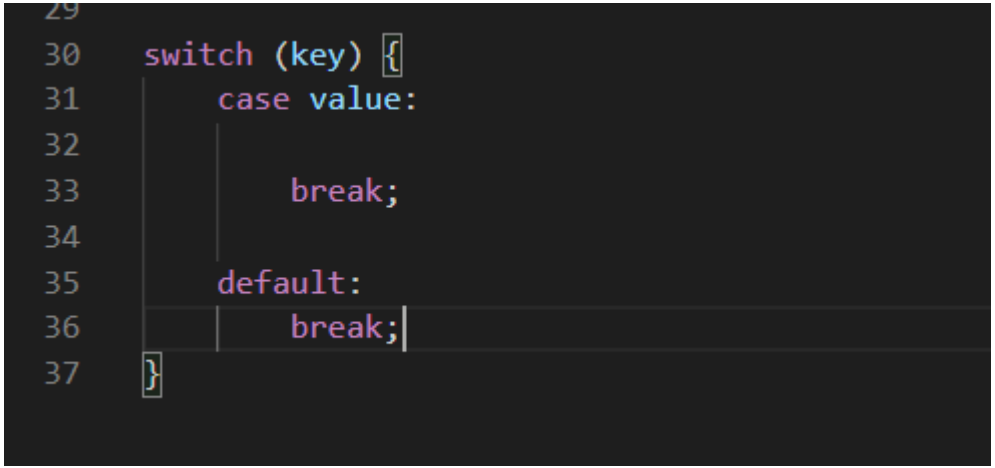
Fundamentals Sesion 5

Sentencia **switch**

Sentencia switch

Se utiliza para evaluar condiciones, es una forma abreviada de la sentencia if anidada.

Se compone de:
key: valor a evaluar.
case: condición que se debe cumplir.
break: termina la condición
default: valor por defecto si no existe caso para evaluar.



Reto 1

```
1   // Utilizando la estructura switch, evalue la siguiente condicion
2   // Si la edad es 15, escriba en consola, 'Adolescente'
3   // Si la edad es 18, escriba en consola, 'Mayor de edad'
4   // Si la edad es 60, escriba en consola, 'Adulto mayor'
5   // Si no es ninguna de las anteriores, escriba 'Edad fuera de rango'
6
```

Reto 2

```
1   // Utilizando la estructura switch, evalue la siguiente condicion
2   // Solicite al usuario un numero entre 1 y 7
3   // De acuerdo al numero dado, escriba en consola el dia de la semana
4   // a la cual pertenece.
5   // Ejemplo: si el numero ingresado es 1, escriba en consola 'Lunes'
6   // Si el numero ingresado es 3, escriba en consola 'Miercoles'
7
```

Reto 3

```
36  let navegador = prompt('Que navegador usas');
37
38  switch (navegador) {
39      case 'Edge':
40          console.log('Navegador de Microsoft');
41          break;
42      case 'Chrome':
43          console.log('Navegador de Google');
44          break;
45      case 'Firefox':
46          console.log('Navegador de Mozilla');
47          break;
48      case 'Safari':
49          console.log('Navegador de Apple');
50          break;
51
52      default:
53          console.log('Tal vez su navegador no esta soportado.');
54          break;
55  }
```

*Utilizando el siguiente fragmento de codigo, convierta la estructura en un **if-else**.*

Fundamentals Sesion 5

Ciclos (loops) **for, while, do-while**

Ciclo **while**

Los ciclos se utilizan cuando necesitamos repetir acciones.

```
while (condición) {
    // instrucción 1
    // instrucción 2
    // ……..
    // instrucción n
}
```

```
do {
    // instrucción 1
    // instrucción 2
    // ……..
    // instrucción n
} while (condición);
```

Reto 4

app.js

```
1   // Constuir un programa que imprima en consola los numeros del 1 al 50, en incrementos de 5
```

Reto 5

app.js

```
1   // Constuir un programa que imprima en consola los primeros 10 numeros pares.
```

Ciclo **for**

Se define de la siguiente forma:

```
for (inicial; condición; incremento) {
    // instrucción 1
    // instrucción 2
    // ……..
    // instrucción n
}
```

Ciclo **for**

Variantes del ciclo for:

```
for (let i = 0; i < 10; i++) {
    console.log(i);
}
```

```
let i = 0;
for ( ; i < 10; ) {
    console.log(i);
    i++;
}
```

```
let i = 0;
for (; i < 10; i++) {
    console.log(i);
}
```

Reto 6

```
// Utilizando la forma pre-incremento, cual seria el resultado
// de ejecutar la siguiente instruccion.
for (let i = 0; i < 5; ++i) alert( i );
```

```
// Utilizando la forma post-incremento, cual seria el resultado
// de ejecutar la siguiente instruccion.
for (let i = 0; i < 5; ++i) alert( i );
```

Reto 7

```
// Remplazar la siguiente estructura for, por un while
for (let i = 0; i < 10; i++) {
    console.log(`Numero: ${i}`);
}
```

Sentencias **break y continue**

Se utilizan para interrumpir la ejecución de un ciclo, la sentencia break, termina el ciclo en el que se encuentra, mientras que la sentencia continue, ignora el bloque de instrucciones y salta a la siguiente iteración:

```
for (let i = 0; i < 10; i++) {
    // break
    console.log(i);
}
```

// break interrumpe la ejecucion.

```
for (let i = 0; i < 10; i++) {
    // continue
    console.log(i);
}
```

// continue, salta a la siguiente iteración

Desafíos

```
// Leer un numero y mostrar su cuadrado en consola, repetir el proceso
// hasta que se introduzca un numero negativo.

// Leer un numero e indicar en consola si es positivo o negativo
// el ciclo termina cuando se ingresa un cero.

// Leer un numero hasta que se ingrese null, o espacio en blanco
// para cada numero mostrar si es par o impar.

// Leer numeros hasta que se introduzca un numero negativo
// y mostrar cuantos numeros fueron leidos.
```

Desafíos

```
// Pedir numeros hasta que se teclee un cero, espacio en blanco
// o null, mostrar la suma de todos los numeros.

// Pedir numeros hasta que se ingrese un numero negativo o null
// calcular el promedio de los numeros ingresados.

// Pedir un numero N y mostrar los numeros del 1 al N
// si el numero es negativo, cero o espacio en blanco, solicitar de
// nuevo el numero indicando un mensaje.
```

End fundamentals