


Introducción al Desarrollo de Software I - Lanzillotta

Capra - Chaves - Di Matteo - Sosa - Villegas - Palavecino

Agenda

- JavaScript
 - Integración con HTML y CSS
 - Ejemplo con Flask
-

JavaScript

A yellow square containing the letters 'JS' in a bold, dark blue, sans-serif font, representing the JavaScript logo.

JS

¿Qué es JavaScript? ¿Para qué se usa?

JavaScript es un **lenguaje de programación** que se utiliza para añadir interactividad a las páginas web. Permite crear animaciones, juegos, formularios dinámicos y mucho más. Es uno de los lenguajes de programación más populares del mundo y se utiliza en millones de sitios web.

A su vez, también es utilizado como lenguaje para el *backend*, a través de aplicaciones de servidor como por ejemplo *Node.js*

¡Antes de arrancar!

- Pueden utilizar **Visual Studio Code** para practicar con JavaScript, instalando *Node.js* y la extensión *Code Runner*. Otra opción es desde la consola de Chrome pulsando Ctrl+Shift+I
- Nos basaremos en el estándar **EcmaScript 2015 (ES6)** para explicar el lenguaje
- El punto y coma (;) al final de cada instrucción es optativo. Usarlo o no queda a su criterio, pero sean consistentes con su elección.

Variables y constantes

`const` PI = 3.14159265359 ← **constante**

`let` alumnosCursando = 237 ← **variable**

PI = 3

TypeError: Assignment to constant variable.

alumnosCursando = 122



Tipos de datos

```
const alumnos = 237           // number
const precio = 1392.50        // number
const nombre = "Bruno"        // string
const esDomingo = false       // boolean
```

```
const persona = {nombre: "pepe", edad: 26} // object
const numeros = [1, 2, 3]                  // object
let valorInicial = null                     // object
```

```
const ultimoDigitoDePI = undefined // undefined
```

podemos preguntar el tipo de dato usando el operador **typeof**

Estructuras de control

```
let numero = 8

if (numero < 20) {
  console.log("El número es menor a 20")
} else if (numero == 20) {
  console.log("El número es igual a 20")
} else {
  console.log("El número es mayor a 20")
}

while (numero < 20) {
  numero++ // numero = numero + 1
}

for (let i=1; i<5; i++) {
  const nuevoValor = numero + i
  console.log("Ahora el número vale: " + nuevoValor)
}
```

```
const color = "red"

switch(color) {
  case "red": {
    console.log("El color es rojo")
    break
  }
  case "blue": {
    console.log("El color es azul")
    break
  }
  case "yellow": {
    console.log("El color es amarillo")
    break
  }
  default: {
    console.log("El color no existe")
  }
}
```


Keywords

Las keywords o **palabras reservadas** son palabras que no se pueden usar para nombrar variables ya que son especiales para el compilador de JavaScript.

Ejemplos: `const, let, var, number, string, object, null, undefined, function, if, else, true, false, return, for, break, continue, while, switch, this, in, length, document`

Lista completa: https://www.w3schools.com/js/js_reserved.asp

Funciones

palabra
reservada
function

nombre de
la función

argumentos

Cuerpo de la función

```
function modulo(x, y) {
```

```
    const sumaCuadrados = x*x + y*y
```

```
    const resultado = Math.sqrt(sumaCuadrados)
```

```
    return resultado
```

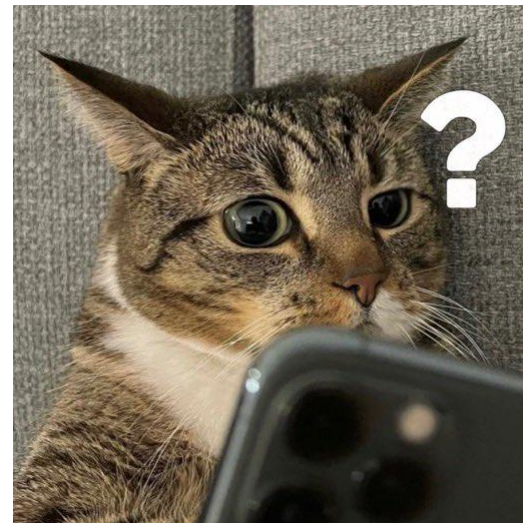
```
}
```

retorno de la función

Funciones como tipo de dato

```
const modulo = function(x, y) {  
    return Math.sqrt(x*x + y*y)  
}  
  
// Función flecha o "arrow function"  
// (args) => returnValue  
const suma = (x, y) => x + y  
  
console.log(typeof suma) // function  
console.log(typeof modulo) // function  
console.log(suma(2, 3)) // 5  
console.log(n
```

```
console.log(((x, y) => x + y)(2, 3)) // 2 + 3 = 5
```



