OTEC
SOCATECH

SOCIEDAD DE CAPACITACIÓN
TECNOLÓGICA DE CHILE

BOOTCAMP

JAVASCRIPT FULLSTACK AVANZADO

Docente: Javier Navarrete Herrera

¿QUÉ UTILIZAREMOS?

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

Durante el bootcamp, utilizaremos un editor de código para escribir y ejecutar nuestros programas. En este caso, usaremos Visual Studio Code (VSCode).

📎 Descargar VSCode:
https://code.visualstudio.com/download

Además, trabajaremos con Node.js, por lo que es importante instalar la versión LTS (Long Term Support) para garantizar estabilidad y compatibilidad.

📎 Descargar Node.js (versión LTS):
https://nodejs.org/es/download

¿QUÉ ES JAVASCRIPT?

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

JavaScript es un lenguaje de programación interpretado que se ejecuta en navegadores web y en entornos de servidor como Node.js. Es utilizado para agregar interactividad a las páginas web y construir aplicaciones completas en el backend.

¿QUÉ ES NODE.JS?

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

Node.js es un entorno de ejecución de JavaScript basado en el motor V8 de Chrome. Permite ejecutar código JavaScript en el servidor, lo que amplía las capacidades del lenguaje más allá del navegador.

JS VS NODE.JS

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

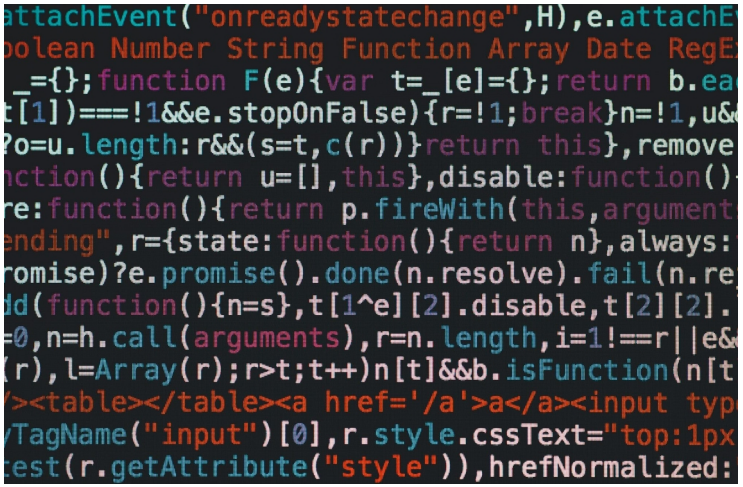
CARACTERÍSTICAS	JS EN EL NAVEGADOR	JS EN NODE.JS
USO PRINCIPAL	Manipulación del DOM	Backend y CLI
APIS DISPONIBLES	document, window, fetch, localStorage y sessionStorage	fs, http, path, os, crypto, process, child_process
EJECUCIÓN	En el motor del navegador	En la terminal

PONIENDO EN PRÁCTICA:

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

- Manipulación del DOM
- Uso de window para obtener información de usuario
- Consumo de API con fetch
- Módulo fs (File System)
- Módulo http
- Módulo path



DATOS PRIMITIVOS

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

En JavaScript, los tipos de datos primitivos son aquellos que no son objetos y que representan un único valor. Aquí tienes una lista de los tipos de datos primitivos en JavaScript:

- **Number:** Representa números, tanto enteros como decimales. Ejemplo: 42, 3.14,-7
- **BigInt:** Manejo de grandes números enteros. Pueden crecer tanto como lo permita la memoria disponible de la máquina.
- **String:** Representa secuencias de caracteres o textos. Ejemplo: 'JavaScript', "JavaScript", `Esto es un template string, y si, puedo poner comas en medio.`
- **Boolean:** Representa un valor lógico, puede ser true o false.
- **Undefined:** Variable sin valor asignado.

ESTRUCTURAS DE CONTROL

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

Las estructuras de control en JavaScript permiten controlar el flujo de ejecución de un programa:

- **if - Sí** (el bloque de código se ejecuando cuando la condición if es verdadera).
- **else - Si no** (se ejecuta cuando la condición if es false).
- **else if - Si no, pero si** (permite evaluar múltiples condicionales).
- **switch - Cambio de casos** (Permite ejecutar uno de varios bloques de código).
- **for - Bucle básico** (Se utiliza cuando sabes de antemano cuántas veces necesitas repetir el bloque de código).
- **while - Mientras** (Repite el bloque de código mientras se cumpla una condición).
- **do...while - Hacer, mientras** (Ejecuta el código de bloque al menos una vez, y luego se evalúa su condición).

OPERADORES TERNARIOS Y STRINGS LITERALS

Introducción a JavaScript

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

- Operador Ternario en JavaScript
  - Es una forma abreviada de escribir una estructura condicional if-else.

    Ejemplo: condición ? expresión 1 : expresión 2;
- Strings Literals (Template Literals)
  - Los strings literales o template literals son una forma de crear cadenas de texto en JavaScript de una manera más flexible y potente que los string tradicionales.

    Ejemplo: `cadena de texto`
  - Además nos permite **interpolación** de variables:

```
let nombre = “Juan”;
let edad = 25;
let mensaje = `Hola, mi nombres es ${nombre} y tengo ${edad} años.`;
console.log(mensaje); // “Hola, mi nombre es Juan y tengo 25 años.”
```

OTEC SOCATECH
SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE

📎 **Documentación oficial de JavaScript (MDN Web Docs)**
https://developer.mozilla.org/es/docs/Web/JavaScript
📎 **APIs del Navegador (DOM, Window, Fetch, etc.)**
`document`: https://developer.mozilla.org/es/docs/Web/API/Document
`window`: https://developer.mozilla.org/es/docs/Web/API/Window
`fetch`: https://developer.mozilla.org/es/docs/Web/API/Fetch_API
📎 **Documentación de Node.js**
https://nodejs.org/docs/latest/api/
📎 **Módulos principales de Node.js**
`fs` (Sistema de archivos): https://nodejs.org/docs/latest/api/fs.html
`http` (Servidor HTTP): https://nodejs.org/docs/latest/api/http.html
`path` (Manejo de rutas): https://nodejs.org/docs/latest/api/path.html