

# Los condicionales

IF TERNARIO / SWITCH

DigitalHouse >  
Coding School



Certified Tech  
Developer

The Ultimate Degree

# Índice

1. [If ternario](#)
2. [Switch](#)

# 1 | If ternario

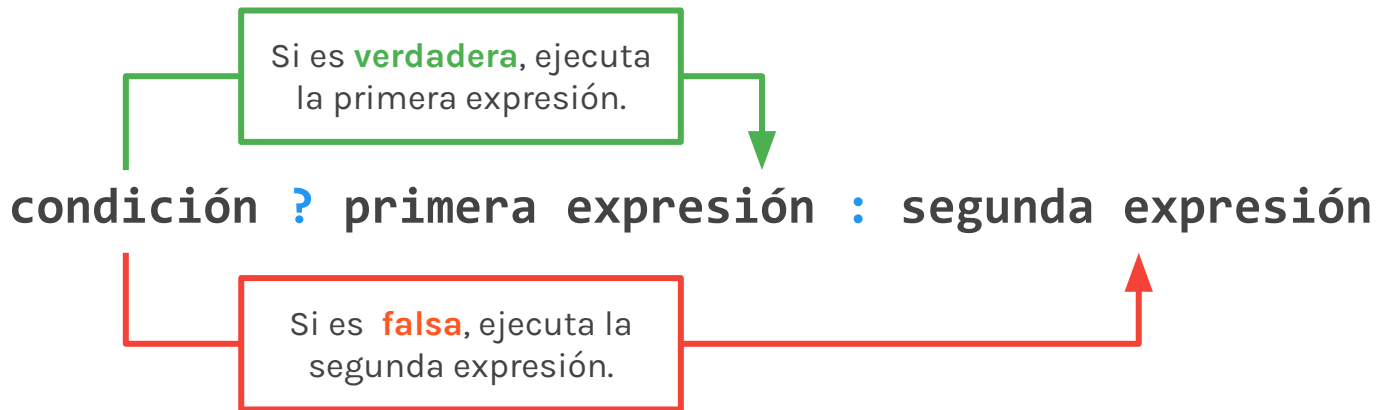


Como ya dijimos antes: si algo **se usa mucho** en programación, los lenguajes suelen darnos una **versión abreviada**.



# Estructura básica

A diferencia de un if tradicional, el **if ternario** se escribe de forma **horizontal**. Al igual que el if tradicional, tiene el mismo flujo (si esta condición es verdadera hacer esto, si no, hacer esto otro), pero en este caso **no hace falta** escribir la palabra **if** ni la palabra **else**.



# Estructura básica

Para el if ternario **es obligatorio** poner código en la **segunda expresión**. Si no queremos que pase nada, podemos usar un string vacío ''.

```
{ } 4 > 10 ? 'El 4 es más grande' : 'El 10 es más grande';
```

## Condición

Declaramos una expresión que se evalúa como true o false.

## Primera expresión

Si la condición es verdadera, se ejecuta el código que está después del signo de interrogación.

## Segunda expresión

Si la condición es falsa, se ejecuta el código que está después de los dos puntos. Es obligatorio escribirla.

# 2 | Switch



El switch nos propone una sintaxis más legible para los casos en los que queremos evaluar muchas posibilidades de un solo valor.





# Estructura básica

El switch está compuesto por una expresión a evaluar, seguida de diferentes casos —tantos como queramos— cada uno contemplando un escenario diferente. Los casos deberán terminar con la palabra reservada **break** para evitar que se ejecute el próximo bloque.

```
{  
  switch (expresión) {  
    case valorA:  
      // código a ejecutar si la expresión es igual a valorA  
      break;  
    case valorB:  
      // código a ejecutar si la expresión es igual a valorB  
      break;  
  }  
}
```

# Agrupamiento de casos

El switch también **nos permite agrupar casos** y ejecutar un mismo bloque de código para cualquier caso de ese grupo.

```
{  
    switch (expresión) {  
        case valorA:  
        case valorB:  
            // código a ejecutar si la expresión es igual a ValorA o B  
            break;  
        case valorC:  
            //código a ejecutar si valorC es verdadero  
            break;  
    }  
}
```

