SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

CIEF

MÓDULO 1. MF0951_2
INTEGRAR COMPONENTES SOFTWARE EN PÁGINAS WEB

UNIDAD FORMATIVA 1.
UF1305 INTEGRAR COMPONENTES SOFTWARE EN PÁGINAS WEB.

CIEF

@mihifidem
creativity is intelligence having fun

@mihifidem

creativity is intelligence having fun

SOC Servei d'Ocupació de Catalunya

Generalitat de Catalunya

MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

ÍNDEX

3.4 **Estructuras de control.**

_Sentencia IF.

_Sentencia IF ELSE

_Sentencia IF ELSE anidados

_Sentencia SWITCH.

_Sentencia WHILE & DO WHILE

_Sentencia FOR.

_Sentencia BREAK & CONTINUE

SOC Servei d'Ocupació de Catalunya

Generalitat de Catalunya

MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

"Primero aprende informática y toda la teoría. Después desarrolla un estilo de programación. Entonces, olvídalo todo y hackea“

-- George Carrette

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social
Servicio Público de Empleo Estatal SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

CIEF

3.

Javascript.I

SOC Servei d'Ocupació de Catalunya

Generalitat de Catalunya

MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

CIEF

3.4.

Estructuras de control

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control.

Las estructuras de control básicas son:

- Secuencias
- Estructuras de selección o condicionales
- Repetición o bucles

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social · Servicio Público de Empleo Estatal · SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control.Anidamiento (nesting)

