

JS

JavaScript

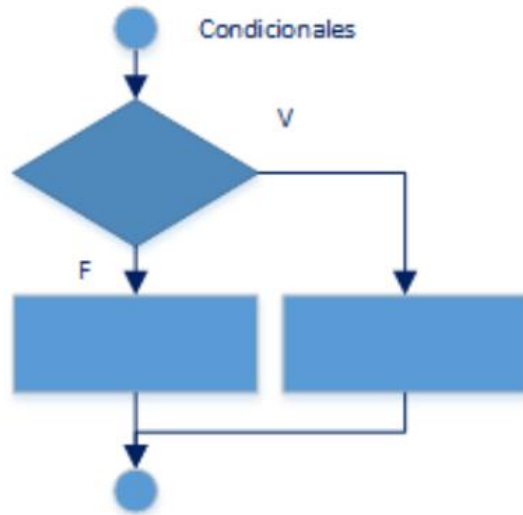
Control de flujo

Estructuras de control

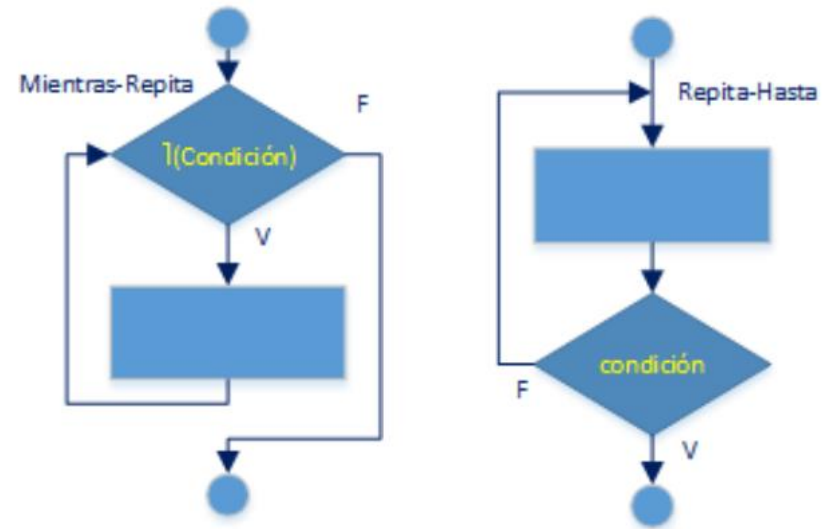


Las estructuras de control permiten controlar el flujo de ejecución de un programa. Existen dos tipos:

CONDICIONALES



REPETITIVAS



Estructuras condicionales *if*



Una ***condicional*** ejecuta un ***bloque*** de código se ejecute ***si se cumple una determinada condición***:

Sintaxis:

```
if ( condición ) {  
    bloque de código a ejecutar si condición es cierta  
}
```

- El bloque de código controlado por la condicional se delimita entre llaves ***{ .. }***

Estructuras condicionales *if*



Ejemplo: El siguiente código solicita al usuario su nombre, y se indica “Roger” se muestra por consola un mensaje de bienvenida:

```
let nombre = prompt("Cual es tu nombre", "");  
if ( nombre == "Roger")           // Condicion  
{                                  // Bloque de código -> Se ejecuta si la condicion es cierta  
    console.log("Hola " + nombre)  
    console.log("Bienvenido");  
}                                  // Fin bloque de código.  
// Código no condicionado -> Se ejecuta independientemente de la condición  
console.log("Adios")
```

Al final se indica el mensaje “Adios” ponga el nombre que ponga.

Estructuras condicionales *if*



Resultados de la ejecución según el nombre indicado:

localhost dice

Cual es tu nombre

Aceptar

Cancelar

localhost dice

Cual es tu nombre

Aceptar

Cancelar

Si se cumple la condición:

Hola Roger

Bienvenido

Adios

prueba.js:6

prueba.js:7

prueba.js:10

Si no se cumple la condición:

Adios

prueba.js:10

Estructura condicional *if .. else*



Puede añadirse una cláusula ***else*** para añadir un segundo bloque de código que se ejecute alternativamente si la condición es falsa.

