

«Talento Tech»

Desarrollo Web 1

Clase 08





Clase N° 8 | Conceptos básicos

Temario:

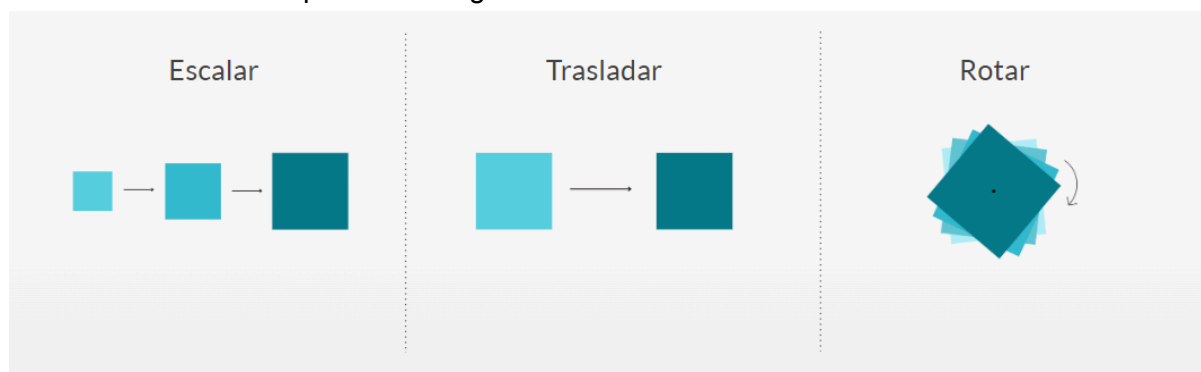
- Transitions
- Transform
- Librerías de animaciones y cómo utilizarlas
- Qué es Bootstrap
- Integración de componentes

Transform.

Como su nombre lo indica en inglés, transform es transformar. Se refiere a un conjunto de propiedades que le podemos aplicar a un elemento para generar un cambio visual (escalar, trasladarlo o rotarlo). Por ejemplo podríamos hacer uso de la pseudoclase **:hover** en css como activador de transformaciones, en este caso, el efecto sucederá cuando le pasamos el mouse por encima al elemento.

Otro ejemplo: Cuando interactuamos con un botón o un link en alguna página en internet. notarás que lo que se transforma es una característica del elemento. Puede ser desde su color de fondo, el color de la tipografía, o algo con más movimiento como su rotación, traslación (que se mueva de posición) o tamaño.

Las transformaciones que vamos a generar son:



ESCALAR

Dentro de transform se usa el valor scale (escalar o agrandar), y entre paréntesis un número.

Veamos un ejemplo de código CSS aplicado a un div vacío:

```
CSS

.agrandar {
  width: 50px;
  height: 50px;
  background-color: orange;
  margin: 50px;
}

.agrandar:hover{
  transform: scale(2);
}
```

Dentro de los parámetros se ponen números enteros.

1 = 100% del tamaño del elemento

2 = 200% del tamaño del elemento, es decir que medirá el doble.

3 = 300% y así sucesivamente.

TRASLADAR

Dentro de transform se usa el valor translate (trasladar), y entre paréntesis se colocan dos números (en px o en %) para indicar cuánto se quiere correr ese elemento en el eje X y en el eje Y. Veamos un ejemplo de código CSS aplicado a un div vacío:



```

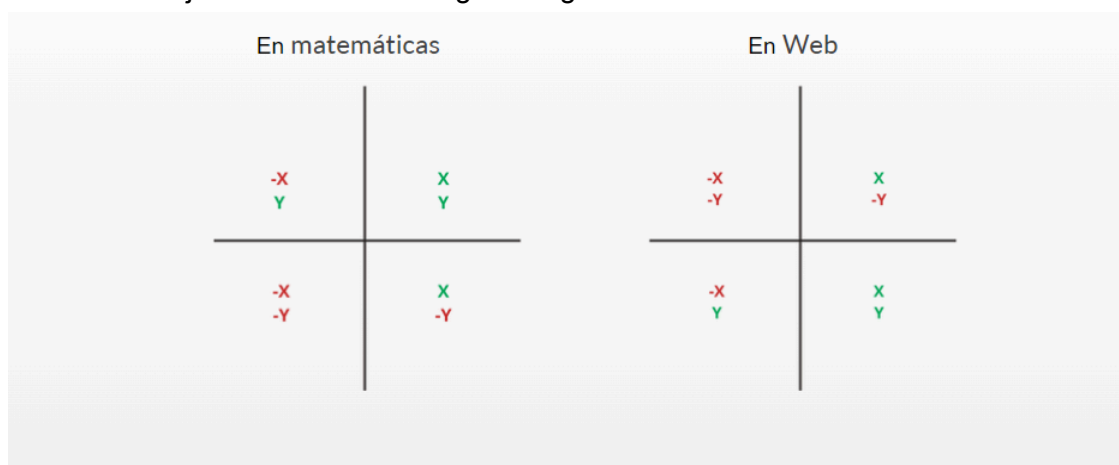
    CSS

    .caja {
      width: 50px;
      height: 50px;
      background-color: orange;
      margin: 50px;
    }

    .caja:hover{
      transform: translate(20px,
      30px);
    }
    
```

