

Desarrollo Web en Entorno Cliente

UD 01. Sintaxis Javascript ES6

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

ÍNDICE DE CONTENIDO

7.	Estructura repetitivas (Bucles)	3
7.1	Bucle for	3
7.2	Bucle while	4
7.3	Bucle do-while	5
7.4	Instrucciones BREAK y CONTINUE	6
8.	Arrays	7

UD01. SINTAXIS JAVASCRIPT ES6

1. ESTRUCTURA REPETITIVAS (BUCLES)

1.1 Bucle for

Cuando la ejecución de un programa llega a un bucle for:

- Lo primero que hace es ejecutar la “Inicialización del índice”, que solo se ejecuta una vez.
- A continuación, analiza la condición de prueba y si esta se cumple ejecuta las instrucciones del bucle.
- Cuando finaliza la ejecución de las instrucciones del bucle se realiza la modificación del índice, se retorna a la cabecera del bucle for y se realiza de nuevo la condición de prueba.
- Si la condición se cumple se ejecutan las instrucciones y si no se cumple la ejecución continúa en las líneas de código que siguen posteriores al bucle.

Sintaxis

```
for (Inicialización del índice; Condición de prueba; Modificación en el índice){  
  // ...instrucciones...  
}
```

Ejemplo: números pares del 2 al 30

```
for (i=2;i<=30;i+=2) {  
  console.log(i);  
}  
console.log("Se han escrito los números pares del 2 al 30");
```

Ejemplo: Escribe las potencias de 2 hasta 3000

```
let aux=1;  
for (i=2;i<=3000;i*=2) {  
  console.log("2 elevado a "+aux+" es igual a "+i);  
  aux++;  
}  
console.log("Se han escrito las potencias de 2 menores de 3000");
```

Ejercicio 1

Busca el error de sintaxis en este código .js

```
for (let j=0; j<3; j++ {  
    console.log("Ciclo: " + j);  
}
```

Ejercicio 2

Completa el siguiente código .js para sume los números de 1 a 10 y muestre el resultado por consola

```
let total = 0;  
for (let i=1; _____ ; i++) {  
    total += i;  
}  
console.log(total);
```

1.2 Bucle while

Con el bucle while se pueden ejecutar un grupo de instrucciones mientras se cumpla una condición.

- Si la condición nunca se cumple, entonces tampoco se ejecuta ninguna instrucción.
- Si la condición se cumple siempre, nos veremos inmersos en el problema de los bucles infinitos, que pueden llegar a colapsar el navegador, o incluso el ordenador.
 - Por esa razón es muy importante que la condición deba dejar de cumplirse en algún momento para evitar bucles infinitos.

Sintaxis

```
while (condición){  
    //...instrucciones...  
}
```

Ejemplo: Escribe los números pares de 0 a 30

```
let i=2;  
while (i<=30) {  
    console.log(i);  
    i+=2;  
}  
console.log("Ya se han mostrado los números pares del 2 al 30");
```

Ejemplo: Pregunta una clave hasta que se corresponda con una dada.

```
let auxclave="";
while (auxclave!="vivaYO"){
    auxclave=prompt("introduce la clave ","claveSecreta")
}
console.log("Has acertado la clave");
```

Ejercicio 3

Completa el siguiente código .js para calcular la suma de los números del 1 al 9

```
let suma = 0;
let numero = 1;
while (_____ < 10) {
    suma += numero;
    numero____;
}
console.log(suma);
```

Ejercicio 4

Detecta el error del siguiente código .js

```
let suma;
let numero = 1;
while (numero < 10) {
    suma += numero;
    numero++;
}
console.log(suma);
```

1.3 Bucle do-while

La diferencia del bucle do-while frente al bucle while reside en el momento en que se comprueba la condición: el bucle do-while no la comprueba hasta el final, es decir, después del cuerpo del bucle, lo que significa que el bucle do-while se recorrerá, una vez, como mínimo, aunque no se cumpla la condición.

Sintaxis

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

Ejemplo: Preguntar por una clave hasta que se introduzca la correcta

```
let auxclave;
do {
    auxclave=prompt("introduce la clave ", "vivaYo")
} while (auxclave!="EstaeslaclaveJEJEJE")
console.log("Has acertado la clave");
```

Ejercicio 5. Completa el ejercicio6 del tema anterior comprobando que el rango sea solo un numero entre 1 y 4 y no permitas que el salario sea mayor de 20000€ ni menor de 1000€.

Ejercicio 6. Realiza un programa que pida 3 números y muestre con un alert() el número que sea mayor. Al finalizar, pedirá al usuario si quiere continuar o por lo contrario finalizar la ejecución del programa.

1.4 Instrucciones BREAK y CONTINUE

En los bucles for, while y do-while se pueden utilizar las instrucciones break y continue para modificar el comportamiento del bucle.

La instrucción “break” dentro de un bucle hace que este se interrumpa inmediatamente, aun cuando no se haya ejecutado todavía el bucle completo. Al llegar la instrucción, el programa se sigue desarrollando inmediatamente a continuación del final del bucle.

Ejemplo: Pregunta por la clave y permitir tres respuestas incorrectas

```
let numveces=0;
//Mientras no introduzca la clave
while (auxclave !== "anonimo"){
    auxclave=prompt("Introduce la clave ", "");
    numveces++;
    if (numveces === 3)
        break;
}
if (auxclave=="SuperClave"){
    console.log("La clave es correcta");
}else{
    console.log("La clave no es correcta correcta");
}
```

El efecto que tiene la instrucción “continue” en un bucle es el de hacer retornar a la

secuencia de ejecución a la cabecera del bucle, volviendo a ejecutar la condición o a incrementar los índices cuando sea un bucle for. Esto permite saltarse recorridos del bucle.

Ejemplo: Presenta todos los números pares del 0 al 50 excepto los que sean múltiplos de 3

```
let i;  
for (i=2;i<=50;i+=2){  
    if ((i%3)===0)  
        continue;  
    console.log(i);  
}
```

Ejercicio 7. Crea un programa que pregunte un usuario y contraseña hasta que introduzca los datos correctos. Solo tendrá 6 intentos.

2. Arrays

Un array (también llamado vector o arreglo) es una variable que contiene diversos valores. Lo creamos simplemente con “[]” o con “new Array()” o inicializando los elementos. Podemos referenciar los elementos de un array indicando su posición.

Los arrays poseen una propiedad llamada “length” que podemos utilizar para conocer el número de elementos del array.

```
// Array definido 1 a 1  
let miVector=[]; // let miVector=new Array(); es equivalente  
miVector[0]=22;  
miVector[1]=12;  
miVector[2]=33;  
//Array definido en una línea inicializando elementos  
let otroArray=[1,2,"Cancamusa"]; // Valores dentro del array  
console.log(miVector[1]);  
console.log(otroArray[2]);  
console.log(miVector.length ); // tamaño
```

Ejercicio 8

Completa el espacio en blanco para que se imprima "agua" al ejecutar el código.

```
let bebidas = [____, "te", "café"];  
console.log(bebidas[0]);
```

Ejercicio 9

Corrige el error del siguiente código para que se muestre por consola Italia

```
let paises = ["España" "Francia" "Italia"];  
console.log(paises.length);
```