



Laboratorio I - 2024



Clase Teórica 06

Docente: Myriam Ruiz

Licenciada en Informática

Profesora en Computación y Matemática

Programadora Universitaria





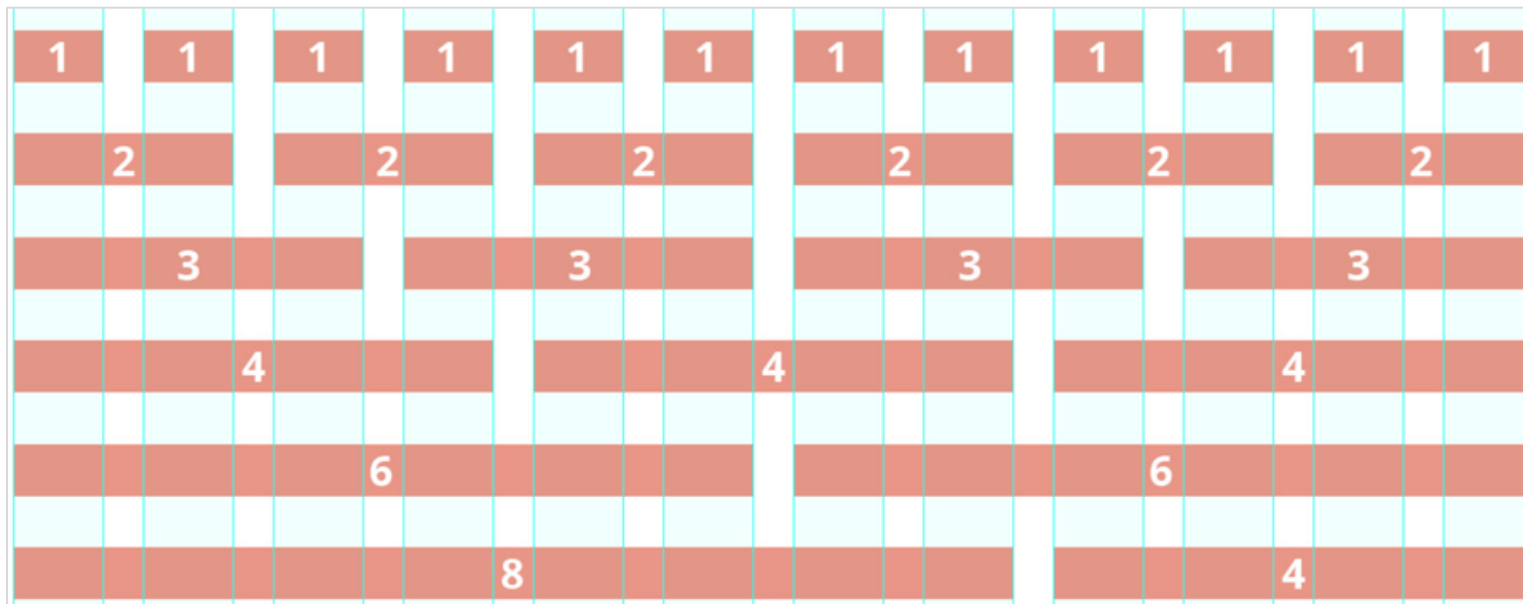
Introducción a Bootstrap

Características:

- **Biblioteca de CSS y Javascript, con código predefinido**
- **Creada por Twitter**
- **Acelera el proceso de desarrollo**
- **Pensada para Diseño Responsive**

Cómo funciona Bootstrap?

- Utiliza el concepto de grilla de 12 columnas, para maquetar



¿Cómo funciona Bootstrap?

- Todo se logra mediante la utilización de clases con nombres predefinidos, que se aplican dependiendo del tamaño de pantalla
- Algunos ejemplos de nombre de clase:

text-center: alinea al centro el contenido de un elemento

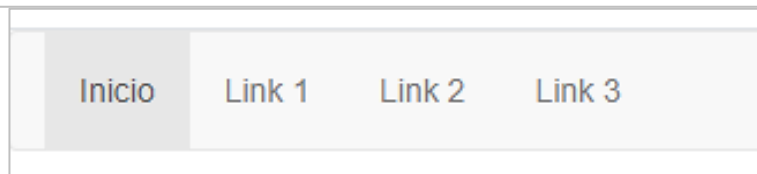
btn-success: muestra un botón verde

flex-column: muestra elementos en columna

¿Cómo funciona Bootstrap?

Por cada etiqueta html podemos introducir utilizar varias clases, separadas por espacio vacío:

```
<nav class="navbar navbar-default">
  <section class="container-fluid">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Inicio</a></li>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
    </ul>
  </section>
</nav>
```



Documentación sugerida

- Documentación oficial:

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

- Versión en castellano de la documentación oficial:

<https://getbootstrap.esdocu.com/docs/5.3/getting-started/introduction/>

- Lista de Propiedades CSS con ejemplos (inglés)

https://www.w3schools.com/bootstrap/bootstrap_ref_all_classes.asp

Como incorporar Bootstrap

- Existen 2 formas de incorporar Bootstrap:
- **Linkar el CDN (Content Delivery Network) provisto, que está online:**

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.c
ss" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.
min.js" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIds1K1eN7N6jIeHz"
crossorigin="anonymous"></script>
```

Descargar CSS y JS (recomendado para pruebas offline):

<https://getbootstrap.com/docs/5.3/getting-started/download/>

Como incorporar Bootstrap

- **Descargar CSS y JS (recomendado para pruebas offline):**
<https://getbootstrap.com/docs/5.3/getting-started/download/>
- Luego de descargar, descomprimos la carpeta y linkeamos en el head de todas las páginas el css minimizado (llamado así porque no tiene espacios entre líneas)

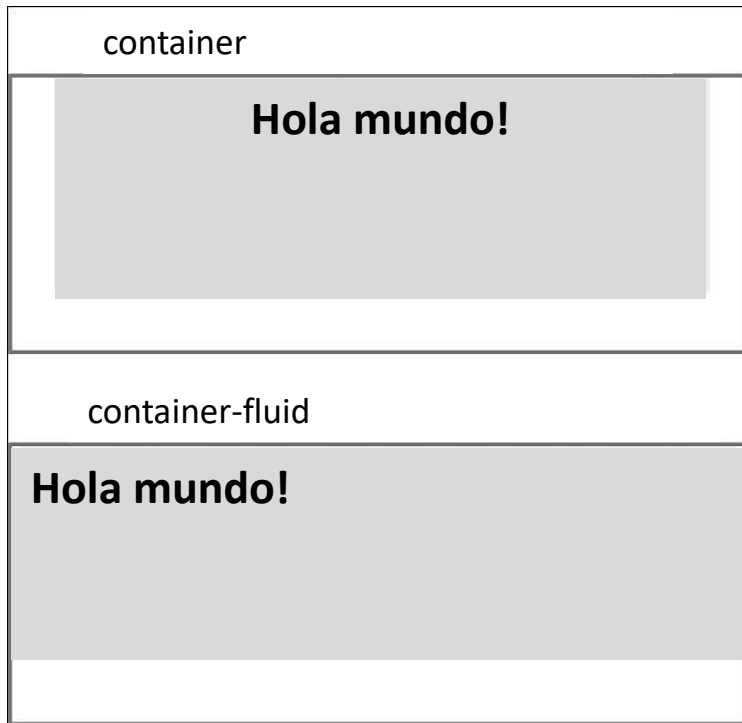
```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="author" content="Myriam Ruiz">
  <title>Primer Sitio con Bootstrap</title>
  <link rel="stylesheet" href="bootstrap-5.3.1-dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="css/style.css">
</head>
```



* Se puede linkear otro css si se desea trabajar en aspectos particulares

Maquetación con Bootstrap - contenedor

- Necesitamos tener un contenedor, que debe tener una de 2 clases posibles:



container: el contenido aparece centrado y con un ancho fijo.

container-fluid: el contenido ocupa todo el ancho disponible (100%)

¿Cómo funciona Bootstrap para ser Responsive?

- Para determinar el tamaño en el que aplicará una clase, lleva un infijo de tamaño, por ejemplo: col-**md**-6
- **md** indica que para el tamaño medium ocupará 6 columnas

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

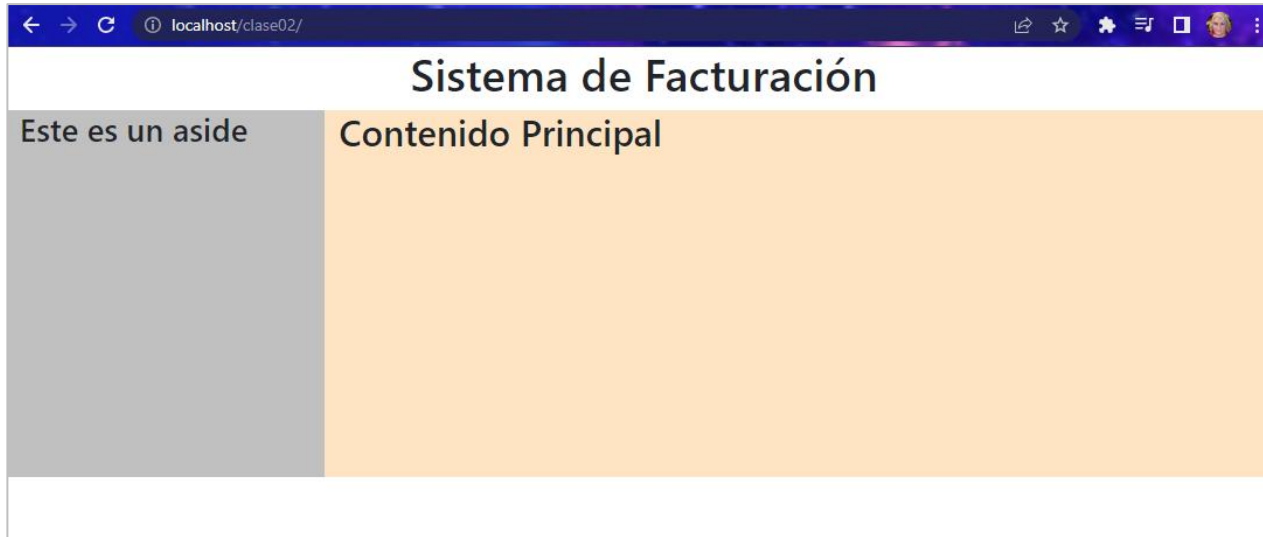
Maquetación con Bootstrap - elementos alineados

- Dentro un elemento con clase de contenedor, deberemos crear una fila (row) y dentro indicar cuantas columnas ocupará cada elemento, pudiendo poner más de una clase para distintos tamaños:

