



2.A. Estructura del lenguaje JavaScript

6.- Estructuras de control

6.1.- Sentencias condicionales



En los lenguajes de programación, las sentencias condicionales permiten decidir acciones en función del resultado de evaluar condiciones o expresiones lo que nos permite dirigir el flujo de ejecución de nuestro programa.

Una parte muy importante de una sentencia condicional es la "condición" o "expresión" a evaluar y que puede consistir en cualquier expresión válida de la que se obtenga un valor.

Es importante recordar que en JavaScript que algunos valores que no son booleanos tiene implícito el valor `false` cuando se evalúan en una condición, por ejemplo, el 0 y la cadena vacía ("") tienen un valor de verdad `false`

En JavaScript tenemos varias sentencias condicionales para las diferentes situaciones que te puedas encontrar durante la programación.

Construcción `if`

La decisión más simple que podemos tomar en un programa, es la de seguir una rama determinada si una determinada condición es `true` .

Sintaxis:

```
if (typeof miVariable == "number") {  
    miVariable = parseInt(miVariable);  
}
```

Ejemplo:

```
if (miEdad >30) {  
    alert("Ya eres una persona adulta");  
}
```

Construcción `if ... else`

En este tipo de construcción, podemos gestionar que haremos cuando se cumpla y cuando no se cumpla una determinada condición.

Sintaxis:

```
if (condición) { // entre paréntesis irá la condición que se evaluará a true o false.  
    // instrucciones a ejecutar si se cumple la condición  
} else {  
    // instrucciones a ejecutar si no se cumple la condición  
}
```

Ejemplo:

```
if (miEdad >30) {  
    alert("Ya eres una persona adulta.");  
} else {  
    alert("Eres una persona joven.");  
}
```

```
}
```

Construcción **switch**

Equivale a varias construcciones **if...else** anidadas, sin embargo es bastante más legible de cara a los desarrolladores. No se debe olvidar que en esta sentencia el flujo no se rompe en ningún momento por lo que es importante escribir las correspondientes sentencias **break** para controlar el flujo.

Sintaxis:

```
switch (expresión) { // entre paréntesis escribimos la expresión a evaluar
    case valor1: // instrucciones a ejecutar si la evaluación de la expresión coincide con
valor1
                    break; // importante para romper el flujo
    case valor2: // instrucciones a ejecutar si la evaluación de la expresión coincide con
valor2
                    break; // importante para romper el flujo
    .....
    case valorN: // instrucciones a ejecutar si la evaluación de la expresión coincide con
valorN
                    break: // importante para romper el flujo
    default: // instrucciones a ejecutar si la evaluación de la expresión no coincide con
ningun valor.
}
```

Ejemplo:

```
switch (tipoIVA) {
    case 4: alert("Superreducido");
        break;
    case 10: alert("Reducido");
        break;
    case 21: alert("General");
        break;
    default: alert("Valor de IVA incorrecto");
}
```

Créditos de la imagen

Autoría: nordicdesign.

Licencia: CC BY-SA 2.0.