Comparar A con B

Si es distinto, verificar si es mayor

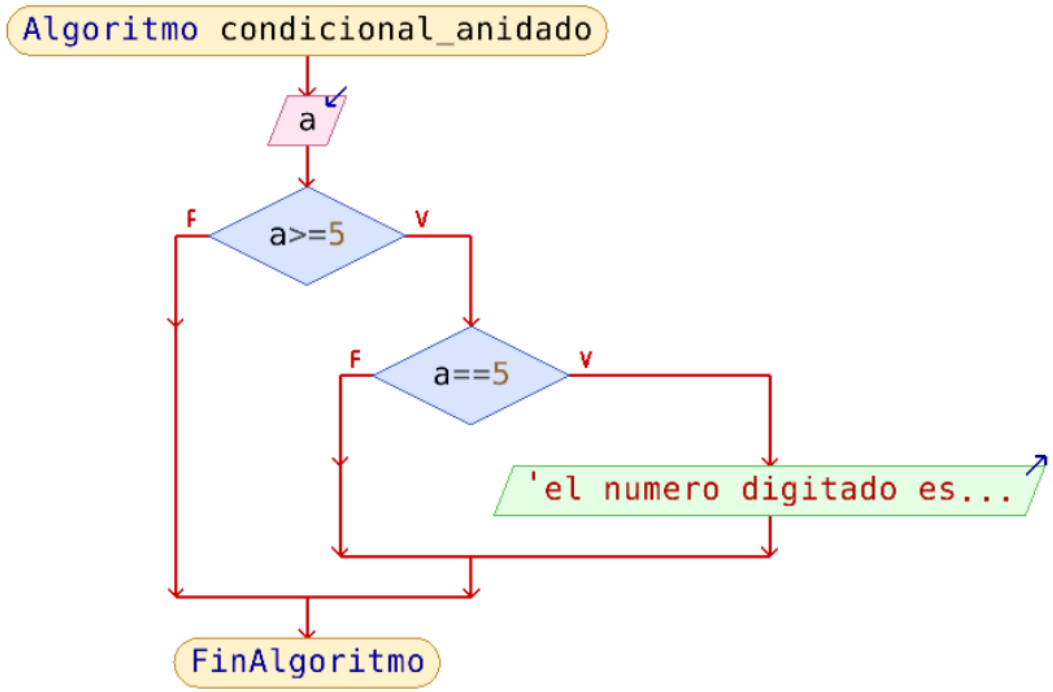
------Si es mayor, sumar A + B

----------Si el resultado es par, impr

----------Si el resultado es impar, gr

------Si no es mayor, restar

Si no es distinto, multiplicar



3.4 _Estructuras de control._Estructuras de control del flujo y condicional

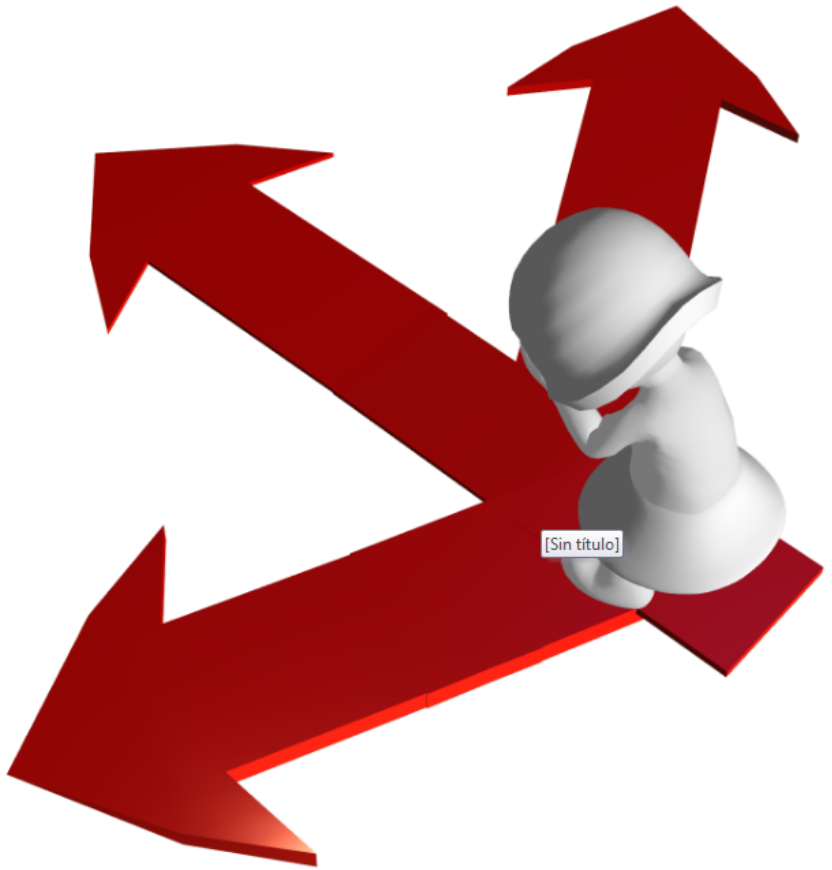
Los programas que se pueden realizar utilizando solamente variables y operadores son una simple sucesión lineal de instrucciones básicas.

Sin embargo, no se pueden realizar programas que muestren un mensaje si el valor de una variable es igual a un valor determinado y no muestren el mensaje en el resto de casos. Tampoco se puede repetir de forma eficiente una misma instrucción, como por ejemplo sumar un determinado valor a todos los elementos de un array.

Para realizar este tipo de programas son necesarias **las estructuras de control de flujo,** que son instrucciones del tipo "**si se cumple esta condición, hazlo; si no se cumple, haz esto otro**". También existen instrucciones del tipo "**repite esto mientras se cumpla esta condición**".

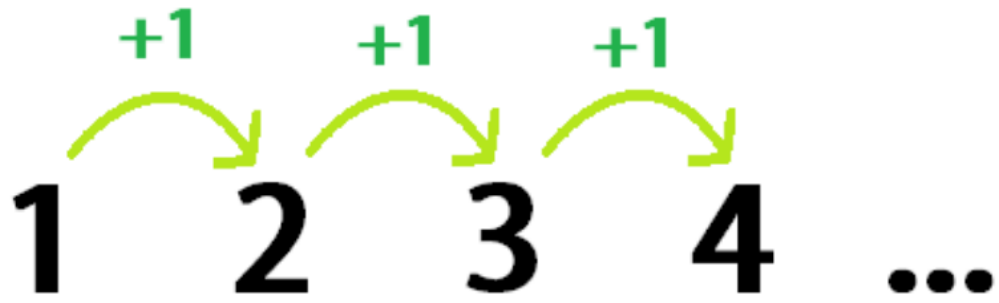
3.4 _Estructuras de control._Estructuras de control del flujo y condicional

Si se utilizan estructuras de control de flujo, los programas dejan de ser una sucesión lineal de instrucciones para convertirse en **programas inteligentes que pueden tomar decisiones** en función del valor de las variables.



Las secuencias sirven para agrupar varias instrucciones.

En Javascript esto se hace poniendo las instrucciones que queremos incluir en la secuencia entre llaves, {...}.