```
<section class="container-fluid">
  <section class="row"> <!-- creamos una fila -->
    <aside class="col-md-3 lateral"> <!-- establecemos el ancho en 3 columnas -->
      <h3>Este es un aside</h3>
    </aside>
    <main class="col-md-9 principal"> <!-- establecemos el ancho en 9 columnas -->
      <h2>Contenido Principal</h2>
    </main>
  </section>
</section>
```

Maquetación con Bootstrap - elementos alineados

- Resultado



Maquetación con Bootstrap - elementos alineados

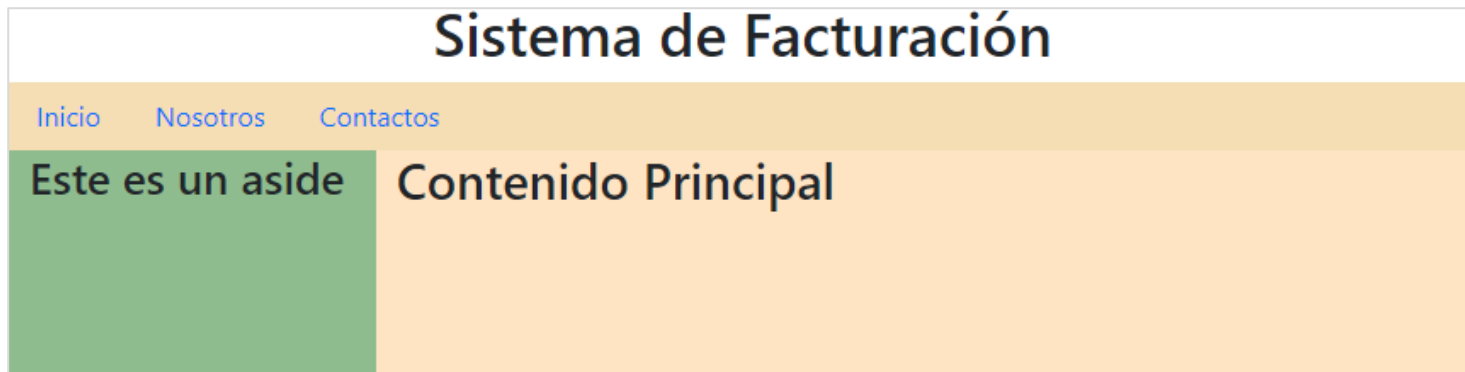
- Otros anchos:

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

Maquetación con Bootstrap - Menú

- Clases básicas para hacer un menú: nav, nav-item y nav-link

```
<nav>
  <ul class="nav">
    <li class="nav-item"><a href="#" class="nav-link">Inicio</a></li>
    <li class="nav-item"><a href="php/nosotros.php" class="nav-link">Nosotros</a></li>
    <li class="nav-item"><a href="php/contactos.php" class="nav-link">Contactos</a></li>
  </ul>
</nav>
```



Maquetación con Bootstrap - Menú responsive

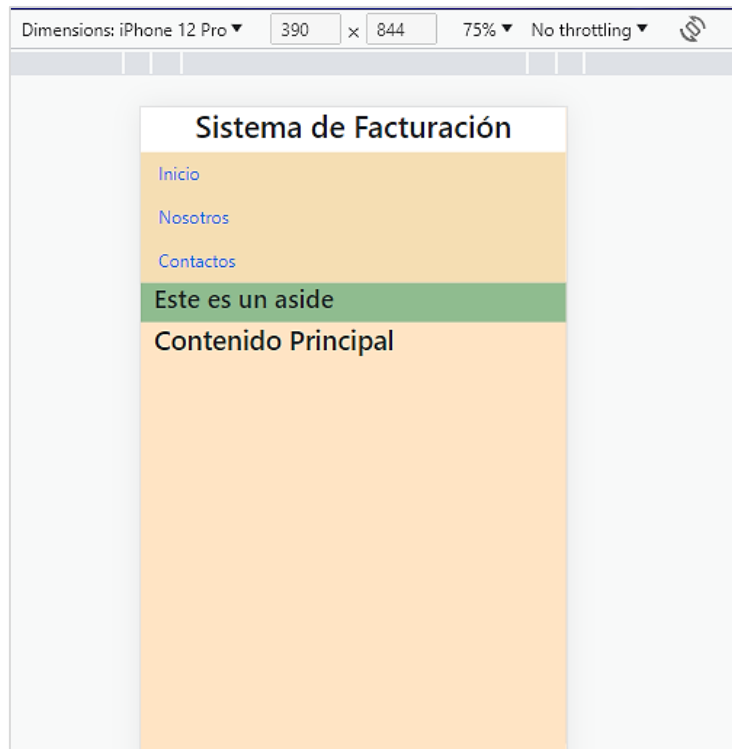
- Clases que aplican flexbox. Tener en cuenta que sin infijo de tamaño, es para el tamaño menor al primer breakpoint.

```
<nav>
  <ul class="nav flex-column flex-md-row">
    <li class="nav-item"><a href="#" class="nav-link">Inicio</a></li>
    <li class="nav-item"><a href="php/nosotros.php" class="nav-link">Nosotros</a></li>
    <li class="nav-item"><a href="php/contactos.php" class="nav-link">Contactos</a></li>
  </ul>
</nav>
```

- **flex-column:** muestra de forma vertical en tamaños pequeños
- **flex-md-row:** muestra de manera horizontal a partir del tamaño mediano

Maquetación con Bootstrap - Menú responsive

- En tamaño pequeño se vería:



Maquetación con Bootstrap - apariencia de botón

- Los botones de formulario y enlaces pueden lucir como un tipo de botón estandarizado: debemos decir que es un botón (btn) y luego el color (btn-primary)

```
<nav>
  <ul class="nav flex-column flex-md-row">
    <li class="nav-item"><a href="#" class="btn btn-primary">Inicio</a></li>
    <li class="nav-item"><a href="php/nosotros.php" class="btn btn-primary">Nosotros</a></li>
    <li class="nav-item"><a href="php/contactos.php" class="btn btn-primary">Contactos</a></li>
  </ul>
</nav>
```



- Existen otras clases para hacer menú, que se verán en próximas teorías

Maquetación con Bootstrap - apariencia de botón

- Más colores:



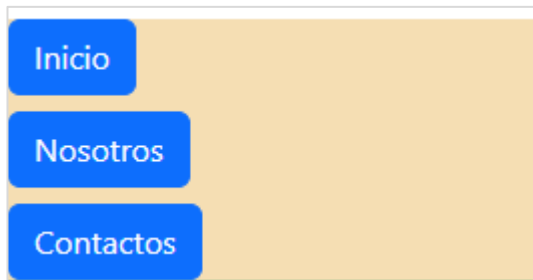
```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</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">Enlace</button>
```

Maquetación con Bootstrap - apariencia de botón

- Espaciado de botones (gap-número) y alineación vertical:

```
<nav>
  <ul class="nav flex-column gap-2 ">
    <li class="nav-item"><a href="#" class="btn btn-primary">Inicio</a></li>
    <li class="nav-item"><a href="php/nosotros.php" class="btn btn-primary">Nosotros</a></li>
    <li class="nav-item"><a href="php/contactos.php" class="btn btn-primary">Contactos</a></li>
  </ul>
</nav>
```



Maquetación con Bootstrap - apariencia de botón

- Para que todos los botones ocupen el mismo ancho (los *a* se puede mostrar como bloque, d-block):

```
<nav class="col-md-2">
  <ul class="nav flex-column gap-2">
    <li class="nav-item"><a href="#" class="btn btn-primary d-block">Inicio</a></li>
    <li class="nav-item"><a href="php/nosotros.php" class="btn btn-primary d-block">Nosotros</a></li>
    <li class="nav-item"><a href="php/contactos.php" class="btn btn-primary d-block">Contactos</a></li>
  </ul>
</nav>
```



Maquetación con Bootstrap - display

- Existen más tipos de display. Las clases se nombran usando el formato:
.d-{value} para xs
.d-{breakpoint}-{value} para sm, md, lg, xl, y xxl.
- Donde *value* es uno de:
 - none
 - inline
 - inline-block
 - block
 - grid
 - table
 - table-cell
 - table-row
 - flex
 - inline-flex

Clases Bootstrap para Tablas

<https://getbootstrap.esdocu.com/docs/5.3/content/tables/>

- Las siguiente clases se aplican a la etiqueta `<table class = "">` separadas por espacios

Clase de Bootstrap	Descripción
table	Aplica estilos básicos para crear una tabla con diseño limpio y responsivo.
table-bordered	Agrega bordes a las celdas y alrededor de la tabla.
table-striped	Alternancia de colores en filas para una mejor legibilidad.
table-hover	Resalta filas cuando el cursor pasa sobre ellas.
table-responsive	Hace que la tabla sea responsive, permitiendo el desplazamiento horizontal en pantallas pequeñas.
table-sm	Reduce el espacio entre celdas para crear una tabla más compacta.

Bootstrap para Tablas: table-bordered y table-hover

Ejemplo: bordes y hover