Funciones como tipo de dato

En JavaScript, las funciones son ***first-class citizen***, es decir, se pueden manipular como valores (como si fuera un string, por ejemplo)

```
const numeros = [1, 2, 3, 4, 5, 6]      función anónima
const filtrados = numeros.filter(x => x % 2 == 0)

console.log(filtrados) // resultado: [2, 4, 6]
```

En este ejemplo, estamos pasando una función para saber si un número es par como valor (por parámetro) al método `filter`, quien internamente la va a llamar para filtrar los números de nuestra lista

Bonus track: API Requests

```
const pokemonURL = "https://pokeapi.co/api/v2/pokemon?limit=3&offset=0"
```

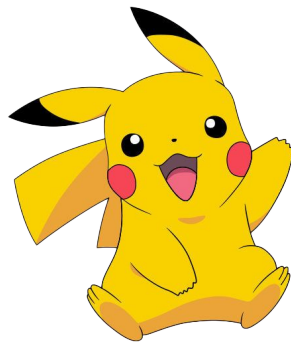
```
fetch(pokemonURL)
```

```
.then(response => response.json())
```

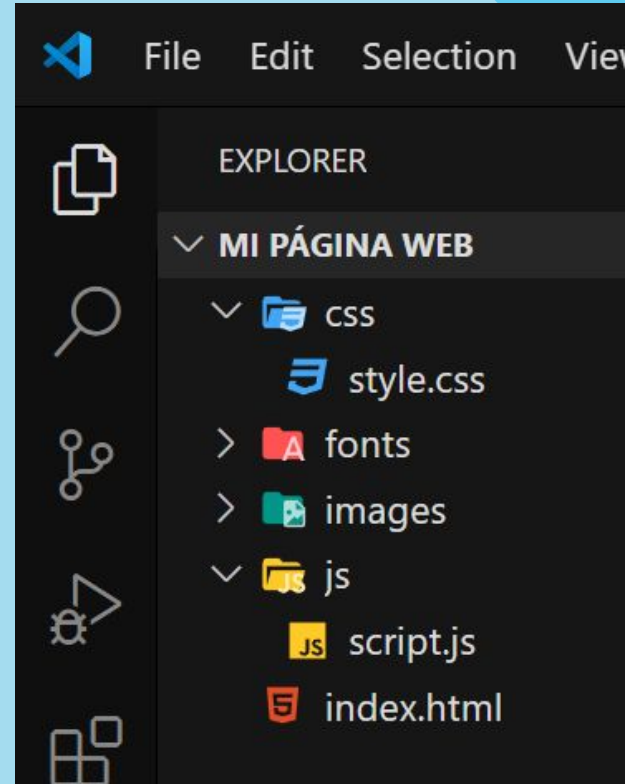
```
.then(json => console.log(json))
```

Bonus del bonus:
tipo de dato Promise

```
{
  count: 1302,
  next: 'https://pokeapi.co/api/v2/pokemon?offset=3&limit=3',
  previous: null,
  results: [
    { name: 'bulbasaur', url: 'https://pokeapi.co/api/v2/pokemon/1/' },
    { name: 'ivysaur', url: 'https://pokeapi.co/api/v2/pokemon/2/' },
    { name: 'venusaur', url: 'https://pokeapi.co/api/v2/pokemon/3/' }
  ]
}
```

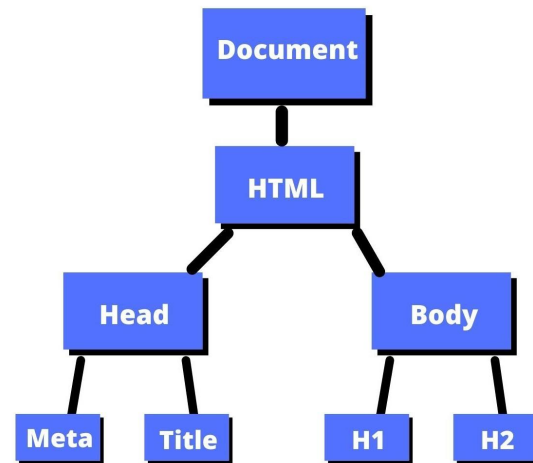


Integración con HTML y CSS



DOM: modificando nuestra página Web desde JavaScript

El **DOM** (**Document Object Model**) es una interfaz de programación para representar elementos HTML como objetos en JavaScript. Permite la manipulación de nuestra página Web a través del código.



