

Introducción a JavaScript y Git

JavaScript es un lenguaje de programación fundamental para el desarrollo web. Se utiliza para agregar interactividad a las páginas web, desde simples validaciones de formularios hasta aplicaciones completas en el navegador. En este tutorial, cubriremos estructuras de control y repetición, así como una introducción a Git, una herramienta esencial para el control de versiones en proyectos de desarrollo.

1. Condicionales en JavaScript

Las estructuras condicionales permiten ejecutar diferentes bloques de código dependiendo de una condición. Son fundamentales para la toma de decisiones dentro de un programa.

1.1 If, Else y Else If:

La estructura ``if`` evalúa una condición y, si es verdadera, ejecuta el bloque de código correspondiente. Si la condición es falsa, se puede ejecutar un bloque alternativo usando ``else``.

```
```javascript
let edad = 18;
if (edad >= 18) {
 console.log("Eres mayor de edad");
} else {
 console.log("Eres menor de edad");
}
```
```

Si hay varias condiciones, se puede usar ``else if``:

```
```javascript
let nota = 85;
if (nota >= 90) {
 console.log("Sobresaliente");
}
```

```
} else if (nota >= 70) {
 console.log("Aprobado");
} else {
 console.log("Reprobado");
}
...
```

### ### 1.2 Switch:

El `switch` es una alternativa a `if-else` cuando se comparan múltiples valores de una variable.

```
```javascript  
let dia = "lunes";  
switch (dia) {  
    case "lunes":  
        console.log("Inicio de semana");  
        break;  
    case "viernes":  
        console.log("Fin de semana");  
        break;  
    default:  
        console.log("Día normal");  
}  
...
```

1.3 Operador Ternario:

El operador ternario es una forma compacta de escribir un `if-else`.

```
```javascript  
let permiso = edad >= 18 ? "Acceso permitido" : "Acceso denegado";
console.log(permiso);
...
```

---

## ## 2. Bucles en JavaScript

Los bucles permiten repetir código varias veces, lo que ayuda a automatizar tareas repetitivas.

### ### 2.1 For:

El `for` se usa cuando sabemos cuántas veces queremos ejecutar un bloque de código.

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

### ### 2.2 While:

El `while` ejecuta un bloque de código mientras la condición sea verdadera.

```
```javascript
let contador = 0;
while (contador < 5) {
  console.log("Contador: " + contador);
  contador++;
}
```
```

### ### 2.3 Do While:

El `do while` siempre ejecuta el bloque al menos una vez antes de evaluar la condición.

```
```javascript
let num = 0;
do {
  console.log("Valor: " + num);
  num++;
} while (num < 5);
```
```

---

## ## 3. Introducción a Git

Git es un sistema de control de versiones que permite rastrear cambios en el código y trabajar de manera colaborativa.

Es ampliamente utilizado en desarrollo de software para gestionar proyectos.

### ### 3.1 Instalación:

Descarga e instala Git desde: [<https://git-scm.com/>](<https://git-scm.com/>)

### ### 3.2 Comandos Básicos:

#### **\*\*Configurar Git:\*\***

Estos comandos establecen el nombre de usuario y el correo electrónico que se registrará en los commits.

```
```sh
git config --global user.name "Tu Nombre"
git config --global user.email "tuemail@example.com"
```
```

#### **\*\*Inicializar un repositorio:\*\***

Este comando crea un nuevo repositorio en la carpeta actual.

```
```sh
git init
```
```

#### **\*\*Agregar cambios al área de preparación:\*\***

Antes de hacer un commit, los archivos deben ser agregados al área de preparación.

```
```sh
git add .
```
```

#### **\*\*Hacer un commit:\*\***

Guarda los cambios en el historial del repositorio.

```
```sh
git commit -m "Mensaje del commit"
```
```

**\*\*Ver el estado del repositorio:\*\***

Muestra los cambios realizados y pendientes de confirmación.

```
```sh
git status
```
```

**\*\*Ver historial de commits:\*\***

Lista todos los commits realizados en el repositorio.

```
```sh
git log
```
```

**\*\*Conectar con un repositorio remoto:\*\***

Vincula el repositorio local con uno en GitHub.

```
```sh
git remote add origin https://github.com/usuario/repositorio.git
git push -u origin main
```
```

---

Con estos conceptos básicos de JavaScript y Git, ya puedes comenzar a desarrollar y gestionar tu código de manera eficiente.

La práctica es clave para dominar estas herramientas, así que sigue explorando y aplicando lo aprendido. ¡Buena suerte!