

Javascript + Node.js y Express

¿Qué es Javascript?

Es un lenguaje de programación que permite añadir comportamiento dinámico e interactivo a las páginas web. Es el único lenguaje de programación que entienden de forma nativa los navegadores. Es orientado a objetos, débilmente tipado y dinámico (el tipo está asociado al valor, no a la variable. X puede ser un numero en un principio y luego una cadena).

Ejemplos de uso:

- Validar formularios antes de enviarlos.
- Mostrar u ocultar elementos de la página según ciertos eventos como click.
- Cambiar el estilo de los componentes de forma dinámica.

¿Dónde escribimos el código Javascript?

Hay tres formas comunes:

1. Inline (en una etiqueta HTML):

```
<button onclick="alert('hiciste click!')">Click</button>
```

2. Dentro de una etiqueta <script> en el html:

```
<script>
  console.log('Hola desde JavaScript');
</script>
```

3. En un archivo externo .js:

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

y en el archivo script.js

```
function saludar(nombre){
  console.log("Hola " + nombre);
}
```

Tipos de datos y variables

Para declarar variables podemos usar let, var y const.

let → variable que puede cambiar y vive dentro de un "bloque".

const → constante

var → variable que puede cambiar y vive dentro de una función.

```
let nombre = "Juan";      // string
let edad = 25;            // number
let activo = true;        // boolean
let colores = ["rojo", "azul"]; // array
let persona = { nombre: "Ana", edad: 30 }; // objeto
```

Condicionales

```
if (edad >= 18) {
  console.log("Mayor de edad");
} else {
  console.log("Menor de edad");
}
```

Bucles

```
for (let i = 0; i < 5; i++) {
  console.log("Número:", i);
}

let i = 0;
while (i < 5) {
  console.log("Iteración:", i);
  i++;
}
```

Funciones

```
function saludar(nombre) {
  console.log("Hola " + nombre);
}

const despedir = (nombre) => {
  console.log("Chau " + nombre);
};
```

DOM (document object model)

Es una representación orientada a objetos de la página web. Se plantea la página como una estructura de árbol en la cual cada elemento es un objeto o nodo con determinadas propiedades y métodos asociados. Javascript utiliza esta representación de manera que podamos cambiar la estructura del documento, sus estilos y contenido.

Una propiedad es un valor que se puede obtener o definir, como por ejemplo obtener y redefinir el contenido de un elemento, cambiar el color de fondo, etc.

Un método es una acción, una función que se puede utilizar para obtener, eliminar o agregar un elemento html.

Ejemplos de propiedades:

- `element.innerHTML`
- `element.classList`
- `element.children`
- `element.id`
- `element.style`
- `element.value`

Ejemplos de métodos:

- `document.getElementById("")`
- `document.getElementsByClassName("")`
- `document.querySelectorAll("")`
- `document.getElementsByTagName("")`
- `element.insertCell("")`
- `element.appendChild()`

Para acceder a elementos

```
const titulo = document.getElementById("titulo");
const input = document.querySelector("input");
```

Para modificar su contenido

```
titulo.innerText = "Nuevo título";
input.value = "Texto por defecto";
```

Para manejar eventos

```
document.getElementById("boton").addEventListener("click", () => {
  alert("¡Botón clickeado!");
});
```

¿Qué es Node.js?

Es una plataforma o entorno de tiempo de ejecución diseñado para la ejecución de JavaScript en el lado del servidor, fuera de un navegador.

Instalación

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash
source ~/.bashrc
nvm install --lts
```

Primera app con Node.js

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Estructura de un proyecto node.js

Node modules: carpeta que tiene todos los archivos de las dependencias que se usan en el proyecto.

Public: archivos estáticos como html.

Package.json: toda la información de las dependencias que están instaladas en el proyecto.

App.js: Punto de entrada del servidor, contiene las rutas principales de la aplicación.

¿Qué es express?

Es un framework para Node.js que facilita la creación de servidores web y APIs de manera sencilla. Nos permite crear diferentes rutas.

Ejemplo

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.get('/contacto', (req, res) => {
  res.sendFile(__dirname + "/formulario.html")
})
```

```
})  
app.get('/insertar', (req, res) => {  
  res.sendFile(__dirname+"/formulario.html")  
})  
  
app.listen(port, () => {  
  console.log(`Example app listening on port ${port}`)  
})
```

Para manejar otros directorios estáticos:

```
app.use(express.static('public'))  
app.use(express.static('files'))
```

Referencias:

<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>

<https://www.npmjs.com/package/mysql#introduction>

<https://expressjs.com/es/starter/hello-world.html>

<https://nodejs.org/en/docs/guides/getting-started-guide>