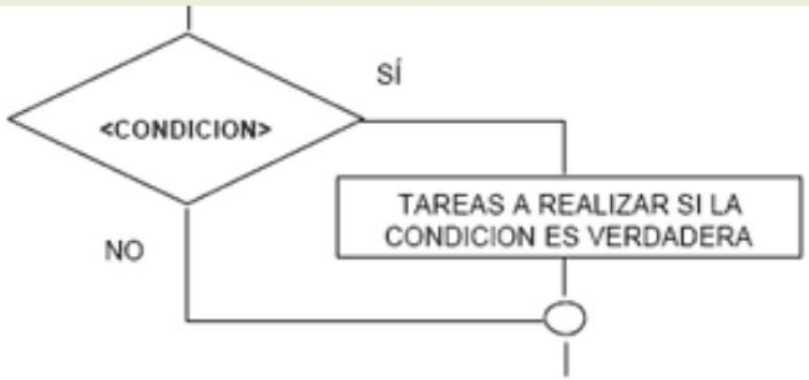


3.4 _Estructuras de control._if

Las estructuras de elección o condicionales permiten ejecutar o no una secuencia de instrucciones en función de un condición.

En JavaScript existen las estructuras condicionales

- **if**
- **if-else**
- **switch.**



SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._if

La estructura condicional if no difiere del resto de los lenguajes de programación.

Si la **condición** especificada entre paréntesis **es verdadera**, **se ejecuta la secuencia** de instrucciones asociada.

```
if(condición){

 instrucciones;

 }
```

Si solo queremos ejecutar una instrucción, no hace falta usar las llaves.

```
 if(condición)

 instrucción;
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._if els

Se puede añadir una **cláusula else** con las instrucciones que se ejecutarán **si la condición del if no se cumple.**

```
if (condicion){

 instrucciones_verdadera;

 } else {

instrucciones_falsa;

 }
```

También nos podemos ahorrar los paréntesis si solo hay una instrucción.

4.4. Condicionales._Estructuras ...

```
if (condición)

 instruccion_verdadera;

 else instruccion_falsa;
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

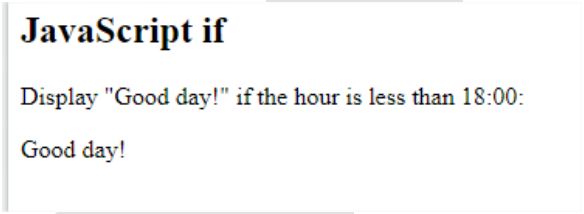
3.4 _Estructuras de control._if else

```
if (hour < 18) {
  greeting = "Good day";
}
```

```
if (hour < 18) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
```

```
if (time < 10) {
  greeting = "Good morning";
} else if (time < 20) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript if</h2>

<p>Display "Good day!" if the hour is less than 18:00:</p>

<p id="demo">Good Evening!</p>

<script>
if (new Date().getHours() < 18) {
  document.getElementById("demo").innerHTML = "Good day!";
}
</script>

</body>
</html>
```



3.4 _Estructuras de control._if else anidados

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript if .. else</h2>

<p>A time-based greeting:</p>

<p id="demo"></p>

<script>
const time = new Date().getHours();
let greeting;
if (time < 10) {
  greeting = "Good morning";
} else if (time < 20) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
document.getElementById("demo").innerHTML = greeting;
</script>

</body>
</html>
```

JavaScript if .. else

A time-based greeting:

Good day

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._if

La estructura condicional **switch** permite agrupar múltiples sentencias condicionales.

Por ejemplo, en lugar de

```
if(a==1){
document.write('Es un 1');
}else if(a==2){ document.write('Es un 2');
}elseif(a==3){
document.write('Es un 3');
}else{
 document.write('No es ni 1, ni 2, ni 3');
}
```

Se puede escribir

```
switch(a){
case 1: document.write('Es un 1'); break;
case 2: document.write('Es un 2'); break;
case 3: document.write('Es un 3'); break;
default: document.write ('No es ni 1, ni 2, ni 3');
}
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._if

La estructura condicional **switch** permite agrupar múltiples sentencias condicionales.

Se puede observar que al final de cada case aparece la sentencia **break**.

Sirve para salir del switch, de manera que no se ejecuten las instrucciones que aparezcan en los otros case.