RECORDANDO LOS EJES

En matemáticas es común que los valores negativos sobre el eje Y sean hacia abajo. En web sucede al revés, como vimos en el ejemplo anterior. Los valores positivos son los que van hacia abajo. Veámoslo en el siguiente gráfico:



ROTAR

Dentro de transform de usa el valor rotate (o rotar), y entre paréntesis un número seguido de “deg”, es decir degrees, que en español significa grados.

```

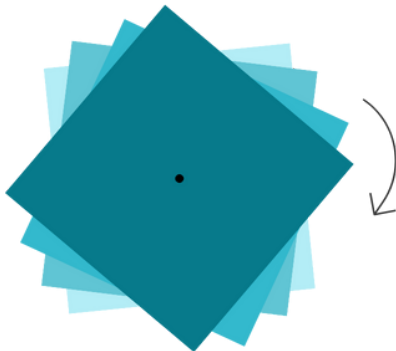
    ● ● ● CSS

    .caja {
      width: 50px;
      height: 50px;
      background-color: orange;
      margin: 50px;
    }

    .caja:hover{
      transform: rotate(45deg);
    }
    
```

ROTAR

Por último, en este caso, podemos cambiar el origen de la rotación:
 Por defecto el giro se hace sobre el eje central del elemento.





Para modificarlo se puede agregar la propiedad transform-origin: y agregarle dos valores: top o bottom (para arriba o abajo) o left o right (para izquierda o derecha)



Transition

Como su nombre lo indica en inglés, transition es transición. Se refiere a una propiedad que podemos aplicar a un elemento para suavizar o transicionar los cambios efectuados con el hover cuando le pasamos el mouse por encima. En el apartado anterior generamos cambios en los elementos con transform, pero con la transition lo que logramos es manipular a través de propiedades la velocidad y el modo en el que un elemento pasa de un estado a otro.

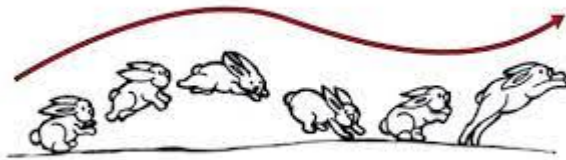
Para ver cómo lograr este efecto te invitamos a ver la siguiente presentación.

[*Insertar presentacion Clase 08 Transitions](#)

Animaciones

Una animación en el contexto del desarrollo web se refiere a la técnica de crear una serie de cambios visuales a lo largo del tiempo para generar una ilusión de movimiento. En lugar de presentar una interfaz estática, una animación permite que los elementos de la página

web se muevan, cambien de forma, color u otras propiedades, creando una experiencia visual más dinámica y atractiva para el usuario.



Hasta el momento hemos aprendido a modificar algunos aspectos utilizando transform y la pseudoclase hover, y los cambios logrados se ven un poco bruscos.