```
<table class="table table-bordered table-hover">
  <thead>
    <tr>
      <th>DNI</th>
      <th>Apellido</th>
      <th>Nombre</th>
      <th>Domicilio</th>
      <th>Teléfono</th>
      <th>Correo</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>12345678A</td>
      <td>Gómez</td>
      <td>Juan</td>
```



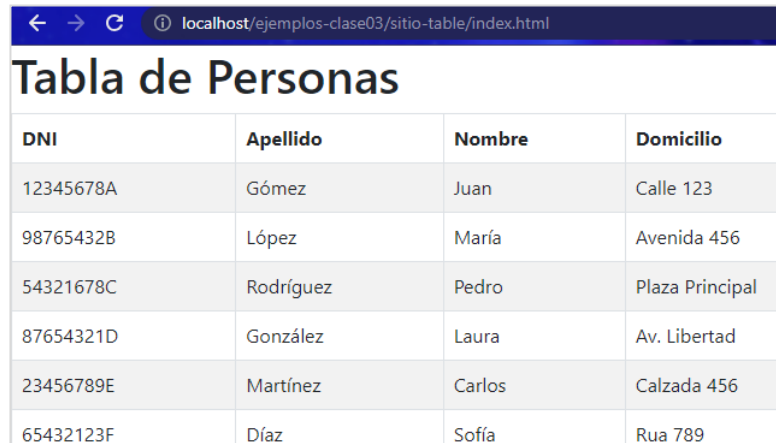
Tabla de Personas

DNI	Apellido	Nombre	Domicilio	Correo
12345678A	Gómez	Juan	Calle 123	
98765432B	López	María	Avenida 456	
54321678C	Rodríguez	Pedro	Plaza Principal	
87654321D	González	Laura	Av. Libertad	

Bootstrap para Tablas: table-striped

Ejemplo: color de fondo como "cebra", a rayas

```
<table class="table table-bordered table-hover table-striped">
  <thead>
    <tr>
      <th>DNI</th>
      <th>Apellido</th>
      <th>Nombre</th>
      <th>Domicilio</th>
      <th>Teléfono</th>
      <th>Correo</th>
    </tr>
  </thead>
```



localhost/ejemplos-clase03/sitio-table/index.html

Tabla de Personas

DNI	Apellido	Nombre	Domicilio
12345678A	Gómez	Juan	Calle 123
98765432B	López	María	Avenida 456
54321678C	Rodríguez	Pedro	Plaza Principal
87654321D	González	Laura	Av. Libertad
23456789E	Martínez	Carlos	Calzada 456
65432123F	Díaz	Sofía	Rua 789

Clases Bootstrap para Tablas

- Mas propiedades

Clase de Bootstrap	Descripción
table-dark	Estiliza con un fondo oscuro, tanto si lo usamos en las etiquetas table como en thead.
table-light	Estiliza con un fondo oscuro, tanto si lo usamos en las etiquetas table como en thead.
table-primary	Aplica un color de fondo primario.
table-secondary	Aplica un color de fondo secundario.
table-success	Aplica un color de fondo de éxito.
table-danger	Aplica un color de fondo de peligro.
table-warning	Aplica un color de fondo de advertencia.
table-info	Aplica un color de fondo de información.
text-center	Alinea el contenido de las celdas horizontalmente al centro.
text-start	Alinea el contenido de las celdas horizontalmente a la izquierda.
text-end	Alinea el contenido de las celdas horizontalmente a la derecha.

Clases Bootstrap para Tablas

- Colores de fondo

Clase	Título
Predeterminado	Celda
Primary	Celda
Secondary	Celda
Success	Celda
Danger	Celda
Warning	Celda
Info	Celda
Light	Celda
Dark	Celda

Clases Bootstrap para Tablas

- Más ejemplos

```
<table class="table table-bordered table-hover table-striped table-primary">
  <thead class="table-dark">
    <tr>
      <th>DNI</th>
      <th>Apellido</th>
      <th>Nombre</th>
      <th>Domicilio</th>
      <th>Teléfono</th>
      <th>Correo</th>
    </tr>
  </thead>
```

```
<tr>
  <td>98765432B</td>
  <td class="text-end">López</td>
  <td>María</td>
  <td>Avenida 456</td>
  <td>987-654-321</td>
  <td>marialopez@hotmail.com</td>
</tr>
```

localhost/ejemplos-clase03/sitio-table/index.html

Tabla de Personas

DNI	Apellido	Nombre	Domicilio
12345678A	Gómez	Juan	Calle 123
98765432B	López	María	Avenida 456
54321678C	Rodríguez	Pedro	Plaza Principal
87654321D	González	Laura	Av. Libertad

Archivo | D:/xampp23/htdocs/ejemplos-clase03/sitio-table/index.html

Tabla de Personas

DNI	Apellido	Nombre	Domicilio
12345678A	Gómez	Juan	Calle 123
98765432B	López	María	Avenida 456
54321678C	Rodríguez	Pedro	Plaza Principal

Bootstrap para Formularios

<https://getbootstrap.esdocu.com/docs/5.3/forms/overview/>

https://www.w3schools.com/bootstrap5/bootstrap_forms.php


Bootstrap para Formularios

Formulario sin clases

Pago de Compra (method="post")

Ingrese sus datos

Apellido:

Fecha Nacimiento: 

Cargar

Limpiar

Formulario sin clases form-label y form-control

Pago de Compra (method="post")

Ingrese sus datos

Apellido:

Fecha Nacimiento:



Cargar

Limpiar

Bootstrap para Formularios

Formulario utilizando clases form-label (agrega margen y permite alinear los input) y form-control (da un ancho uniforme, envía debajo los input y le da un color de borde)

