

MAKE  **REAL**
CAMP

Javascript en el Navegador

Se puede trabajar Javascript en un entorno en línea como repl.it, codesandbox, codepen o stackblitz; o mejor aún, desde tu entorno local directamente en el navegador web.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prueba de JavaScript</title>
</head>
<body>

  <script>
    // acá va el código que vamos a escribir
  </script>
</body>
</html>
```

```
<script src="app.js"></script>
```

Manipulación DOM

Utilizando el objeto document (o window.document) es posible:

- Obtener o modificar el título del documento.
- Obtener información de los elementos.
- Insertar, eliminar, reemplazar, ocultar y mostrar elementos.
- Agregar y eliminar clases de CSS.
- Manipular el CSS de los elementos directamente.

```
console.log(document.title);
```

```
document.title = "Nuevo título";
```

```
// retorna el formulario con id my-form  
var form = document.querySelector("form#my-form");  
form.tagName; // "form"  
form.id; // "my-form"  
form.style; // retorna un objeto con los estilos del elemento  
form.attributes; // retorna un objeto con los atributos del elemento
```

```
// retorna todos los div del documento  
var divs = document.querySelectorAll("div");  
for(var div of divs.entries()) {  
  console.log(div);  
}
```

```
var element = document.querySelector("div");  
element.parentNode.removeChild(element);
```

```
var div = document.querySelector("div");  
div.classList; // "one two three"
```

```
div.classList.add("four");  
div.classList.remove("one");
```

```
var ul = document.createElement("ul");
```

```
var div = document.querySelector("div");  
div.appendChild(ul);
```

```
var p = document.createElement("p");
```

```
// con createTextNode
```

```
var text = document.createTextNode("Hola");  
p.appendChild(text);
```

```
// con innerHTML
```

```
p.innerHTML = "Hola";
```

```
// con textContent
```

```
p.textContent = "Hola";
```

```
const myElement = document.getElementById("demo");
myElement.style.color = "red";
```

```
<div class="example color">
  <p>This is a paragraph.</p>
</div>
<br>
<div class="example color">
  <p>This is a paragraph.</p>
</div>

<script>
const collection = document.getElementsByClassName("example color");
collection[0].style.backgroundColor = "red";
</script>
```

```
<h1>The Document Object</h1>
<h2>The getElementsByTagName() Method</h2>
```

```
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

<script>
document.getElementsByTagName("p")[0].innerHTML = "Hello World!";
</script>
```

Cats:

```
<input name="animal" type="checkbox" value="Cats">
```

Dogs:

```
<input name="animal" type="checkbox" value="Dogs">
```

```
<p>Number of elements with name="animal" is:</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
let num = document.getElementsByName("animal").length;
```

```
document.getElementById("demo").innerHTML = num;
```

```
</script>
```

The getElementsByTagName() Method

Hello World!

This is a paragraph.

This is a paragraph.

Escuchando eventos

Para escuchar eventos (del mouse, teclado, etc.) utiliza el método `addEventListener` sobre cualquier elemento.

```
var p = document.querySelector("p");  
p.addEventListener("click", function() {  
    alert("Hello");  
});
```

Eventos del mouse:

```
document.addEventListener('mousemove', function(e) {  
    console.log("El mouse se encuentra en las coordenadas:" + e.pageX + ", " + e.pageY);  
});
```

Eventos del teclado:

```
document.addEventListener('keyup', function(e) {  
    console.log("La tecla oprimida fue " + e.which);  
});
```



Estamos Trabajando en
la misión 9

Manipulación DOM - Continuación

Para esta parte: afianzar, practicar y tener claro los realizado en la sesión anterior, en especial manejo de eventos, createElement y appendChild.



Estamos Trabajando en
la misión 10

Local Storage

Es un espacio de almacenamiento en el navegador basado en llaves y valores (similar a un objeto de JavaScript con la diferencia que los valores son siempre strings).