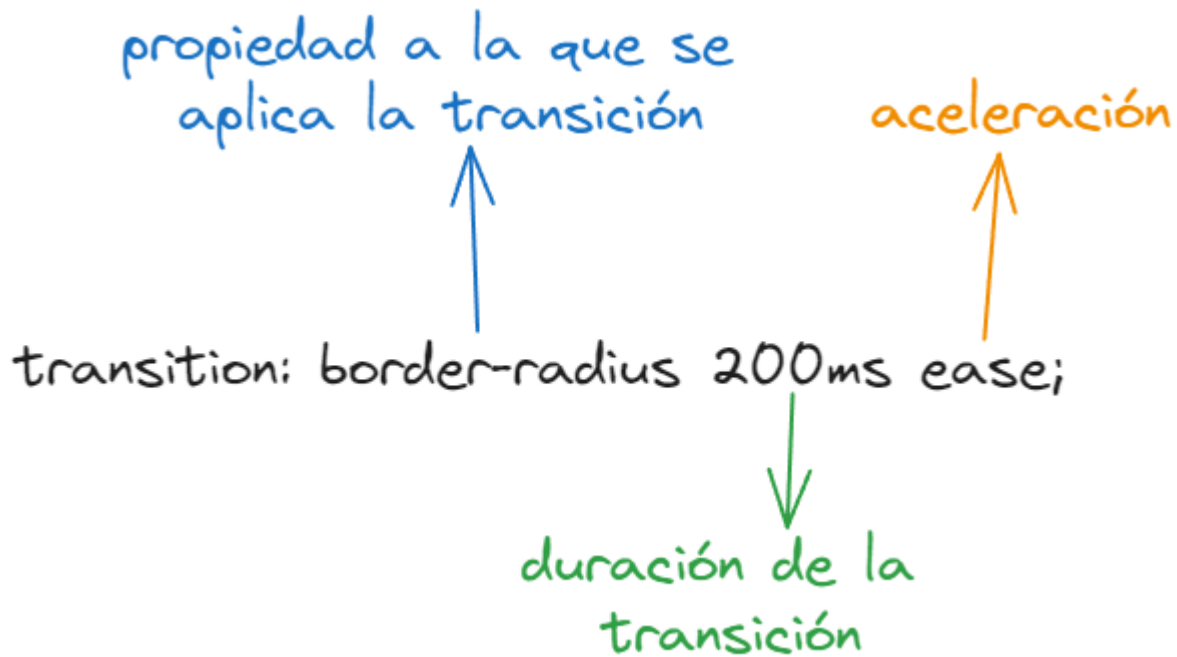


Para lograr un cambio progresivo se utiliza la propiedad transition. De esta manera logramos entonces una transición de una propiedad a otra.



Transition

La propiedad transition acepta varios valores. Por ejemplo:



Si más de una propiedad va a sufrir cambios en nuestro elemento podemos utilizar la palabra **all** para que la transición se aplique a todas ellas.

la transición se
aplica a todas las
propiedades

aceleración

transition: all 0.2s ease-in-out;

duración de la
transición en
segundos

Uso de transition

La propiedad transition es utilizada en el grupo de propiedades "original" del elemento. Y aparte, como venimos trabajando, las propiedades que se harán visible con hover. Veamos la siguiente estructura repasando el tema anterior de transform.

```

    CSS

    .caja {
        width: 50px;
        height: 50px;
        background-color: orange;
        margin: 50px;
        transition: transform 200ms ease;
    }

    .caja:hover{
        transform: rotate(20deg);
    }
    
```

En este ejemplo, la propiedad que va a transicionar es la de “transform”, que sucederá durante el hover. Es decir, rotate.

Librerías

Existen muchos tipos de transiciones, transformaciones y animaciones que podemos lograr hover, algunas son más complejas que otras y pueden llevar bastante tiempo de desarrollo, pero para ahorrar ese tiempo y facilitar su uso se crearon librerías con animaciones pre hechas que podemos copiar y pegar en nuestro propio código.

Las librerías son conjuntos de código preescrito y reutilizable que facilitan el desarrollo de aplicaciones y sitios web. También conocidas como bibliotecas, están diseñadas para abordar tareas comunes y repetitivas, lo que permite a los desarrolladores ahorrar tiempo y esfuerzo al no tener que escribir todo el código desde cero.

Las bibliotecas web generalmente están escritas en lenguajes de programación como JavaScript, y ofrecen una variedad de funciones y utilidades que se pueden incorporar fácilmente en un proyecto.



Algunos ejemplos populares de bibliotecas para desarrollo web son jQuery, React, Vue.js, Bootstrap, entre otras. Cada biblioteca se enfoca en **diferentes áreas** y ofrece ventajas específicas según las necesidades del proyecto. Las dos librerías que te compartimos a continuación son **específicas para trabajar animaciones y transiciones**.



Atención:

Cada librería tiene su manual de uso, por eso es importante que al momento de elegir una leas la documentación que provee. De esta manera podrás incorporarlas a tu sitio sin errores y sabrás cómo aprovechar sus propiedades en cada elemento y parte de tu página.



animate.style

Esta librería es de uso libre y gratuito, y es super fácil de utilizar. El sitio oficial es

<https://animate.style/>

Te recomendamos ver el sitio en inglés para que puedas ver los efectos visuales que ofrece.

En la parte derecha del sitio verás un listado con todos los efectos. Al hacer clic en cada uno podrás previsualizarlos.



Paso a paso

Paso 1) Más abajo en el sitio web <https://animate.style> vas encontrar la documentación que explica distintas formas de linkear esta librería con tu proyecto. El modo de linkearlo será por CDN (Content delivery network). En la documentación encontraremos el **link que debemos copiar y pegar en el head de nuestro archivo HTML.**

Paso 2) En la documentación también está explicado el modo de aplicar la animación a cada elemento.

Este paso nos indica que debemos agregar la clase **animate__animated** al elemento que queremos animar. Ejemplo:

Paso 3) Como el manual de uso indica, luego de la clase `animate__animated` debemos agregarle el nombre de la animación que queremos. Para eso, vamos al menos de la derecha del sitio web, elegimos la animación y copiamos su nombre. Al lado de cada nombre tendrás un icono de “copiar” para hacerle clic.

Paso 4) Pegamos el nombre de la animación a la clase, al lado de **animate__animated**

¡Listo!

Este tipo de efectos (u otros) se puede agregar también a contenedores completos. Esto afectará sólo al contenedor y el efecto visual se verá en conjunto.

Configuración

En el manual de uso de esta librería podrás encontrar configuraciones que pueden aplicarse a las animaciones. Por ejemplo, hacerlas más rápidas o más lentas controlando su duración, hacer que comience con retraso aplicando un delay o también lograr que la misma animación se repita varias veces. Te invitamos a ver el manual de uso en la sección **Propiedades personalizadas de CSS (variables CSS)**



AOS

AOS

Esta librería es de uso libre y gratuito, y es super fácil de utilizar. El sitio oficial es <https://michalsnik.github.io/aos/>

Esta librería es útil para aplicar animaciones al momento de hacer scroll en la página.