Sintaxis:

```
if ( condición ) {  
    bloque de código a ejecutar si condición es cierta  
} else {  
    bloque de código a ejecutar si condición NO es cierta  
}
```



Estructura condicional *if .. else*

Ejemplo: El siguiente código solicita al usuario su nombre, si se indica “Roger” se muestra por consola un mensaje de bienvenida.

```
let nombre = prompt("Cual es tu nombre", "");
```

```
if ( nombre == "Roger")           // Condicion
{                                  // Bloque de código -> Se ejecuta si condicion es cierta.
    console.log("Hola " + nombre)
    console.log("Bienvenido");
}                                  // Fin bloque de código.
```

```
else
{                                  // Bloque de código -> Se ejecuta si condicion es falsa.
    console.log("Usuario desconocido!!");
}                                  // Fin bloque de código.
```

```
// Código no condicionado -> Se ejecuta independientemente de la condición
console.log("Adios")
```

- En caso contrario se muestra el mensaje “*Usuario desconocido.*”. Al final se muestra el mensaje “*Adios.*”.

Estructura condicional *if .. else*



Resultados de la ejecución según el nombre indicado:

localhost dice

Cual es tu nombre

Roger

Aceptar

Cancelar

localhost dice

Cual es tu nombre

Ivan

Aceptar

Cancelar

Si la condición se cumple:

Hola Roger

Bienvenido

Adios

prueba.js:6

prueba.js:7

prueba.js:10

Si la condición no se cumple:

Usuario desconocido!!

Adios

prueba.js:12

prueba.js:16

Estructura condicional *if.. else if*



Puede añadirse una condición anidada en la cláusula ***else*** que se evalúa únicamente si las condición inicial es *falsa*:

Sintaxis:

```
if ( condición_1 ) {  
    bloque de código a ejecutar si condición_1 es cierta  
} else if ( condición_2 ) {  
    bloque de código si condicion_1 es falsa, pero condicion_2 es cierta.  
} else {  
    bloque de código si condicion_1 y condicion_2 son falsas.  
}
```

Estructura condicional *if..else if*



Ejemplo: El siguiente código solicita al usuario que adivine un valor secreto mostrando si el valor indicado es demasiado alto, bajo o acierta.

```
let secreto = 10
let numero = Number(prompt("Adivina el numero secreto: ", ""));

if ( numero > secreto ) {           // Es demasiado alto?
    alert("Demasiado alto.");
} else if ( numero < secreto ) {    // NO: Es demasiado bajo?
    alert("Demasiado bajo.");
} else {                           // NO -> son iguales.
    alert("Correcto!");
}
```

- En cualquier caso, uno y sólo uno de los bloques de código se ejecutará.

Estructura condicional *switch*



El condicional de rama múltiple define ***múltiples bloques de código*** de los que sólo se ejecuta el correspondiente al valor de una variable o el predeterminado:

Sintaxis:

```
switch( variable ) {  
    case valor1: {           // Se ejecuta si variable == valor1  
        instrucciones;  
    }; break;  
    case valor2: {           // Se ejecuta si variable == valor2  
        instrucciones;  
    }; break;  
    default: { // Se ejecuta si no ejecuta ninguno de los bloques anteriores.  
        instrucciones;  
    }  
}
```

Estructura condicional *switch*



Ejemplo: El siguiente código solicita al usuario una letra de talla e indicar a cuál se corresponde:

```
let talla = prompt('Indica la talla: ');
switch( talla ) {
  case "S": {
    alert("Talla pequeña.");
  }; break;
  case "M": {
    alert("Talla mediana.");
  }; break;
  case "L": {
    alert("Talla grande.");
  }; break;
  default: {
    alert("Talla incorrecta.");
  };
}
```

Si la talla indicada no se corresponde con ninguna de las letras “S”, “M” o “L” se muestra el mensaje “*Talla desconocida*”.