- Cada elemento HTML es un objeto en JavaScript
- Cada atributo de ese elemento HTML es una propiedad de ese objeto en JavaScript
- Los elementos HTML responden a eventos. Esos eventos en JavaScript se representan como funciones asignadas al objeto.

¿Cómo se usa el DOM?

Ejemplo: obtener el contenido del segundo párrafo

HTML:

```
<p>Hola mundo!</p>  
<p>¡Aguante FIUBA!</p>
```

**Palabra reservada: el
objeto que representa
nuestro documento HTML**

En JavaScript:

```
const parrafos = document.getElementsByTagName("p")  
const textoParrafo2 = parrafos[1].innerText  
console.log(textoParrafo2)
```

“¡Aguante FIUBA!”

Algunas funciones y props. básicas del DOM

- `document.getElementsByTagName(...)`
- `document.getElementById(...)`
- `document.getElementsByClassName(...)`
- `document.createElement(...)`
- `document.body`

- `elemento.append(...)`
- `elemento.appendChild(...)`

- `elemento.innerText`
- `elemento.innerHTML`
- `elemento.style`
- `elemento.parentElement`

Más info acerca del DOM acá:

https://www.w3schools.com/js/js_htmlDOM.asp

Eventos: nuestra página responde a acciones del usuario

¿Qué pasa si hago click en un botón? ¿Cómo responde nuestra página ante este suceso? Esta situación es lo que se conoce como un **evento**.

Para definir la respuesta a un determinado evento de nuestra página Web necesitamos:

- Un objeto DOM que “escuche” el evento (por ejemplo, un button)
- Una función asignada a ese objeto que realice una acción cada vez que el objeto reciba una señal de ese evento.

Escuchando un evento

Ejemplo: mostrar una alerta cada vez que se haga click en un botón

HTML:

```
<button id="alerta">Pulse aquí!</button>
```

JavaScript:

```
const botonAlerta = document.getElementById("alerta")  
botonAlerta.addEventListener("click", function(event) {  
    alert("Hola mundo!")  
})
```

Escuchando un evento

(Otra forma de resolverlo)

HTML:

```
<button id="alerta">Pulse aquí!</button>
```

JavaScript:

```
const botonAlerta = document.getElementById("alerta")  
botonAlerta.onclick = function(event) {  
    alert("Hola mundo!")  
}
```

Importando scripts en nuestra página Web

Existen 2 formas de importar código JavaScript a nuestro documento HTML:

- Insertando el código dentro del elemento **<script>**
- Enlazando un archivo .js en el atributo **src** de la etiqueta **<script>**

La etiqueta **<script>** se coloca al final del *body*

Importando scripts en nuestra página Web

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Mi página Web</title>
</head>
<body>
  <button id="alerta">Pulse aquí!</button>
  <script>
    const botonAlerta = document.getElementById("alerta")
    botonAlerta.addEventListener("click", function(event) {
      alert("Hola mundo!")
    })
  </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Mi página Web</title>
</head>
<body>
  <button id="alerta">Pulse aquí!</button>
  <script src="script.js"></script>
</body>
</html>
```

Ejemplo básico

Volviendo al ejemplo de la clase 4...

```
const title = document.getElementById("titulo")

title.addEventListener("mouseover", () => {
  alert("Hola")
})

const changeColor = document.getElementById("change-color")

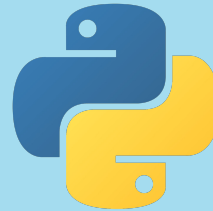
changeColor.onclick = function (event) {
  const R = randomNumber().toString()
  const G = randomNumber().toString()
  const B = randomNumber().toString()
  title.style.color = `rgb(${R},${G},${B})`
}

function randomNumber() {
  return (Math.random() * 256).toFixed(0)
}
```

**¡Vamos a explicar
en vivo cómo
funciona!**

Ejemplo con Flask

(en vivo)



Seguimos con el ejercicio de Flask de la clase anterior

Pero ahora vamos a darle **funcionalidad** a nuestra página...

Ejercitación

Ejercicio

Crear una página Web que contenga 2 campos de texto y un botón. El botón debe ser rojo con letra blanca. Al pulsar dicho botón se deberá mostrar una alerta por pantalla que muestre la concatenación de las cadenas que se ingresen en los campos de texto.

- Pista 1: existe un elemento HTML de tipo “campo de texto”.
- Pista 2: existe un atributo que me devuelve el contenido ingresado en un campo de texto

(No hace falta usar Flask)