Paso 1) Al igual que como hicimos con la librería anterior, debemos linkear esta librería al proyecto. Al final del sitio encontraremos tres links importantes. El primero de ellos es el CDN que debemos copiar y pegar en el head del proyecto.

Paso 2) Luego linkeamos los dos links de javascript que nos indica la librería para que pueda funcionar e iniciarse. Estos links van abajo de todo el documento HTML, justo antes de que cierre el <body>

Paso 3) Por último, queda recorrer el sitio web haciendo scroll para visualizar el efecto en cada cuadro y tomar de ellos el que nos interesa **copiando y pegando la clase completa en algún elemento del proyecto.**

¡Listo!

Resultado:

Este tipo de efectos (u otros) se puede agregar también a contenedores completos. Esto afectará sólo al contenedor y el efecto visual se verá en conjunto. Por ejemplo, en esta grilla o en esta lista de productos o cards.

Configuración.

Un poco más abajo de las animaciones tenemos una sección de configuraciones en donde directamente nos muestra algunos ajustes que podemos aplicar y cómo se ven.

EJEMPLOS DE DIFERENTES CONFIGURACIONES

Un poco más abajo de las animaciones tenemos un sección de configuraciones en donde directamente nos muestra algunos ajustes que podemos aplicar y cómo se ven.

data-aos-duration="3000" (Duración De La Animación)

data-aos-easing="linear" (Tipo De Movimiento)

data-aos-offset="300"

data-aos-anchor-placement="top-bottom" (Dónde Termina La Animación)

Te dejamos a continuación información sobre el modo de uso otras configuraciones haciendo [click acá](#)

Podes ponerlo en español para entenderlo mejor.



Más Librerías

Existen muchas otras librerías para diferentes acciones que se pueden hacer en el sitio web. Cada una de ellas, como mencionamos, tienen su manual de uso y es importante conocerlo para luego poder configurar todo a los gustos y necesidades del sitio web.

Te invitamos a investigar en internet sobre otras librerías de acceso libre y gratuito que se puedan utilizar y a compartirlas con el resto de la clase.



Atención:

Te dejamos por último algunas recomendaciones al momento de utilizar animaciones:

- Utilizar la misma librería, evitar combinar distintas bibliotecas para no tener muchos tipos de operaciones y clases diferentes.
- De la misma librería, utilizar siempre el mismo tipo de animaciones para no generar mucho despliegue de movimientos que incomodan al usuario.
- Las animaciones son herramientas para llamar la atención del usuario, por lo tanto, se recomienda usarlas con precaución sólo en los elementos importantes y que no agobien a la vista.

Ahora sí, ¡a ponerle movimiento a la página web!.

Bootstrap



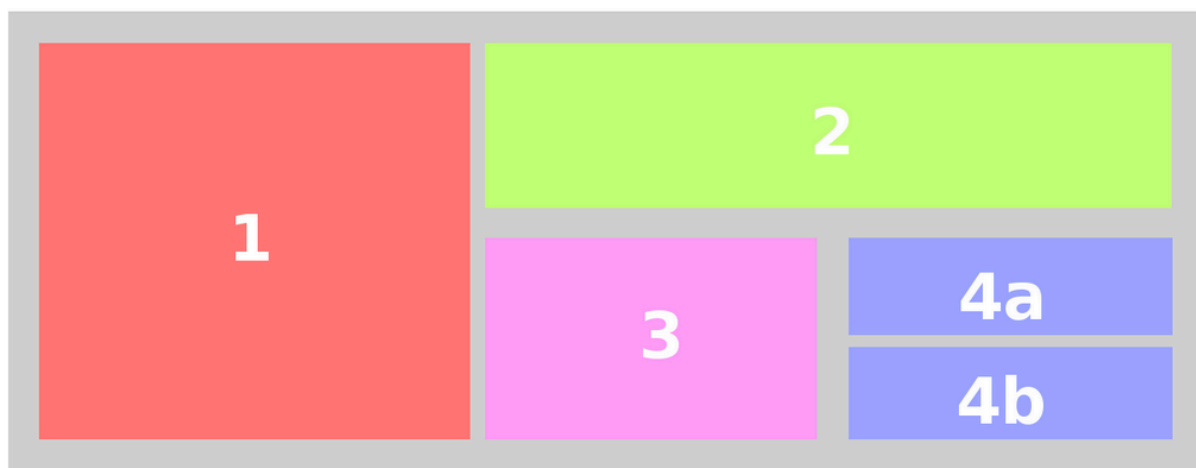
Bootstrap es una librería muy popular de desarrollo web de código abierto que facilita la creación de sitios web y aplicaciones con un diseño responsivo y moderno. Proporciona un conjunto de estilos CSS predefinidos, componentes reutilizables y scripts de JavaScript, que permiten a los desarrolladores crear interfaces de usuario atractivas y funcionales de manera rápida y sencilla.