```
<article class="col-md-3">
  <form action="php/procesar.php" method="get">
    <legend>Ingrese sus datos</legend>
    <section>
      <label for="ape" class = "form-label">Apellido: </label>
      <input type="text" name="apel" id="ape" class = "form-control">
    </section>
    <section>
      <label for="fecha" class = "form-label">Fecha Nacimiento:</label>
      <input type="date" name="fecnac" id="fecha" class = "form-control">
    </section>
    <section>
      <input type="submit" value="Cargar" class="btn btn-primary mt-3">
      <input type="reset" value="Limpiar">
    </section>
  </form>
</article>
```

Pago de Compra (method="post")

Ingrese sus datos

Apellido:

Fecha Nacimiento:

Bootstrap para Formularios

Hay clases para los distintos elementos de formulario:

Clase	Descripción
form-check	Utilizar en el section que contiene a label e input, para garantizar márgenes adecuados para etiquetas y casillas de verificación
form-check-input	Aplicada a casillas de verificación (<input type="checkbox">) y botones de opción (<input type="radio">) para darles estilo y permitir una alineación adecuada con sus etiquetas.
form-check-inline	Coloca los elementos checkbox o radio en la misma línea. Se usa en el section que contiene a label e input
form-range	Estiliza los controles deslizantes (<input type="range">)
form-select	Aplicada a elementos <select> para darles un estilo uniforme
form-switch	Utilizada junto con form-check, genera efecto toggle (interruptor)
shadow	sombra
border	agrega un borde


Bootstrap para Formularios

Hay clases para los distintos elementos de formulario:

Ingrese sus datos

Apellido:

Fecha Nacimiento:



Subscribir a:


☒ Cursos

☐ Webinars


Bootstrap para Formularios

Ejemplo: borde, sombra, checkbox "deslizable"

`<form action="php/procesar.php" method="get" class="border border-primary p-3 shadow">`



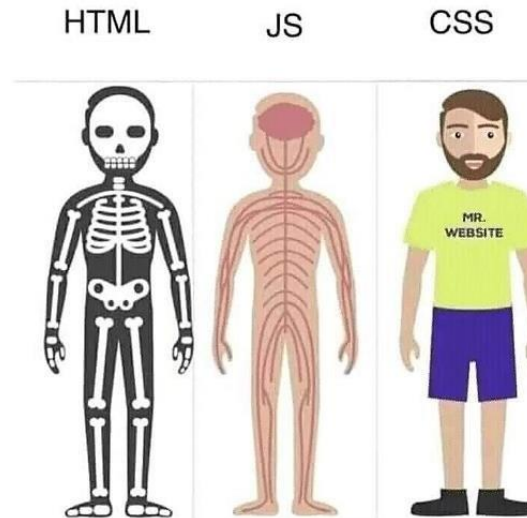
```
<fieldset>
  <legend>Subscribir a:</legend>
  <section class="form-check form-switch">
    <label for="cur">Cursos</label>
    <input type="checkbox" class="form-check-input" id="cur" name="cursos">
  </section>
  <section class="form-check form-switch">
    <label for="webi">Webinars</label>
    <input type="checkbox" class="form-check-input" id="webi" name="webinars">
  </section>
</fieldset>
```





Javascript

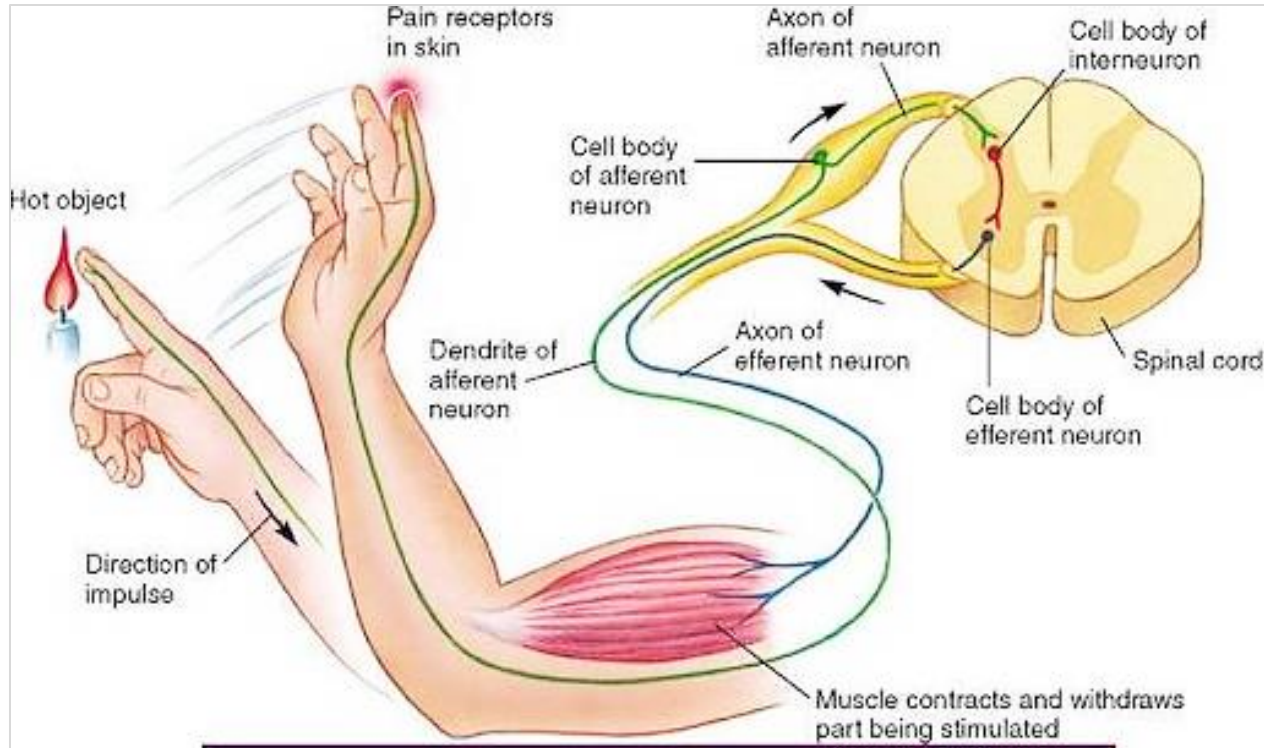
- JS (Javascript), permite que la página "reaccione" ante acciones producidas por los usuarios sobre esta.
- Es el "sistema nervioso" y los "músculos" de nuestra página.





JavaScript

- Es el "sistema nervioso" y los "músculos" de nuestra página.





JavaScript

- Es un lenguaje de programación, del lado del **cliente** (así se verá en Laboratorio I) y del servidor.
- Es un lenguaje ***interpretado*** por el navegador (al igual que html y css). Por tanto no se crea un ejecutable.
- Su sintaxis es similar a la de los lenguajes C y JAVA

Características de Javascript

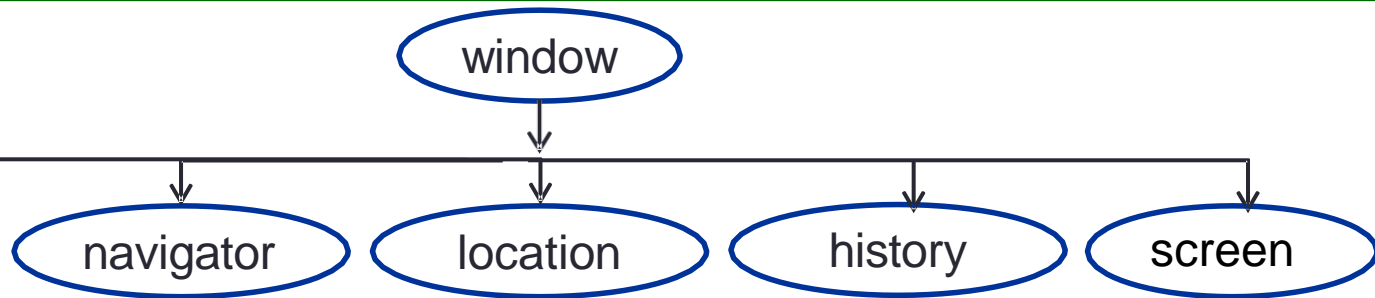
- Distingue mayúsculas y minúsculas (case sensitive).
Ej: miEdad lo considera diferente de miedad o MiEdad.
- En sus versiones más modernas soporta el Paradigma Orientado a Objetos. Versiones previas trabajaban con Prototipos (Clonación de elementos existentes).
- Es un lenguaje multiplataforma (funciona en cualquier dispositivo, sin importar el sistema operativo o hardware).

Conceptos básicos: BOM y DOM

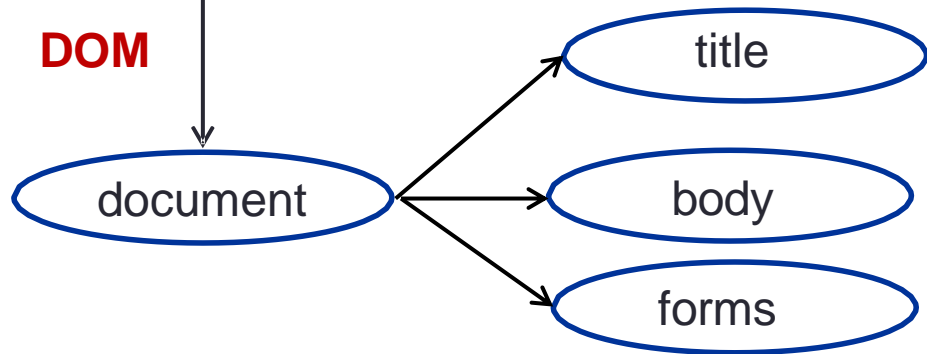
- **BOM** (Browser Object Model) o Modelo de Objetos del Navegador
 - Permite que JS acceda y modifique las propiedades de las **ventanas** del navegador.
- **DOM** (Document Object Model) o Modelo de Objetos del Documento
 - Es una interfaz que permite a JS el **acceso** y la **modificación**, de manera dinámica, del **contenido**, **estructura** y **CSS** de una página Web

Conceptos básicos:

BOM



DOM



Acceder a elementos: usar punto .



"Plantilla"

Clase Auto
+ color - puertas - marca - velocidadActual
+ obtenerMarca() + obtenerPuertas() + acelerar(cuanto) + frenar()

miAuto.color

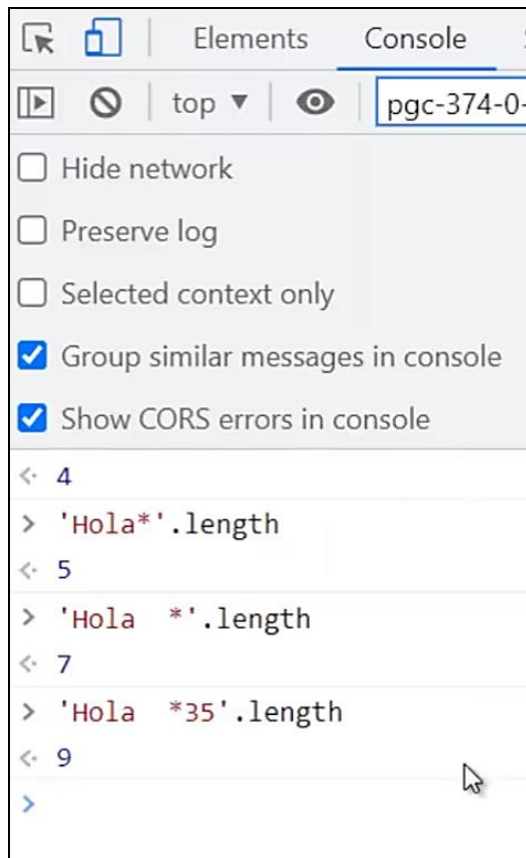
miAuto.frenar()

miAuto.obtenerMarca()

Acceder a elementos: usar punto .

"Plantilla"

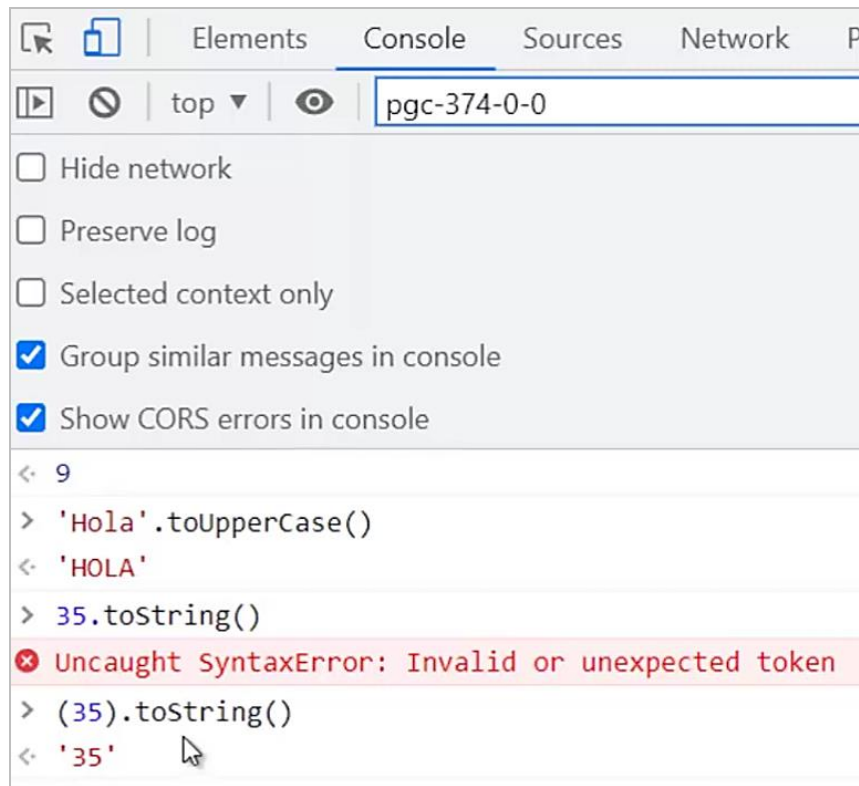
Clase string
+ length
+ toUpperCase() + toLowerCase() + concat(texto2) + toString()



Acceder a elementos: usar punto .

"Plantilla"

Clase string
+ length
+ toUpperCase() + toLowerCase() + concat(texto2) + toString()



Conceptos básicos: Acceder a elementos del DOM

- Forma de acceder a elementos del DOM:

- A través del punto

Ej: Para acceder al Title de la página: `document.title = 'Nuevo título';`

- Mediante la obtención de id

`elemento = document.getElementById('un-id');`

`elemento.textContent = 'Nuevo texto';`

- Mediante la obtención de class

`elemento = document.getElementsByClassName('una class')`

`elemento[0].textContent = 'Nuevo texto';`

Conceptos básicos: Acceder a elementos del DOM

- Mediante la obtención de class: Debido a que una clase se puede utilizar en más de un lugar, cuando obtenemos elementos por su nombre de clase, recibimos como resultado una matriz de los elementos que tienen esa clase.