```
localStorage.setItem("name", "Pedro Perez")
```

```
localStorage.removeItem("name")
```

```
localStorage.getItem("name")
```

```
localStorage.clear()
```



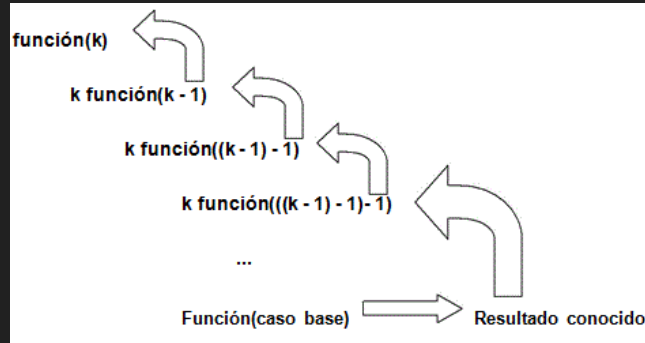
Estamos Trabajando en
la misión 11

Sesión Final

En este punto, se debe tener afianzado todo lo relacionado con Javascript del lado del navegador, manipulación DOM, el objetivo es realizar un menú dinámico con base en un arreglo de objetos, debes mapear correctamente, el estilo va al gusto y el límite está en tu imaginación.

Temas clave:

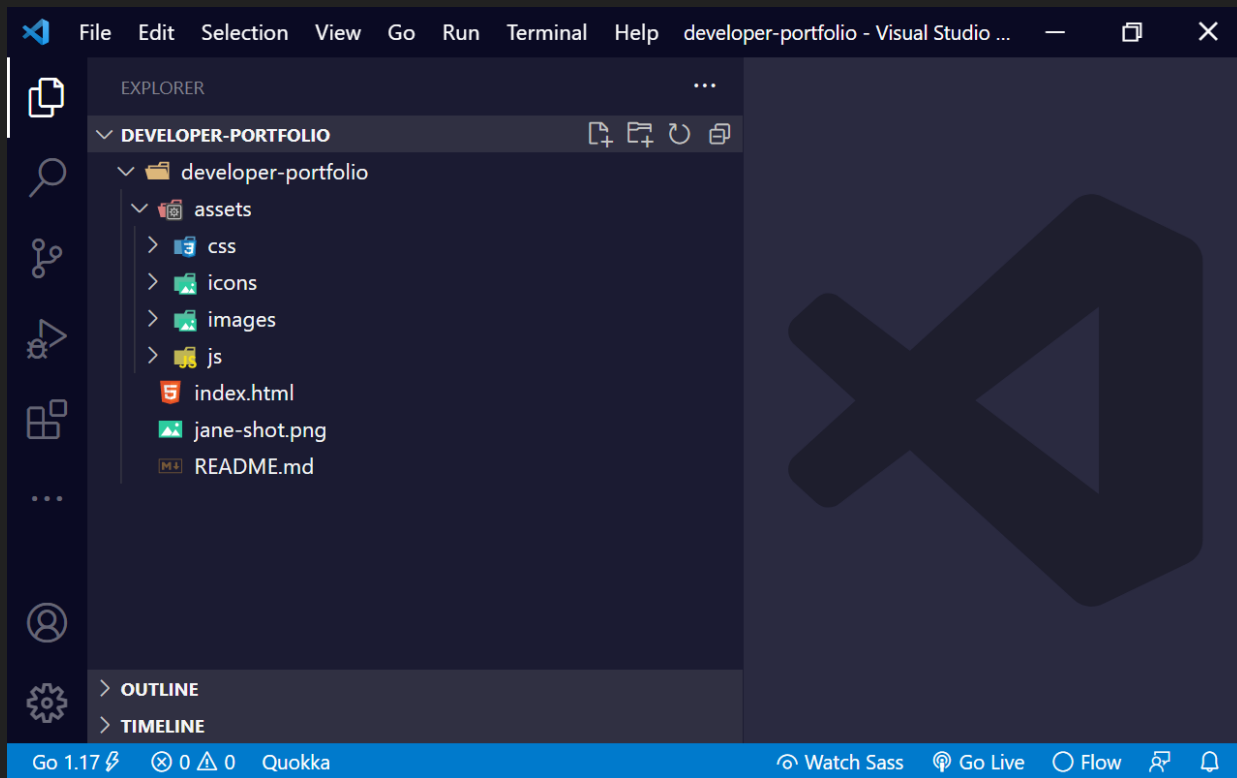
- Recursión algorítmica.



- Ciclos - Estructuras de control.



Estamos Trabajando en
la misión 12



Para todo proyecto se requiere constancia, dedicación, persistencia y capacidad de gestión, animate a ser parte de retos grandes, este es sólo el inicio.

La Nueva Educación

convertirte en
Desarrollador Web?