Características de Bootstrap

Algunas características importantes de Bootstrap son:

Grilla flexible

Bootstrap utiliza un sistema de 12 columnas que permite organizar y distribuir el contenido de manera flexible en diferentes dispositivos y tamaños de pantalla.



Componentes predefinidos

Bootstrap ofrece una amplia variedad de **componentes listos para usar**, como botones, barras de navegación, tarjetas, formularios, entre otros. Estos componentes **se pueden personalizar y combinar** fácilmente para adaptarse a las necesidades del proyecto.

Estilos y temas: Bootstrap proporciona una amplia gama de estilos CSS predefinidos que permiten mejorar la apariencia de los elementos y lograr una coherencia visual en todo el sitio web. También ofrece la posibilidad de personalizar los estilos y crear temas personalizados.

Compatibilidad con dispositivos móviles: Bootstrap está diseñado para ser responsivo, lo que significa que los sitios web y aplicaciones desarrollados con Bootstrap se adaptan automáticamente a diferentes tamaños de pantalla y dispositivos, como teléfonos móviles y tablets.

Integración de JavaScript: Bootstrap incluye una serie de scripts de JavaScript que agregan funcionalidades interactivas a los componentes, como carruseles, modales, pestañas, entre otros. Estos scripts facilitan la implementación de elementos interactivos sin necesidad de escribir código JavaScript personalizado.



En la siguiente presentación verás el paso a paso de cómo linkear la librería desde su sitio oficial.

Link del sitio oficial de Bootstrap haciendo [clic acá](#)

¡Manos a la obra!

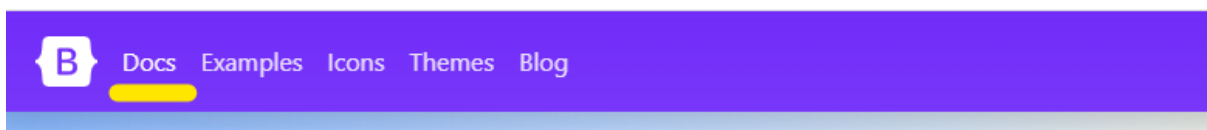
Integrando Bootstrap

Bootstrap contiene un gran conjunto de bloques de código HTML, CSS y JS pre armados que nos permite como desarrolladores ahorrar mucho tiempo de trabajo, pero para utilizarlo hay que conocer su estructura. Podríamos pensar que se trata de “materia prima” que podemos adaptar a nuestras necesidades. Es código ya empezado que podemos “amoldar” a un proyecto personal.

Vamos a integrar bootstrap al proyecto siguiendo estos pasos.

Tené en cuenta que estos pasos se están llevando a cabo con esta versión en el 2024, que a futuro puede cambiar.

Paso 1) Ingresamos al sitio oficial de bootstrap <https://blog.getbootstrap.com/> y nos dirigimos a la parte superior izquierda para ingresar a la parte de “Docs” o documentación.



Al igual que como vimos con las librerías, los marcos de trabajo también vienen con su respectiva documentación para ser estudiada con el fin de utilizar correctamente sus herramientas y evitar errores en el código.

Paso 2) Para ayudarte podés poner el sitio web en español. En la parte de “inicio rápido” vemos que nos pide colocar `<meta name="viewport" content="width=device-width, initial-scale=1">` en el head. Esta línea de código ya la tenemos gracias al atajo Emmet en VS Code. 😎

1. Cree un nuevo `index.html` archivo en la raíz de su proyecto. Incluya la `<meta name="viewport">` etiqueta también para [un comportamiento receptivo adecuado](#) en los dispositivos móviles.

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>

```

Paso 3) En este paso incluimos los link de CSS y JS de este framework.

2. **Incluye CSS y JS de Bootstrap.** Coloque la `<link>` etiqueta en el `<head>` para nuestro CSS y la `<script>` etiqueta para nuestro paquete de JavaScript (incluido Popper para colocar menús desplegables, ventanas emergentes e información sobre herramientas) antes del cierre `</body>`. Obtenga más información sobre nuestros [enlaces de CDN](#).

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
  </body>
</html>

```

Haciendo clic en este ícono que acompaña al código podrás copiar y pegar los ejemplos de



forma más rápida.

¡Atención!

El link CDN de CSS que vá dentro del <head>, va por encima del link CSS. ¿Por qué? Porque así podremos sobrecribir las propiedades de bootstrap con las nuestras personalizadas.