```
<section>
  <article>
    <h2 class="titulo">Titulo nivel 2 - 1 </h2>
    <p>Párrafo 1</p>
    <h3 class="titulo">Titulo nivel 3 - 1</h3>
  </article>
  <article>
    <h2 class="titulo">Titulo nivel 2 - 2 </h2>
    <p>Párrafo 2</p>
  </article>
  <article>
    <h2 class="titulo">Titulo nivel 2 - 3</h2>
    <p>Párrafo 3</p>
  </article>
```

```
let titulos = document.getElementsByClassName('titulo');

/* titulos tiene almacenado:
| [h2.titulo, h3.titulo, h2.titulo, h2.titulo]
*/
```

Conceptos básicos: Acceder a elementos del DOM

- Accedemos con corchete a cada elemento de la matriz, empezando por cero

```
let titulos = document.getElementsByClassName('titulo');  
  
/* titulos tiene almacenado:  
| [h2.titulo, h3.titulo, h2.titulo, h2.titulo]  
*/  
  
titulos[0].style.backgroundColor = '#0000FF'; // azul  
titulos[1].style.backgroundColor = '#FF0000'; // rojo  
titulos[2].style.backgroundColor = '#00FF00'; // verde
```

← → ↻ ⓘ Archivo | D:/Laboratorio%20I/2023/Clase07/ejemplo10/index.html

Ejemplo de getElementByClassName

Título nivel 2 - 1

Párrafo 1

Título nivel 3 - 1

Título nivel 2 - 2

Párrafo 2

Título nivel 2 - 3

Párrafo 3

Conceptos básicos: Comentarios

- **Comentarios:** En JS los comentarios se pueden escribir de 2 maneras:

1)

// mensaje en una línea

2)

/* mensaje en
múltiples
líneas */

Conceptos básicos:

- **Variable:** Un espacio de memoria al cual se le asignó un nombre y que contiene datos.
 - Podemos imaginarla como una caja, que posee un rótulo y que sirve para guardar números, letras, etc.
 - En algoritmos, se suele indicar guardar datos con el símbolo flecha.
 - Ej: edad \leftarrow 19 (la flecha indica llevar de derecha a izquierda el valor 19)



Conceptos básicos:

Tipos de datos

- Los tipos de datos trabajados en los Algoritmos (PU y Lic) o Diagramas de Flujo (Ing), no se utilizan al declarar una variable en JS. Se trata de un lenguaje ***débilmente tipado***.
- Para declarar una variable se escribe:
let miVariable;

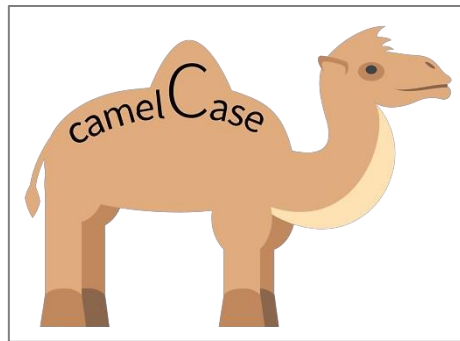
Conceptos básicos:

Tipos de datos

- Sin embargo, al guardar un dato en una variable, esta toma el tipo de dato correspondiente. Se conoce como ***tipado dinámico***.
- Para asignar datos en una variable, en JS utilizaremos el signo igual "=" en lugar del símbolo flecha.
 - Ej: let miVariable1 = 25; // tiene un tipo dato *número entero*
 - Ej: let miVariable2 = 'FACET'; // tiene un tipo de dato string (*tira de caracteres*)

Buenas prácticas

- Escribir los nombres compuestos (de más de una palabra) con la forma **camelCase**. Ej: miNombreCompleto

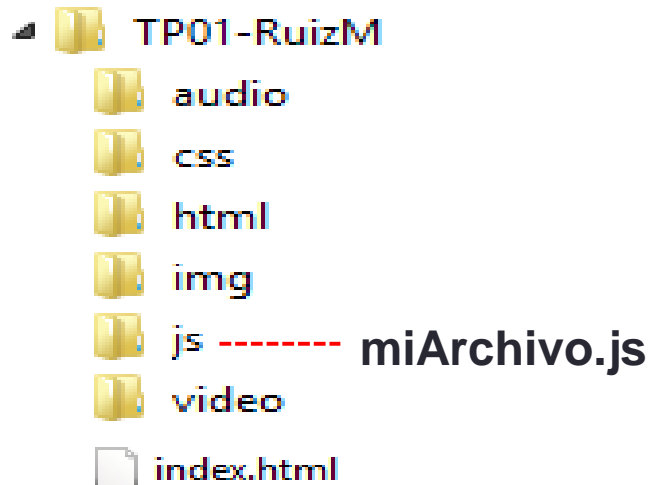


- Escribir una sola instrucción por línea.
- Terminar cada instrucción con ;

¿Cómo Incorporar JS?


- Similar a CSS, trabajaremos sólo con **archivos js externos**, aunque los scripts (códigos JS) podrían estar escritos dentro de la página html entre etiquetas `<script></script>` o en un evento.
- Los archivos js pueden incorporarse en el head o en el body
- Si el script JS requiere seleccionar elementos de la página html, se recomienda incluir el archivo al final del html, **antes de cerrar body**, de modo que todos los elementos se hayan creado.

¿Cómo utilizar JS?



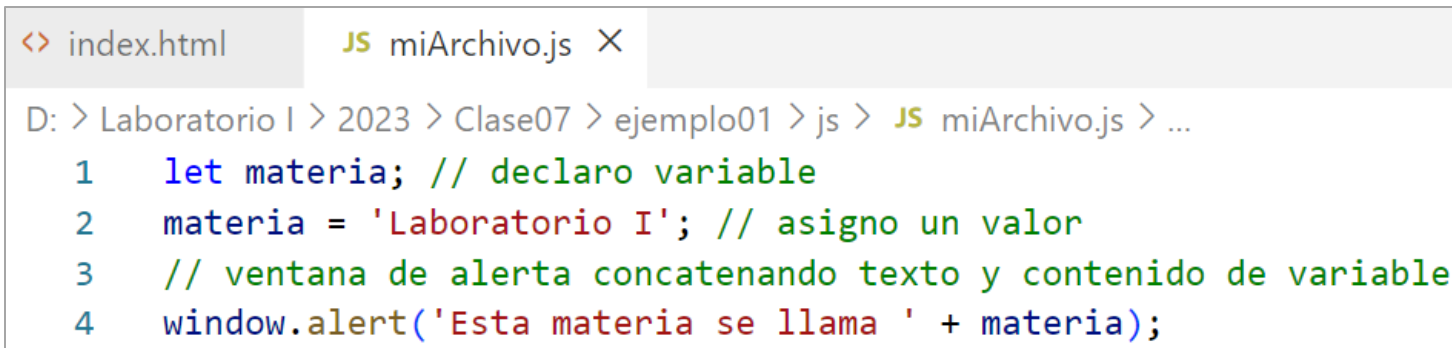
Estando en index.html