Si no aparece, se ejecutan todas las instrucciones por debajo del primer case que se evalúa como verdadero, sin comprobar las condiciones correspondientes

```
if(a==1){
document.write('Es un 1');
}else if(a==2){ document.write('Es un 2');
}else if(a==3){document.write('Es un 3');
}else{
 document.write('No es ni 1, ni 2, ni 3');
}
```

Se puede escribir

```
switch(a){
case 1: document.write('Es un 1'); break;
case 2: document.write('Es un 2'); break;
case 3: document.write('Es un 3'); break;
default: document.write ('No es ni 1, ni 2, ni 3');
}
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._switch

```
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
     day = "Tuesday";
    break;
  case 3:
    day = "Wednesday";
    break;
  case 4:
    day = "Thursday";
    break;
  case 5:
    day = "Friday";
    break;
  case 6:
    day = "Saturday";
}
```

```
switch (new Date().getDay()) {
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
    break;
  default:
    text = "Looking forward to the Weekend";
}
```

```
switch (new Date().getDay()) {
  default:
    text = "Looking forward to the Weekend";
    break;
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
}
```

```
switch (new Date().getDay()) {
  case 4:
  case 5:
    text = "Soon it is Weekend";
    break;
  case 0:
  case 6:
    text = "It is Weekend";
    break;
  default:
    text = "Looking forward to the Weekend";
}
```

```
let x = "0";
switch (x) {
  case 0:
    text = "Off";
    break;
  case 1:
    text = "On";
    break;
  default:
    text = "No value found";
}
```

3.4 _Estructuras de control._switch

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript switch</h2>
<p id="demo"></p>
<script>
let day;
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
    day = "Tuesday";
    break;
  case 3:
    day = "Wednesday";
    break;
  case 4:
    day = "Thursday";
    break;
  case 5:
    day = "Friday";
    break;
  case  6:
    day = "Saturday";
}
document.getElementById("demo").innerHTML = "Today is " + day;
</script>
</body>
</html>
```

Bucle while

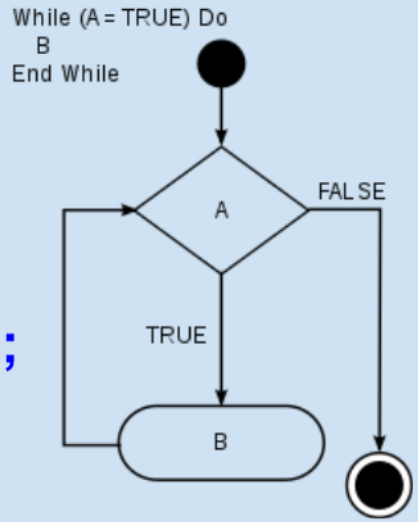
El **bucle while (mientras) se ejecuta mientras se cumpla una condición**.

La estructura general de un while es

```
while (condición){
instrucciones;
}
```

Veamos un ejemplo concreto. El siguiente fragmento de código muestra los números del 0 al 4:

```
var num1=0;
while(num1<5){
console.log(num1);
num1++; }
```



Bucle do … while

El bucle do-while (haz-mientras), es muy parecido al anterior.

La diferencia está en la condición, se evalúa al final del bucle en lugar de al principio.

Esto hace que este bucle se ejecute por lo menos una vez

```
do{
 instrucciones;
} while (condición);
```

Veamos un ejemplo concreto. El siguiente fragmento de código muestra los números del 0 al 4:

```
var num1=0;
    do{
                console.log(num1);
                num1++;
   }while (num1 <5);
```

3.4 _Estructuras de control._while & do while

```
while (i < 10) {
  text += "The number is " + i;
  i++;
}
```

```
do {
  text += "The number is " + i;
  i++;
}
while (i < 10);
```

```
const cars = ["BMW", "Volvo", "Saab", "Ford"];
let i = 0;
let text = "";

for (;cars[i];) {
  text += cars[i];
  i++;
}
```