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/
bootstrap.min.css" rel="stylesheet"
integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRh002i
uK6FUUVM" crossorigin="anonymous">
<link rel="stylesheet" href="./css/index.css">
    
```

El link de <script> va justo antes del cierre de <body>

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/
dist/js/bootstrap.bundle.min.js"
integrity="sha384-geWF76RCwLtnZ8qwWowPONGuL3RmwHVBC9FhGdlKrxdi
JJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>
</body>
    
```

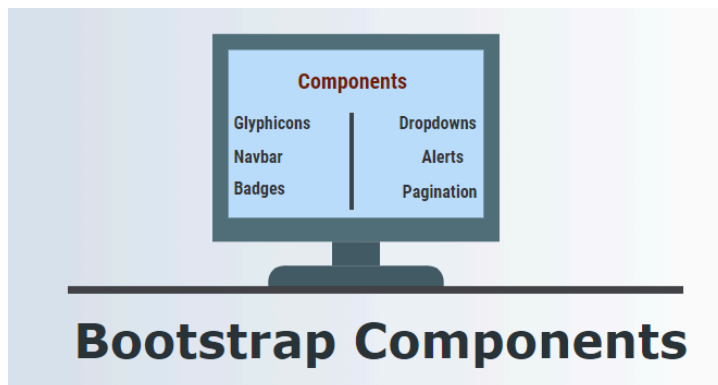
Siempre que utilices esta librería es recomendable traer los links correspondientes desde el sitio web oficial para asegurarte de que están actualizados.

¡Y listo!

Ya podemos utilizar los componentes en nuestros proyectos.



Integrando componentes



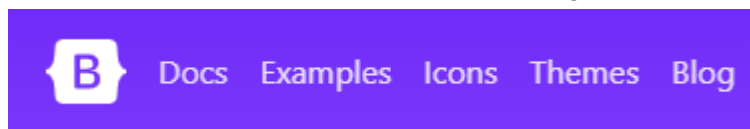
Una vez que tenemos linkado el archivo CSS y JavaScript proporcionado por Bootstrap, es momento de ir a buscar los componentes que nos serán útiles en la construcción de nuestro sitio. Verás que, como se mencionó anteriormente, todo está predeterminado. Los colores, la tipografía, los márgenes y espacios, las medidas. Pero a no decepcionarse porque estos componentes se pueden modificar a gusto y necesidad de cada proyecto, Es decir, los atributos que ya vienen hechos se pueden sobrescribir con un archivo CSS creado por nosotros mismos. Para ver más detalles del paso a paso te invitamos a ver la siguiente presentación.

Ya aprendimos que Bootstrap es una librería que contiene código de terceros listo y disponible para que utilicemos. En el caso de esta librería, el código listo y disponible se encuentra tanto en el .css como en el .js. Como es código que vamos a querer usar lo tenemos que integrar a nuestro proyecto. Para eso utilizamos la etiqueta `<link>` para el CSS y la etiqueta `<script>` para el JS.

Todas las instrucciones se encuentran en la página <https://getbootstrap.com/> igual que en la guía anterior.

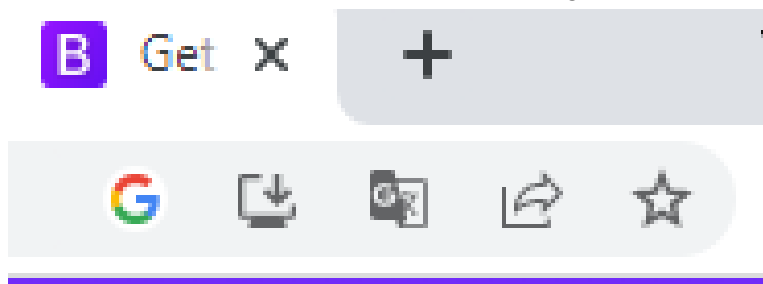


Una vez que tenemos integrada la librería a nuestro proyecto, no nos queda más que comenzar a buscar en la documentación cómo utilizarla. Si observamos la página de Bootstrap, vamos a ver en la barra de navegación un link que dice “Docs”



Si hacemos click ahí, nos llevará a la página de la documentación de la librería.

Como verás, la página está en inglés. Si manejas inglés no hay problema pero si no, siempre existe la opción de traducir el sitio gracias al traductor de Google:



A fines prácticos el lenguaje de la guía va a seguir con la traducida al español

En la documentación vamos a encontrar toda la información para utilizar la librería. Mientras más leamos, mejor vamos a poder utilizar y conocer la librería. Es super importante leer la documentación de todas las librerías que vayamos integrando a nuestro proyecto.

¿Hay que leer todo antes de utilizarla?

No. Si bien está bueno tener un amplio conocimiento de la librería, muchas veces se puede aprender a demanda. Cuando nos encontramos con algo que queremos hacer, leemos y buscamos a ver si encontramos una solución.

En esta guía te contamos lo que sí o sí tenés que saber de Bootstrap, pero hay mucho más por explorar.



Componentes



Una vez que hicimos click en “Docs”, estamos en la sección de la documentación. A la izquierda hay una barra de navegación que nos muestra todos los temas, algo así como un índice.



Cada sección vale la pena ser explorada y vamos a encontrar con que hay secciones más complicadas y otras más fáciles. Por ahora nos vamos a enfocar en la sección de “Componentes”.

Los componentes de Bootstrap son código HTML que ya vienen con las clases adecuadas para que se muestren de una forma particular.