```
...  
<head>  
  <script src="js/miArchivo.js">  
  </script>  
</head>  
<body>  
...  
  <script src="js/miArchivo.js">  
  </script>  
</body>  
</html>
```



Qué escribir en el archivo JS

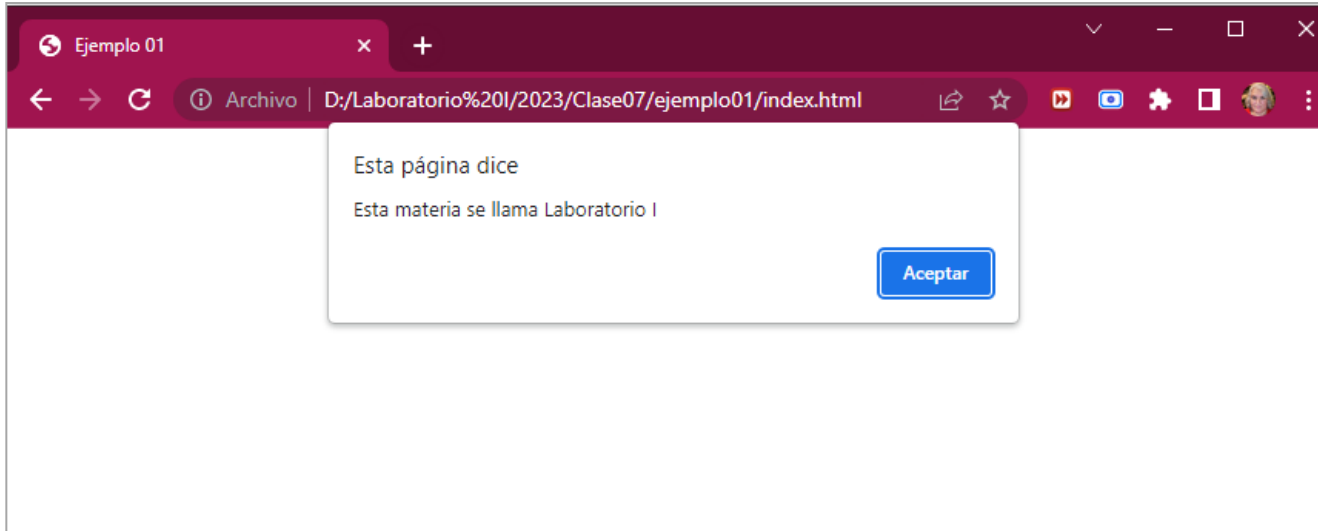
- En el archivo .js se escriben las instrucciones del lenguaje, no pudiendo escribir código html o css "suelto", es decir, que no sea manejado por código JS.
- ejemplo01:



```
<> index.html    JS miArchivo.js X
D: > Laboratorio I > 2023 > Clase07 > ejemplo01 > js > JS miArchivo.js > ...
1  let materia; // declaro variable
2  materia = 'Laboratorio I'; // asigno un valor
3  // ventana de alerta concatenando texto y contenido de variable
4  window.alert('Esta materia se llama ' + materia);
```

Qué escribir en el archivo JS

- ejemplo01:



Funciones

- Si escribimos todo el código que necesitamos en nuestro sitio, una instrucción debajo de la otra como en el ejemplo01, estas se ejecutan por completo, una sola vez, al incorporar el archivo .js.
- Debido a esto, es necesario poder separar los códigos e ir llamándolos sólo cuando sea necesario utilizarlos. Esto se logra a través del uso de **funciones** definidas por nosotros.

Funciones

- Una **Función** es un conjunto de instrucciones que realiza una tarea específica, que tiene asociado un nombre y que puede o no recibir datos (llamados parámetros) para realizar su tarea.
- En forma general:

```
function nombreFuncion (parámetros) {  
    // código  
    return (resultado); // si es que devuelve algún valor  
}
```


Funciones

- Ejemplo02: incorporar un script js que calcule el área de un triángulo

```
<> index.html X
D: > Laboratorio I > 2023 > Clase07 > ejemplo02 > <> index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
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=1.0">
7      <title>Ejemplo 02</title>
8  </head>
9  <body>
10     <script src="js/miArchivo.js">
11     </script>
12 </body>
13 </html>
```

Funciones

- ejemplo02

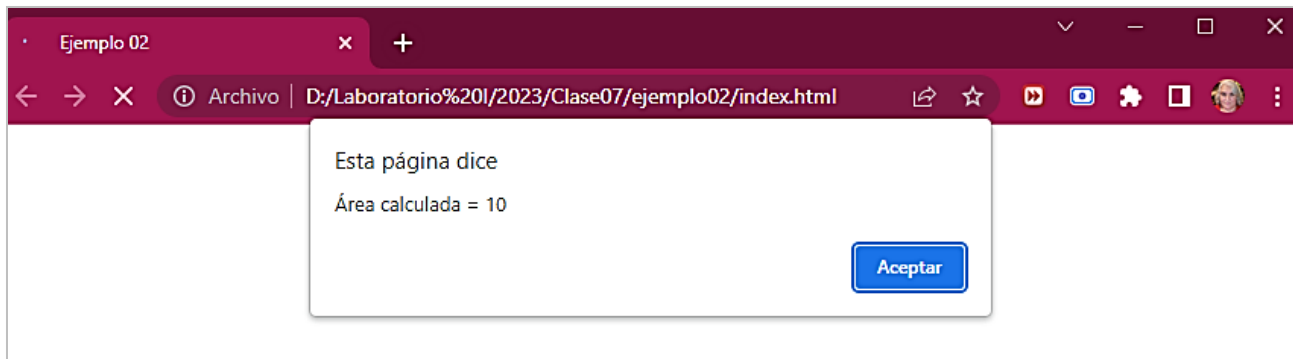
JS miArchivo.js X

D: > Laboratorio I > 2023 > Clase07 > ejemplo02 > js > JS miArchivo.js > ...

```
1  function areaTriangulo (b, h) {  
2      |      return (b * h / 2); // devuelve el área calculada  
3  }  
4  
5  // Probamos la función  
6  area = areaTriangulo (4,5);  
7  window.alert('Area calculada = ' + area);
```

Funciones

- ejemplo02



Si bien llamamos a la función, sigue activándose apenas cargamos la página. Veamos a continuación como llamar a una función sólo en determinadas situaciones.

Eventos

- Cada vez que un usuario interactúa con nuestro sitio, se produce lo que se llama un **Evento**. Ejemplos de esto son:
 - Hacer click en un botón
 - Escribir en un elemento de formulario, produciendo un cambio en su contenido.
 - Acercar y retirar el mouse sobre un enlace o una imagen
 - Etc.

Eventos que reconoce Js

Eventos del teclado:

Evento	Descripción	Elementos para los que está definido
keydown	Pulsar una tecla y no soltarla	Elementos de formulario y <body>
keyup	Soltar una tecla pulsada	Elementos de formulario y <body>
keypress	Pulsar una tecla (implica un keydown seguido de un keyup)	Elementos de formulario y <body>

Eventos del body:

Evento	Descripción	Elementos para los que está definido
load	Cuando la página se carga completamente	<body>
unload	Se abandona la página, por ejemplo al cerrar el navegador	<body>
resize	Cuando se cambia el tamaño del navegador	<body>

Eventos que reconoce Js

Eventos del mouse:

Evento	Descripción	Elementos para los que está definido
click	Pulsar y soltar el botón mouse	Todos los elementos
dblclick	Pulsar dos veces seguidas con el mouse	Todos los elementos
mousedown	Pulsar un botón del mouse y no soltarlo	Todos los elementos
mouseup	Soltar el botón del mouse	Todos los elementos
mousemove	Mover el mouse sobre algo	Todos los elementos
mouseout	El mouse "sale" del elemento	Todos los elementos
mouseover	El mouse "entra" en el elemento	Todos los elementos

Eventos que reconoce Js

Eventos de formulario:

Evento	Descripción	Elementos para los que está definido
blur	Un elemento pierde el foco	<button>, <input>, <label>, <select>, <textarea>, <body>
change	Un elemento ha sido modificado	<input>, <select>, <textarea>
focus	Un elemento obtiene el foco	<button>, <input>, <label>, <select>, <textarea>, <body>
reset	Inicializar el formulario	<form>
select	Seleccionar un texto	<input>, <textarea>
submit	Enviar el formulario	<form>

Reaccionar a Eventos

- Existen varias maneras para que el código JS se entere de que está sucediendo un evento. En esta materia se trabajará con la forma más moderna: "Escuchar" Eventos.
- La instrucción que permite escuchar eventos tiene la forma:

elemento.addEventListener('evento', funciónALlamar);

elemento: es sobre quien se estará escuchando

Reaccionar a eventos

- Ejemplo03: 'escuchar' el botón con id 'miBoton', para que cuando se le suceda el evento 'click' muestre una ventana diciendo 'Presionaste el botón'

Reaccionar a eventos

- Ejemplo03:

```
<body>
  <header><h1>Ejemplo con botón</h1></header>
  <main>
    <section>
      <article>
        <form action="../../../php/procesar.php" method="post">
          <input type="button" id="miBoton" value="Presionar">
        </form>
      </article>
    </section>
  </main>
  <script src="js/miArchivo.js"></script>
</body>
```

Reaccionar a eventos

- Ejemplo03:

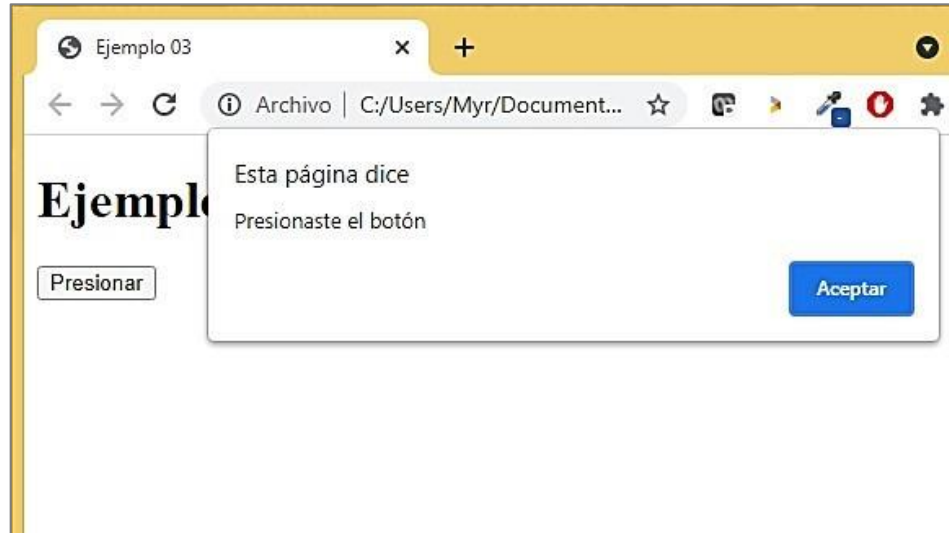
JS miArchivo.js ✕

D: > Laboratorio I > 2023 > Clase07 > ejemplo03 > js > JS miArchivo.js > ...