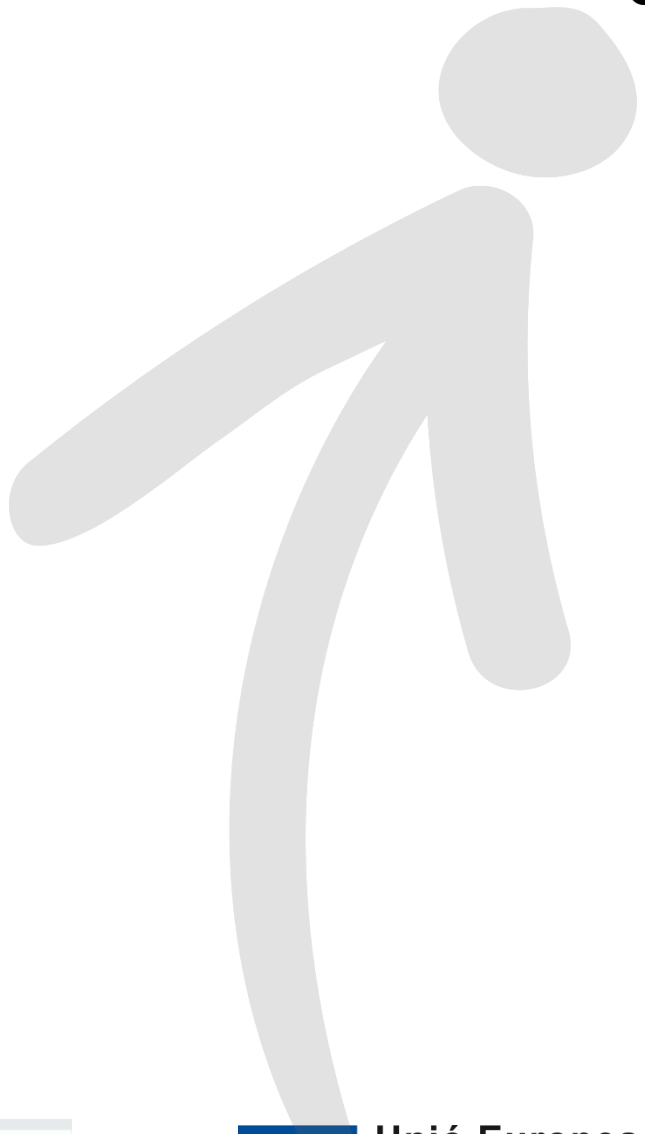
```
const cars = ["BMW", "Volvo", "Saab", "Ford"];
let i = 0;
let text = "";

while (cars[i]) {
  text += cars[i];
  i++;
}
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._while

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
const cars = ["BMW", "Volvo", "Saab", "Ford"];

let i = 0;
let text = "";
while (cars[i]) {
  text += cars[i] + "<br>";
  i++;
}

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

BMW
Volvo
Saab
Ford



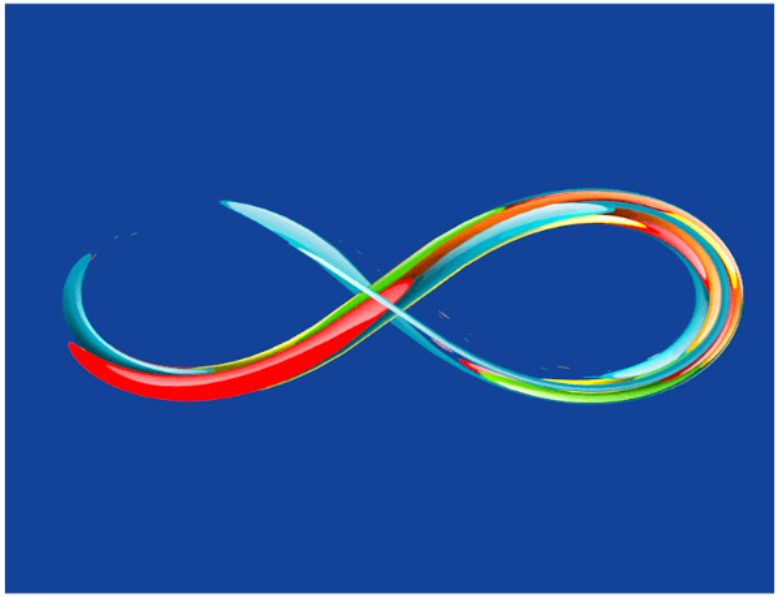
Bucles

Los bucles son estructuras de repetición.

Las instrucciones situadas dentro del bucle se repiten un cierto número de veces o mientras se cumpla cierta condición.

En Javascript tenemos los habituales bucles

- **for**
- **while**
- **do-while.**



3.4 _Estructuras de control._for

El **bucle for** un tiene una sintaxis algo más compleja que los anteriores.

Hay que definir la condición de permanencia en el bucle y las instrucciones de inicialización e incremento.

La estructura general se puede escribir así

```
for(instrucciones inicialización; condición; instrucciones incremento){

   instrucciones

}
```

```
for(var i= 0; i< 5;i++){

   console.log(i);

   }
```

También es habitual hacer que la variable del bucle sea decreciente en lugar de creciente.