El paso siguiente es copiar estos códigos y pegarlos en nuestro proyecto, y ya vamos a tener los componentes funcionando.

Existen muchos componentes y vamos a poder verlos cuando hacemos click sobre “Componentes”.

Barra de navegación/ Navbar

Cuando hacemos click en Componentes > Barra de navegación, se nos despliega toda la documentación pertinente a la barra de navegación. En los primeros párrafos nos explica cómo funciona, qué clases admite y otros detalles.

Si comenzamos a ir hacia abajo en la página vamos a ver que hay ejemplos de barra de navegaciones junto con el código a copiar.

La lógica con los componentes, es siempre la misma:

1. Hacemos click en el componente que nos interesa.
2. Leemos la información / documentación.
3. Buscamos el modelo del componente que nos sirva.
4. Copiamos el código.
5. Lo pegamos en nuestro proyecto..

No olvidemos tener linkado el CSS y el JS de Bootstrap para que puedan funcionar los componentes que queremos incorporar.

Una vez que encontremos el componente que nos guste y resuelva la necesidad que tenemos, simplemente lo pegamos en nuestro proyecto.

Si ahora abrimos nuestro archivo, vamos a ver algo como lo siguiente:
Es decir que tenemos la navbar lista para usar en nuestro proyecto.

¿Eso es todo?

Bueno... depende. Eso es todo a la hora de integrar un componente, pero ahora podemos cambiar al código y modificarlo para que se adecúe a las necesidades que tenemos.

Modificando la barra de navegación / navbar

Una vez que detectamos cada elemento en su línea de código traída desde Bootstrap, es leer sus clases. Veamos el siguiente ejemplo de un elemento `` del navbar anterior:

Su línea de código es así:

```
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
```

Como podemos ver en este ejemplo el elemento `` trajo clases ya asignadas desde la librería. Para saber más sobre el significado de estas clases podemos ver el instructivo de cada una de ellas que nos ofrece Bootstrap.

¡No te preocupes!. Puede ser un poco confuso al principio pero te contamos que no es necesario que utilices las clases que ofrece la librería si o si (aunque es mejor no eliminarlas hasta no estar seguros).

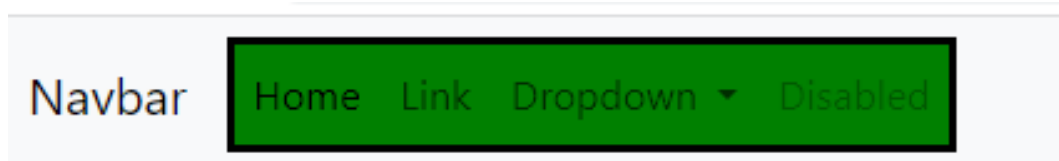
La buena noticia es que las podemos sobrescribir. Es decir, podemos inventar nuestras propias clases y aplicarlas. Veamos un ejemplo:

Para sobrescribir una característica determinada por Bootstrap lo que debo hacer es buscar en mi propio archivo HTML el elemento que copié desde la librería (continuamos con el ejemplo de elemento ``) y cuando lo encuentro, creo una nueva clase.

El siguiente paso, es agregar propiedades a esa clase desde CSS. Como hacemos normalmente, ingresamos al archivo CSS previamente linkado al archivo HTML, nombramos a la clase y le asignamos las propiedades y valores que deseamos cambiar del elemento.



Como resultado, el nuevo navbar modificado se verá de la siguiente manera:



Con este ejemplo pudimos ver cómo sobrescribir clases y asignarles otras que se ajusten a las necesidades de cada proyecto, y las funcionalidades quedan intactas.

Conceptualizando:

Si bien no vamos a ir componente por componente, lo importante acá es registrar el modo de operar que tuvimos:

1. Linkeamos CSS y JS de bootstrap en nuestro documento.
2. Buscar los componentes que nos sirven y los integramos al proyecto copiando y pegando el código que nos ofrece la librería.
3. Investigamos sus líneas de código y efectuamos alguna modificación agregando clases y propiedades nuevas desde CSS.

Siempre tengan presente que pueden incorporar sus propios estilos y sobrescribir, en caso de ser necesario. El uso de Bootstrap es el piso, el techo es el que ustedes quieran.

Grillas con Bootstrap (grid system)

Bootstrap nos provee un sistema de grillas muy potente para crear layouts responsive. Con un enfoque mobile-first vamos a poder crear diseños con formas muy variadas para cualquier tipo de dispositivo o pantalla. Lo lograremos gracias a su sistema de 12 columnas y 5 puntos de quiebre para trabajar el responsive con muchas clases predefinidas.

¿Cómo funciona?

El sistema de grillas de bootstrap usa una serie de contenedores, filas y columnas para organizar y alinear el contenido de nuestra página. Está basado en flexbox y es completamente responsive. Así que deberás estar familiarizado con flexbox antes de comenzar a explorar el sistema de grillas.

Veamos un primer ejemplo para apreciar lo sencillo que resulta trabajar con este sistema una vez que nos hayamos familiarizado con él.

