

Precurso HTML, CSS y Javascript

1. Programas del curso:

1.1. Visual Code Studio Este programa nos permite visualizar, escribir y ejecutar código en cualquier lenguaje de programación.

Node.js: Este es el entorno que nos permite ejecutar nuestro código basado en Java Script

2. JavaScript

Es un lenguaje de programación que nos permitirá crear programas que ejecuten acciones complejas en paginas web. Estos programas tienen la extensión .js en el título.

3. Conceptos de JS

3.1. Variables

- **var:** Crear variable.
- **const:** Crear variable que no se cambiará.
- **let:** Crear variable que cambiará.

3.2. Tipos de Datos

- **string:** Datos en formato de Texto.
- **numbers:** Datos numéricos.
- **booleans:** Datos de verdadero o falso (Sí o No).

3.3. Métodos para Strings

Lista de métodos: https://www.w3schools.com/jsref/jsref_obj_string.asp

- **endsWith():** Genera boolean si el texto termina con lo que preguntemos.
- **toUpperCase():** Convierte todos los caracteres a MAYUSCULAS.
- **indexOf():** Busca en que posición se encuentra por primera vez los caracteres que preguntemos.
- **slice():** El resultado son todos los caracteres desde la primera posición hasta antes de la última que indiquemos.
- **includes():** Verifica si los caracteres que escribamos se encuentran o no en el string mediante un boolean.

3.4. Array: Un array es una estructura de datos que almacena más de un valor.

Estructura: **const x = []**

3.5. Métodos para Arrays

Lista de métodos: https://www.w3schools.com/jsref/jsref_obj_array.asp

- **pop():** Elimina el último elemento de un arreglo.
- **push():** Agregar un elemento al final de un arreglo.
- **splice():** Agrega y eliminar elementos de cualquier index del array.
- **forEach():** Ejecuta una función para cada elemento de un array.

3.6. Objetos

Los objetos tienen una estructura que nos permite manipular datos de forma muy similar a como lo hacemos en el mundo real, donde tenemos objetos y estos tienen ciertas características

Estructura: **const x = { }**

3.7. Operadores

- **>** : Mayor que.
- **<** : Menor que.
- **===** : Igual que.
- **!==** : Distinto que.
- **>=** : Mayor o igual que.
- **<=** : Menor o igual que.
- **&&**: Y.
- **||**: O.

3.8. Condicionales

Sirve para ejecutar código solo si se cumple una condición. Estructura: **if() { }**

Ejemplo:

```
const a = 45
if (a >= 18 && a < 30) {
  console.log('Eres el grupo A')
} else if (a >= 30 && a < 40) {
  console.log('Eres el grupo B')
} else if (a >= 40 && a < 50) {
  console.log('Eres el grupo C')
}
```

3.9. Funciones

Sirven para reutilizar código y optimizar el código. Estructura: **function x() { }**

Ejemplo:

```
function storeAcademloUser(user) {
  const isAcademloUser = user.email.endsWith('gmail.com')
  if(isAcademloUser) {
    academloUsers.push(user.email);
  }
}
```

3.10. Ejercicio extraer Gmails

```

JS precurso.js > ...
1  const users =
2  [{
3      name: 'Aria',
4      email: 'aria@academlo.com',
5      gender: 'Female',
6      social: [
7          { name: 'facebook', url: 'facebook/ariawa' },
8          { name: 'instagram', url: 'instagram/aria' }
9      ]
10 },
11 > { ...
19 },
20 > { ...
28 },
29 > { ...
37 },
38 > { ...
46 }]
47 const academloUsers = []
48 function storeAcademloUser(user) {
49     const isAcademloUser = user.email.endsWith('gmail.com')
50     if(isAcademloUser) {
51         academloUsers.push(user.email);
52     }
53 }
54 users.forEach(storeAcademloUser)
55 console.log('Los correos Gmail de los usuarios son: ' + academloUsers)

```

TERMINAL PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN

```

PS C:\Users\alless\OneDrive\Documentos\precurso-node-js> node .\precurso.js
Los correos Gmail de los usuarios son: grimix@gmail.com,modogh@gmail.com,khey4@gmail.com
PS C:\Users\alless\OneDrive\Documentos\precurso-node-js>

```