Estructura condicional *switch*



Ejemplo: Es posible agrupar los valores de modo que varios permitan la ejecución de un bloque de código común:

```
let talla = prompt('Indica la talla: ');
switch( talla ) {
  case "S": {
    alert("Talla pequeña.");
  }; break;
  case "M": {
    alert("Talla mediana.");
  }; break;
  case "L": {
    alert("Talla grande.");
  }; break;
  case "XL":
  case "XXL": {
    alert("Talla muy grande.");
  }; break;
  default: {
    alert("Talla incorrecta.");
  };
}
```

En el código, los valores “XL”, “XXL” provocan la ejecución del mismo ***bloque***.



Estructuras repetitivas (bucles)

Un ***bucle*** ejecuta un ***bloque*** de código ***repetidamente***.

Mientras se cumple una condición.

- Bucles ***while***
- Bucles ***do..while***

Un número de veces conocido.

- Bucle ***for***



Estructura repetitiva *do..while*

- Bucle *do .. while* → Ejecuta un *bloque* de código una o más veces *mientras* se cumpla una condición.

```
do {  
    bloque de código a repetir  
} while ( condición )
```

Estructura repetitiva *do..while*



Ejemplo: El siguiente código solicita al usuario un valor secreto repetidamente hasta que lo acierta:

```
let secreto = 10
let numero;
do {
    // Solicita numero secreto
    numero = Number(prompt("Adivina el numero secreto: ", ""));
}
while( numero != secreto ); // Repetir mientras no acierte.
alert("Correcto");
```

El código del bucle se ejecuta al menos una vez como mínimo. Al usuario se le solicitará al menos una vez la clave en el mejor de los casos.



Estructura repetitiva *while*

- **Bucle *while*** → Ejecuta un ***bloque*** de código ninguna o más veces ***mientras*** se cumpla una condición.

```
while ( condición )  
{  
    bloque de código a repetir  
}
```

Estructura repetitiva *while*



Ejemplo: El siguiente código solicita al usuario un valor secreto repetidamente hasta que lo acierta:

```
let secreto = 10
let numero;
// Solicita número secreto por primera vez.
numero = Number(prompt("Adivina el numero secreto: ", ""));
while( numero != secreto )           // Repetir mientras no acierte.
{
    // Vuelve a solicitar el número secreto.
    numero = Number(prompt("Vuelve a intentarlo: ", ""));
}
alert("Correcto");
```

El código del bucle puede no ejecutarse si la condición es inicialmente falsa y el usuario indica el número secreto correcto a la primera.



Estructura repetitiva *for*

Ejecuta un ***bloque*** de código ***cierto numero de veces***:

```
for ( inicio, condición, paso )  
{  
    bloque de código a repetir  
}
```

El bucle emplea una ***variable contador*** que contabiliza las ejecuciones del ***bloque*** de código desde un valor de inicio a un final aplicando un paso por ejecución.

Estructura repetitiva *for*



Ejemplo: El siguiente código muestra un bucle que se ejecuta 5 veces según los valores que recorre la **variable contador** comprendidos entre 0 y 4.

```
for ( let i = 0; i < 5; i++) {  
    console.log("Valor de i: ", i);  
}
```

| | |
|---------------|-------------|
| Valor de i: 0 | script.js:5 |
| Valor de i: 1 | script.js:5 |
| Valor de i: 2 | script.js:5 |
| Valor de i: 3 | script.js:5 |
| Valor de i: 4 | script.js:5 |

Partes:

- **Asignación** → Inicia la **variable contador** (i) con valor inicial 0.
- **Comprobación** → El bucle continua mientras la **variable contador** (i) es inferior a 5.
- **Paso** → La **variable contador** (i) se incrementa en 1 por cada ejecución.

Estructura repetitiva *for*



Si la **variable contador** se declara en el bucle (inline), sólo existe en el interior:

```
for ( let i = 0; i < 5; i++) {  
  console.log("Valor de i: ", i);  
}  
console.log("Valor de i: ", i); // ERROR. Variable no definida.
```

Si la **variable contador** se declara fuera del bucle, su último valor permanece tras terminar el bucle:

```
let i;  
for ( i = 0; i < 5; i++) {  
  console.log("Valor de i: ", i);  
}  
console.log("Valor de i: ", i); // Muestra 5.
```