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      La primera de tres columnas
    </div>
    <div class="col-sm">
      La segunda de tres columnas
    </div>
    <div class="col-sm">
      La tercera de tres columnas
    </div>
  </div>
</div>
```

Analicemos

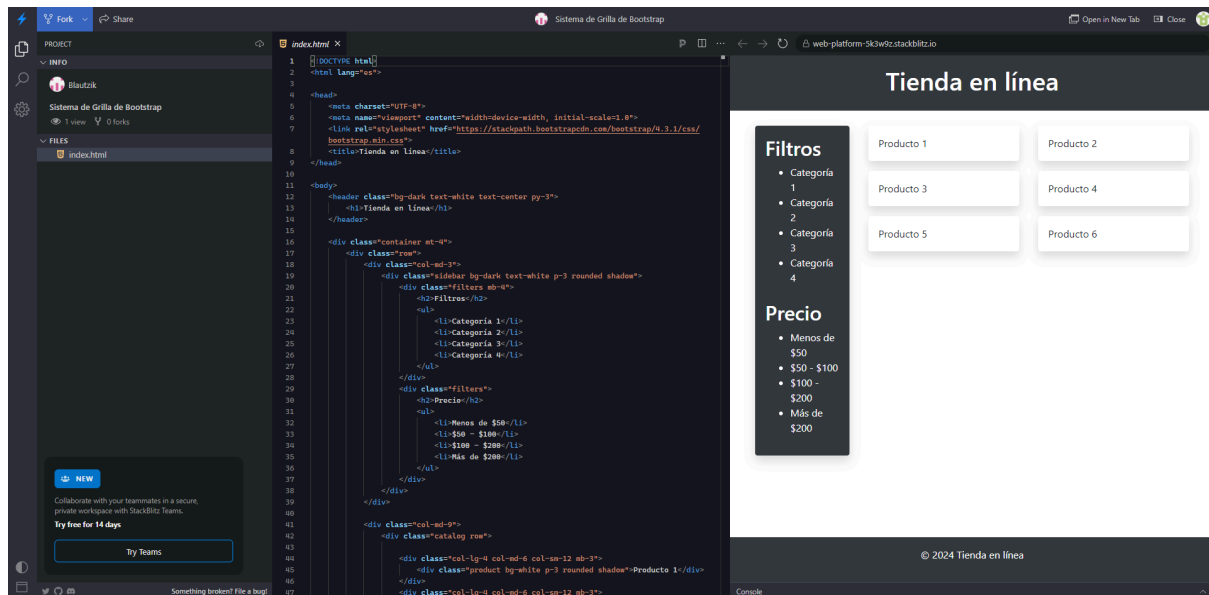
Analicemos lo que está sucediendo

- Las 3 columnas están contenidas dentro de un contenedor. Los contenedores nos ayudan a centrar y acomodar horizontalmente el contenido. Usamos la clase `.container` para tener un ancho dinámico y responsive manteniendo el contenido centrado. (para ocupar el 100% del ancho de la pantalla deberíamos usar la clase `.container-fluid`).
- La row envuelve las columnas. es comparable a la fila de una tabla. Cada row puede dividirse en 12 columnas
- El contenido debe ir siempre en columnas y solamente elementos que sean columnas pueden ser hijos directos de una row.
- Gracias a flexbox, las columnas que no tengan un ancho definido, automáticamente se van a transformar en columnas con el mismo ancho.
- Para hacer la grilla responsive contamos con 5 breakpoints (`xs`, para pantallas de smartphone (menos de 576px). `sm`, para dispositivos pequeños (más de 576px). `md`, para tablets (más de 768 px). `lg`, para notebooks o desktop (mas de 992px). `xl`, para pantallas extra grandes (más de 1200 px)
- Al tener 3 elementos con la clase `.col-sm` estamos indicando que cada elemento ocupará un ancho del 33% del espacio disponible a partir del breakpoint `sm`. es decir que en pantallas de más de 576px de ancho se verán separadas en 3 columnas pero en pantallas más pequeñas se verán una debajo de la otra.

Ejemplo

Les compartimos este ejemplo para que puedan ver cómo crear un layout similar al que hicimos en la clase de grillas utilizando solamente bootstrap

<https://stackblitz.com/edit/web-platform-5k3w9z?file=index.html>



Preguntas de reflexión

- ¿De qué manera puedo cambiar el tamaño de un elemento cuando el mouse para por encima?
- ¿Para qué sirve la propiedad transition?
- ¿Que es bootstrap y para que se utiliza?
- ¿Cómo podemos sumar componentes bootstrap a nuestro proyecto?



Desafío N° 8:

- Incorpora en tu proyecto al menos un componente de bootstrap
- Recomendamos un navbar o carousel.
- Aplica un efecto de hover a algunos de los elementos de tu página.

Ejemplo: Hacer que los productos se vean más grandes al pasar por encima de ellos con el mouse.

Recordá incluir los links de bootstrap para que funcione correctamente la librería.



Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad