



## M4.250 - Introducción a la programación en JavaScript I aula 1

### Control de flujo: Condicionales

Inicio:

**21/10/19**

Fin:

**09/11/19**

Dedicación:

**20 h**

### Introducción

En este capítulo se tratan dos temas. Por un lado las estructuras de control condicionales. Por el otro el control de errores.

Empieza con una explicación sobre los bloques que incluye algún comentario acerca del funcionamiento el ámbito de las variables en versiones anteriores de JavaScript, que aunque interesante, no es relevante para nuestro curso.

Este tema se encuentra explicado en el material de Mozilla en el apartado [Control de flujo y manejo de errores](#)

### Sentencias condicionales

Son dos los tipos de sentencias condicionales con los que cuenta JavaScript.

Por una parte está el `if ... else` y por la otra el `switch`.

#### **if ... else**

La estructura `if ... else` puede usarse de diversas maneras:

Una primera sería esta:

```
if (condicion) {  
    sentencias;  
}
```

Es decir, sin el `else`. En este caso, si la condición es cierta, se ejecutan las sentencias que hay dentro del bloque. En caso contrario se obvian.

Una segunda manera sería añadiendo el `else` :

```
if (condicion) {  
    sentencias;  
} else {  
    sentencias;  
}
```

En este caso, si la condición es cierta se ejecutarán las instrucciones que están en el bloque del `if` y si es falsa, se ejecutarán las instrucciones que hay en el bloque del `else` .

Podemos además encadenar `if` :

```
if (condicion1) {  
    sentencias1;  
} else if (condición2) {  
    sentencias2;  
} else if (condición3) {  
    sentencias3;  
} else {  
    sentencias4;  
}
```

Si la primera condición es cierta, se hace el primer bloque de sentencias. Si es falsa, se evalúa la siguiente condición. Si es cierta, se ejecuta el segundo bloque de sentencias, ...

Nota: Nunca se debe usar una asignación como parte de una condición. Por tanto, es importante acordarse de que la comparación se representa con `==` o `===` (preferiblemente `===`).

## switch

En la instrucción `switch` se compara la expresión con los valores que hay en los `case`. Si son iguales, se ejecutan las instrucciones que hay bajo el `case` que se cumple.

Un ejemplo sencillo:

```
let dia = 3;  
switch (dia) {  
    case 1:  
        console.log("lunes");  
        break;  
    case 2:  
        console.log("martes");  
        break;  
    case 3:  
        console.log("miércoles");  
        break;  
    case 4:  
        console.log("jueves");  
        break;  
    case 5:  
        console.log("viernes");  
        break;  
    default:  
        console.log(";Fin de semana!");  
}
```

Pruébalo aquí: <https://goo.gl/JfZbo3>

El `break` que finaliza cada caso, impide que se sigan ejecutando las instrucciones que hay debajo. Prueba a quitarlos en el ejemplo de la web .

---

## Manejo de errores

Las excepciones se utilizan para tratar situaciones tanto en las que se produce un error como en aquellas en que se produce una situación inesperada (cómo por ejemplo encontrar un número negativo en un marcador de un partido de fútbol).

El funcionamiento es el siguiente, el conjunto de instrucciones que quiere analizarse por si tiene un error se ponen dentro del bloque de la instrucción `try` . Las instrucciones que se ejecutarán en caso de que se produzca un error, se escriben en el bloque de la instrucción `catch` . Existe una instrucción adicional, `finally` , que indica que esas instrucciones se ejecutarán siempre, se produzca o no un error.

Veamos un ejemplo:

```
let golesVillaverde = 0;
let golesBadalona = 11;
try {
  if (golesVillaverde < 0 || golesBadalona < 0) {
    throw "¿Goles negativos?";
  }
  if (golesVillaverde > 10 || golesBadalona > 10) {
    throw "No me lo creo";
  }
  if (golesVillaverde < golesBadalona){
    console.log("Gana el Badalona");
  } else if (golesBadalona > golesVillaverde) {
    console.log("Gana el Villaverde");
  } else {
    console.log("Empate");
  }
  console.log("Si se escribe esto no hay error");
} catch (e) {
  switch (e) {
    case "¿Goles negativos?":
      console.log("Error: Goles negativos");
      break;
    case "No me lo creo":
      console.log("Más de 10 goles... Difícil");
      break;
  }
}
```

Se puede probar este ejemplo en: <https://goo.gl/2zR9VM>

Algunos detalles:

- Cualquier instrucción que haya dentro del `try` , después de ejecutar la instrucción `throw` , no se ejecutará.

- Podemos actuar de manera diferente según el error gracias al parámetro que acompaña a la instrucción catch .
- Si añadimos una instrucción finally , lo que haya en su bloque se ejecutará siempre, pero después de ejecutar lo que haya en el bloque catch .

---

## **Recursos d'aprenentatge**

---

### **Fuentes de información**

---