# {código}

```
let edad = 5;
```

Definimos la variable **edad** y le asignamos el número 5.

```
switch (edad) {  
  case 10:  
    console.log('Tiene 10 años');  
    break;  
  case 5:  
    console.log('Tiene 5 años');  
    break;  
}
```

# {código}

```
let edad = 5;
```

```
switch (edad) {  
  case 10:  
    console.log('Tiene 10 años');  
    break;  
  case 5:  
    console.log('Tiene 5 años');  
    break;  
}
```

Iniciamos el condicional con la palabra reservada **switch** y, entre paréntesis, la expresión/condición que queremos evaluar.

En este caso vamos a **evaluar qué valor tiene la variable edad**.

# {código}

```
let edad = 5;
```

```
switch (edad) {
```

```
  case 10:
```

```
    console.log('Tiene 10 años');
```

```
    break;
```

```
  case 5:
```

```
    console.log('Tiene 5 años');
```

```
    break;
```

```
}
```

Por cada caso escribimos la palabra reservada **case** y, a continuación, el valor que queremos evaluar.

En este caso, **preguntamos si** el valor de la variable **edad es 10**.

Como este caso **NO es verdadero**, JavaScript ignora el código de este caso y pasa a evaluar el siguiente.

# {código}

```
let edad = 5;
```

```
switch (edad) {  
  case 10:  
    console.log('Tiene 10 años');  
    break;  
  case 5:  
    console.log('Tiene 5 años');  
    break;  
}
```

Este caso es verdadero, por lo tanto, se ejecutará el código del bloque.

La palabra reservada **break** corta la ejecución del switch.

Si olvidamos el **break**, los bloques se seguirán ejecutando sin importar si los casos se cumplen o no.

# El bloque **default**

Si queremos considerar la posibilidad de que ninguno de los casos sea verdadero, utilizamos la palabra reservada **default** seguida de dos puntos `:` y el bloque de código que queramos que se ejecute.

Por lo general, escribimos el bloque default a lo último. En ese caso, no es necesario escribir el break.

```
{}  
    switch (expresión) {  
        case valorA:  
            // código a ejecutar si valorA es verdadero  
            break;  
        default:  
            // código a ejecutar si ningún caso es verdadero  
    }
```

# {código}

```
let fruta = 'wefwef';  
switch (fruta) {  
  case 'manzana':  
    console.log('Qué rica la manzana');  
    break;  
  case 'naranja':  
    console.log('¡Naranja, me encanta!');  
    break;  
  default:  
    console.log('¿Qué fruta es?');  
}
```

Definimos la expresión que vamos a evaluar en el switch.

En este caso queremos preguntar por el valor de la variable fruta.



# {código}

```
let fruta = 'wefwef';  
switch (fruta) {
```

```
  case 'manzana':  
    console.log('Qué rica la manzana');  
    break;
```

Este caso **es falso**, por lo tanto, no se ejecuta su código.

```
  case 'naranja':  
    console.log('¡Naranja, me encanta!');  
    break;
```

```
  default:  
    console.log('¿Qué fruta es?');
```

```
}
```

# {código}

```
let fruta = 'wefwef';  
switch (fruta) {  
  case 'manzana':  
    console.log('Qué rica la manzana');  
    break;  
  case 'naranja':  
    console.log('¡Naranja, me encanta!');  
    break;  
  default:  
    console.log('¿Qué fruta es?');  
}
```

Este caso **también es falso**, por lo tanto, no se ejecuta su código.

# {código}

```
let fruta = 'wefwef';  
switch (fruta) {  
  case 'manzana':  
    console.log('Qué rica la manzana');  
    break;  
  case 'naranja':  
    console.log('¡Naranja, me encanta!');  
    break;  
  default:  
    console.log('¿Qué fruta es?');  
}
```

Como ningún caso fue verdadero, se ejecuta el código dentro del bloque default.

**DigitalHouse** >  
Coding School