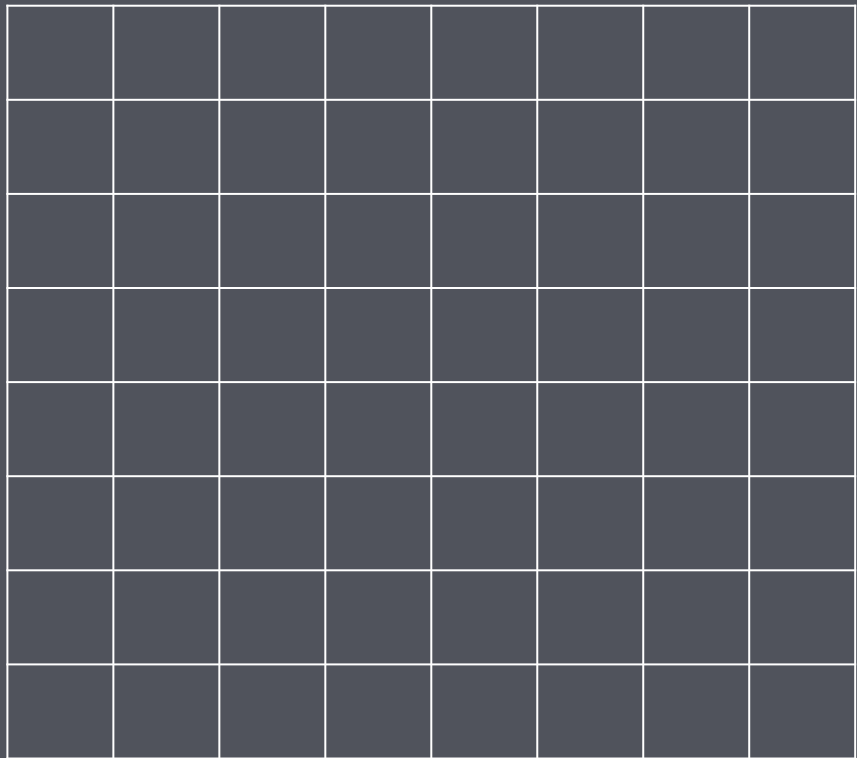
Luego debemos poner el ícono donde queramos de la siguiente manera:

```
<i class="bi-alarm"></i>
```

Documentación y lista de íconos: <https://icons.getbootstrap.com/>



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

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

CSS

```
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

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

CSS

```
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

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

CSS

```
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/>



**Argentina
programa
4.0**

Gracias!