```
1  let boton = window.document.getElementById('miBoton');
2  boton.addEventListener('click', miFuncion);
3
4  function miFuncion() {
5      window.alert('Presionaste el botón');
6  }
```

Reaccionar a eventos

- Ejemplo03:



Modificación de CSS desde JS

- JS accede a todas las propiedades vistas en CSS, pero las interpreta convirtiendo su nombre compuesto, eliminando todos los guiones medios (-) y escribiendo en mayúscula la letra siguiente a cada guión medio.
- Ejemplos:
 - font-family se transforma en fontFamily
- Forma de modificar el css:
 - `window.document.elemento.style.propiedad = Valor;`

Modificar css mediante radio

- Ejemplo04: Crear un script que permita activar y desactivar el Modo Oscuro al seleccionar un input de tipo radio. Se debe cambiar el css del body.



Modificar css mediante radio

- Ejemplo04:

```
<article>
  <form action="#" method="GET">
    <label for="btnOscuro">Modo Oscuro</label>
    <input type="radio" id="btnOscuro" name="modo">
    <label for="btnClaro">Modo Claro</label>
    <input type="radio" id="btnClaro" name="modo">
  </form>
</article>
<article>
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing
elit. Perspiciatis accusantium doloribus minima eius
tempore architecto quidem, officia mollitia modi commodi!
Mollitia, enim voluptas neque pariatur dolores dicta
voluptate vel iste?</p>
</article>
```

Modificar css mediante radio

- Ejemplo04:

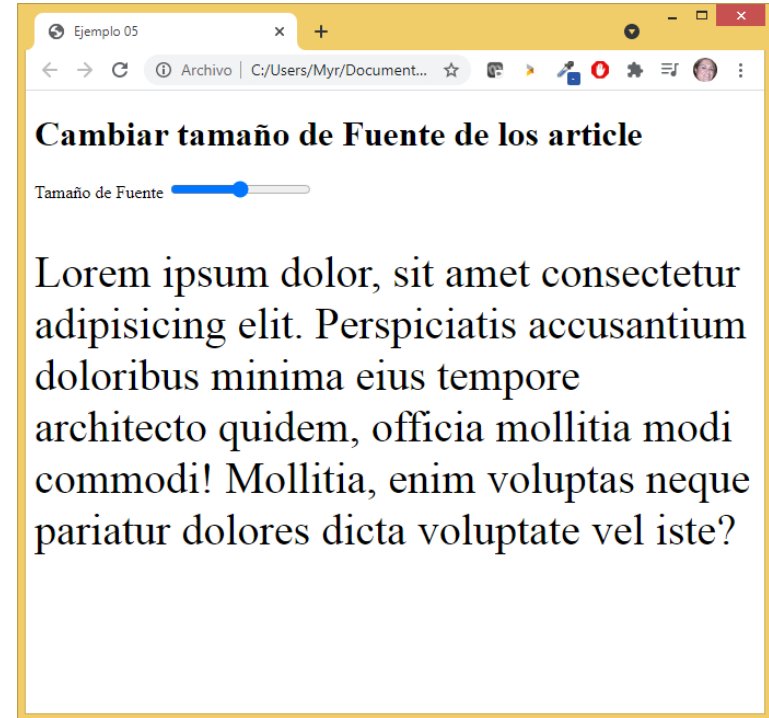
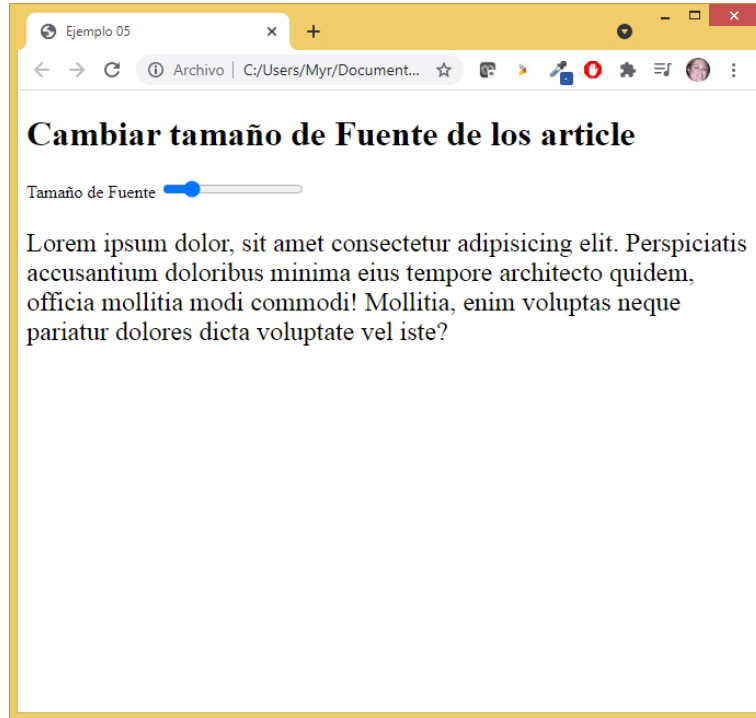
```
JS miArchivo.js X
D: > Laboratorio I > 2023 > Clase07 > ejemplo04 > js > JS miArchivo.js > ...
1  function modoOscuro () {
2      document.body.style.backgroundColor = '#000000';
3      document.body.style.color = '#FFFFFF';
4  }
5
6  function modoClaro () {
7      document.body.style.backgroundColor = '#FFFFFF';
8      document.body.style.color = '#000000';
9  }
10
11 let botonOscuro = document.getElementById('btnOscuro');
12 let botonClaro = document.getElementById('btnClaro');
13
14 botonOscuro.addEventListener('change', modoOscuro);
15 botonClaro.addEventListener('change', modoClaro);
```


Ejemplo: Cambiar tamaño fuente con range

- Ejemplo05: Crear un script que permita aumentar y disminuir el tamaño de la fuente de un párrafo, mediante un input de tipo range.

Ejemplo: Cambiar tamaño fuente con range

- Ejemplo05:



Ejemplo: Cambiar tamaño fuente con range

- Ejemplo05:

```
<article>
  <form action="#" method="GET">
    <label for="tama">Tamaño de Fuente</label>
    <input type="range" id="tama" min="1" max="4" step
      ="0.5" value="2" name="tama">
  </form>
</article>
<article>
  <p id="parrafo">Lorem ipsum dolor, sit amet consectetur
    adipisicing elit. Perspiciatis accusantium doloribus
    minima eius tempore architecto quidem, officia mollitia
    modi commodi! Mollitia, enim voluptas neque pariat
    dolores dicta voluptate vel iste?</p>
</article>
</section>
```

Ejemplo: Cambiar tamaño fuente con range

- Ejemplo05:

miArchivo.js ×

```
r > Documents > Laboratorio I 2017 > 2021 > ejemplos-semana-10 > ejemplo05 > js >  
1   let cambio = window.document.getElementById('tama');  
2   cambio.addEventListener('input', cambiaFuente);  
3  
4   function cambiaFuente() {  
5       let miTexto = window.document.getElementById('parrafo');  
6       miTexto.style.fontSize = cambio.value + "em";  
7   }
```

Escribir en la página desde JS

- Para escribir en la página, se puede acceder así:

`document.getElementById('un id').innerHTML = 'texto';`

- Si fuera un elemento de formulario:

`document.getElementById('un id').value = 'texto';`

Leer datos de un Formulario

- **Importante**: Todo valor ingresado en un **input**, aunque sea un número, para JS es **texto**, por tanto debe convertirse a número.

- Para obtener el contenido escrito en el input se trabaja así:

```
let dato = window.document.getElementById('valor1').value;
```

- Convertir a entero:

```
let numEntero = parseInt(dato);
```

- Convertir a real:

```
let numReal = parseFloat(dato);
```

Procesar datos de un Formulario

- Para poder validar o utilizar los datos ingresados en un formulario, se puede acceder a cada elemento de este por su id, teniendo el cuidado de "**prevenir**" que el formulario haga la limpieza de los datos cuando hace el submit.

Procesar datos de un Formulario

- ejemplo06:

```
7 > ejemplo06 > <> index.html > html > body > section > main
<form action="#" method="get" id="formu">
  <section>
    <label for="num1">Numero 1</label>
    <input type = "number" id = "num1" name = "n1">
  </section>
  <section>
    <label for="num2">Numero 2</label>
    <input type = "number" id = "num2" name = "n2">
  </section>
  <section>
    <label for="solu">Solucion</label>
    <input type = "number" id = "solu" name = "solucion">
  </section>
  <section>
    <input type="submit" value="Sumar">
  </section>
</form>
```


Procesar datos de un Formulario

- ejemplo06:

```
<> index.html    JS miArchivo.js X
D: > Laboratorio I > 2023 > Clase07 > ejemplo06 > js > JS miArchivo.js > ...
1  let miForm = document.getElementById('formu');
2  miForm.addEventListener('submit', procesar);
3
4  function procesar (event) {
5      event.preventDefault(); // impide que se borren los datos
6      let valor1 = document.getElementById('num1');
7      let valor2 = document.getElementById('num2');
8      let solucion = document.getElementById('solu');
9      solucion.value = parseInt(valor1.value) + parseInt(valor2.value);
10 }
```

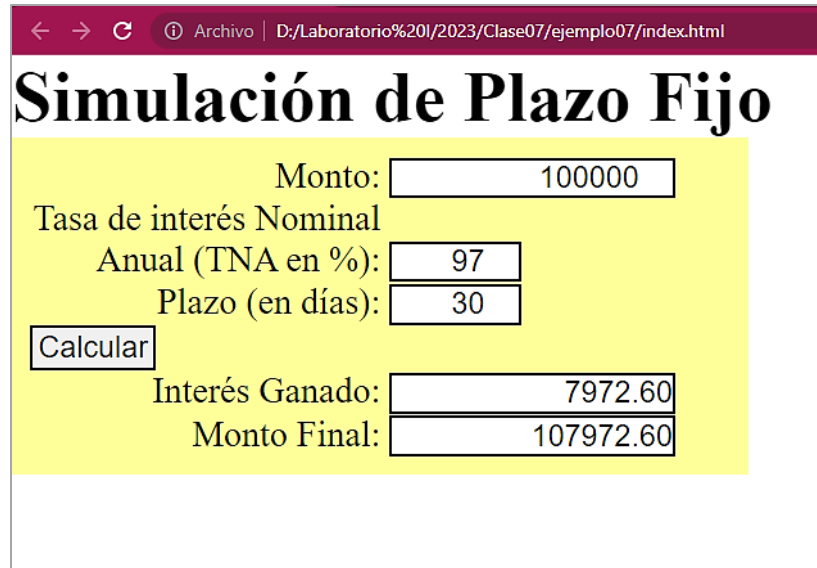
Procesar datos de un Formulario

- Cuando se utiliza `addEventListener` para registrar un controlador de eventos, el navegador pasa *automáticamente* un objeto **event** como argumento a la función del controlador.
- En nuestro caso, el evento es el envío del formulario (submit).
- El objeto event contiene información y métodos relacionados con el evento en sí.

preventDefault(): Es un método que evita el comportamiento predeterminado del evento. En el caso del evento submit del formulario, el comportamiento predeterminado es enviar los datos y recargar la página.

Procesar datos de un Formulario

- ejemplo07: Realizar un script que haga la simulación del cálculo de un plazo fijo



← → ↻ ⓘ Archivo | D:/Laboratorio%20I/2023/Clase07/ejemplo07/index.html

Simulación de Plazo Fijo

Monto:

Tasa de interés Nominal Anual (TNA en %):

Plazo (en días):

Interés Ganado:

Monto Final:

Procesar datos de un Formulario

- ejemplo07: html

```
<form action="#" method="post">
  <section>
    <label for="lblMonto">Monto:</label>
    <input type="number" id="lblMonto" name="monto" required>
  </section>
  <section>
    <label for="lblTasa">Tasa de interés Nominal Anual (TNA en %):</label>
    <input type="number" id="lblTasa" name="tasa" required>
  </section>
  <section>
    <label for="lblPlazo">Plazo (en días):</label>
    <input type="number" id="lblPlazo" name="plazo" min="30" required>
  </section>
</form>
```

Procesar datos de un Formulario

- ejemplo07: html

```
<input type="button" value="Calcular" id="botonCalcular">

<section>
  <label for="lblInteres">Interés Ganado:</label>
  <input type="text" id="lblInteres" name="interes" readonly>
</section>

<section>
  <label for="lblMontoFinal">Monto Final:</label>
  <input type="text" id="lblMontoFinal" name="montoFinal" readonly>
</section>
</form>
```

Procesar datos de un Formulario

- ejemplo07: js

JS simular.js ×

D: > Laboratorio I > 2023 > Clase07 > ejemplo07 > js > JS simular.js > ...

```
1  let boton = window.document.getElementById('botonCalcular');
2  boton.addEventListener('click', calcularPlazoFijo);
3
4  function calcularPlazoFijo() {
5      // Obtener los valores de los campos del formulario
6      let monto = parseFloat(document.getElementById("lblMonto").value);
7      let tasa = parseInt(document.getElementById("lblTasa").value);
8      let plazo = parseInt(document.getElementById("lblPlazo").value);
9
10     // Calcular el interés ganado
11     let interes = (monto * tasa * plazo) / (365 * 100);
12
13     // Calcular el monto final
14     let montoFinal = parseFloat(monto) + parseFloat(interres);
15 }
```

Procesar datos de un Formulario

- ejemplo07: js

```
16 // Mostrar los resultados en los campos correspondientes
17 document.getElementById('lblInteres').value = interes.toFixed(2);
18 document.getElementById('lblMontoFinal').value = montoFinal.toFixed(2);
19 }
```



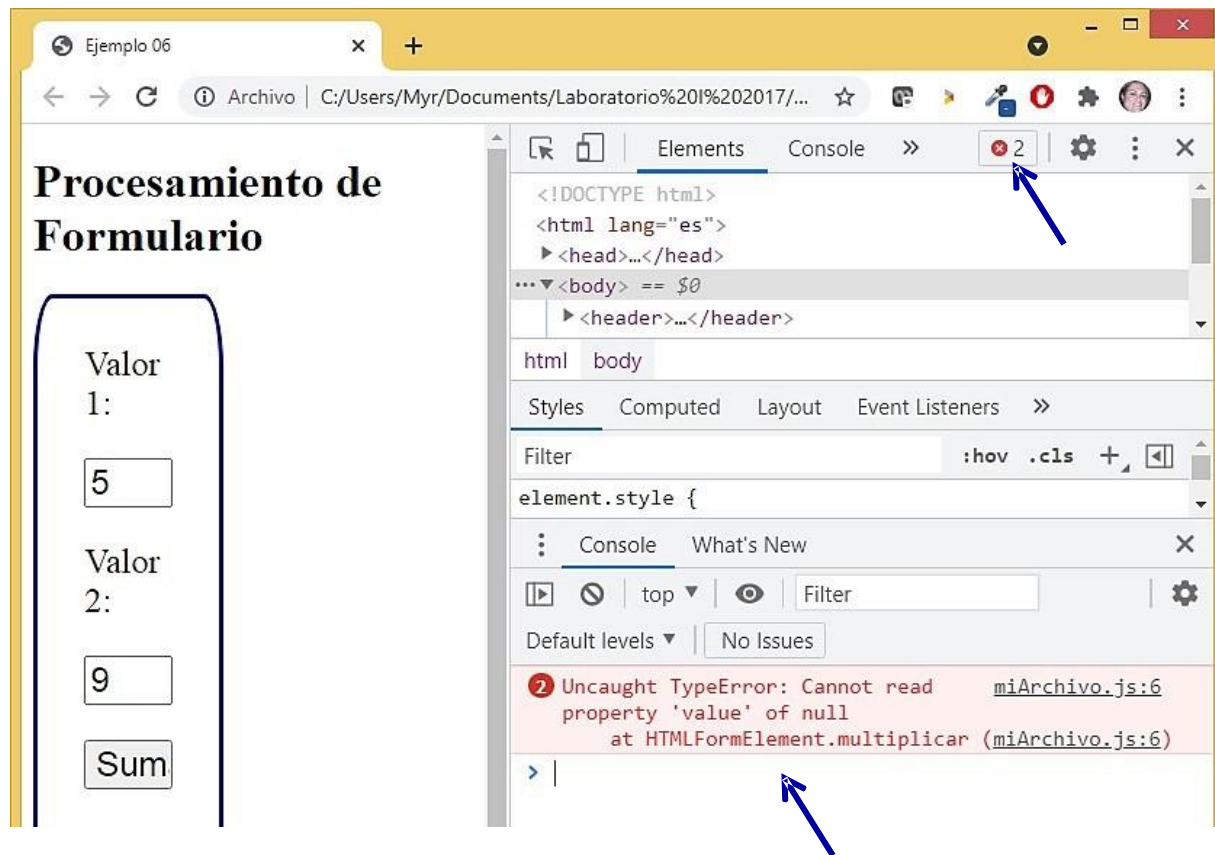
Para mostrar con 2 decimales

Cómo saber si tenemos errores en JS

- Cada Navegador permite ver los errores que tenemos al escribir código JS.
- Se conocen como Herramientas de Desarrolladores o Developers Tools
- Para acceder usaremos la misma combinación de teclas que para ver los sitios en tamaño para móviles:
 - Las versiones actuales de Edge, Chrome y Mozilla: Presionar F12
 - Otras combinaciones Ctrl + Shift + i

Cómo saber si tenemos errores en JS

En Chrome



Cómo saber si tenemos errores en JS

En Mozilla