```
for(var i=5;i > 0; i--){

   console.log(i);

   }
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

Break y Continue

En JavaScript también están disponibles las sentencias break y continue.

Ya hemos visto el break al explicar el switch y aquí tiene una función parecida.

Si se ejecuta un break, el bucle actual finaliza.

El siguiente bucle muestra de nuevo los números del 0 al 4.

```
vari=0;
    while(1){
    console.log(i);
    i++;
   if(i>=5) break;
    };
```

3.4 _Estructuras de control._for

En lugar de escribir:

```
text += cars[0] + "<br>";
text += cars[1] + "<br>";
text += cars[2] + "<br>";
text += cars[3] + "<br>";
text += cars[4] + "<br>";
text += cars[5] + "<br>";
```

Puedes escribir:

```
for (let i = 0; i < cars.length; i++) {
  text += cars[i] + "<br>";
}
```

```
for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

Declaración 1

Normalmente usará la instrucción 1 para inicializar la variable utilizada en el bucle (sea i = 0).
Este no es siempre el caso, a JavaScript no le importa. La declaración 1 es opcional.
Puede iniciar muchos valores en la declaración 1 (separados por comas):

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript For Loop</h2>

<p id="demo"></p>

<script>
const cars = ["BMW", "Volvo", "Saab", "Ford"];
let i, len, text;
for (i = 0, len = cars.length, text = ""; i < len; i++) {
  text += cars[i] + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social · Servicio Público de Empleo Estatal SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._for

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript For Loop</h2>

<p id="demo"></p>

<script>
const cars = ["BMW", "Volvo", "Saab", "Ford"];
let i = 2;
let len = cars.length;
let text = "";

for (; i < len; i++) {
  text += cars[i] + "<br>";
}

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

3.4 _Estructuras de control._for

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript For Loop</h2>

<p id="demo"></p>

<script>
const cars = ["BMW", "Volvo", "Saab", "Ford"];

let i = 0;
let len = cars.length;
let text = "";

for (; i < len; ) {
  text += cars[i] + "<br>";
  i++;
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

3.4 _Estructuras de control._break in for

Break y Continue

El uso de **continue no es tan sencillo** de entender como el del break, especialmente si aparece en un bucle algo complicado, por lo que es menos habitual.

Al llegar a un continue, la iteración actual del bucle finaliza. Las instrucciones que estén por debajo del continúe no se ejecutan. Comienza una nueva iteración siempre que sigan cumpliendo las condiciones de permanencia en el bucle. Si es un bucle for, se ejecutan las instrucciones de incremento.

El siguiente ejemplo muestra la utilidad del continue.

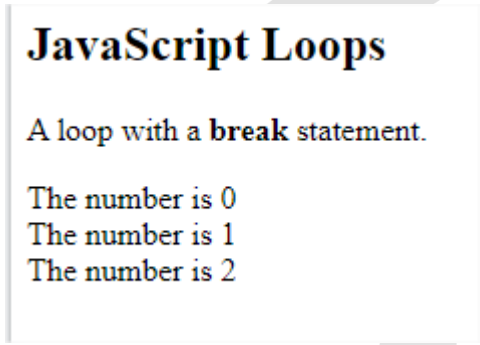
```
for(var i=0;i<5;i++){
    if(i==3){
    continue;
    } console.log(i);
    }
```

El bucle no muestra el número 3 por el continue. Cuando i vale 3 se ejecuta el continue y la instrucción que está después no se ejecuta.

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

3.4 _Estructuras de control._break in for

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Loops</h2>

<p>A loop with a <b>break</b> statement.</p>

<p id="demo"></p>

<script>
let text = "";
for (let i = 0; i < 10; i++) {
  if (i === 3) { break; }
  text += "The number is " + i + "<br>";
}

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```



SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

Grup CIEF

el teu partner

Barcelona

Francesc Tàrrega 14

08027 Barcelona

93 351 78 00

Madrid

Campanar 12

28028 Madrid

91 502 13 40

Reus

Alcalde Joan Bertran 34-38

43202 Reus

977 31 24 36

info@grupcief.com