

LENGUAJES DE SCRIPT DE CLIENTE III

Definición de BUCLE.

Un bucle nos permite automatizar un conjunto de acciones que se deban realizar de forma repetitiva con el fin de ahorrar código. Los bucles más utilizados en javascript son: for, do..while y while.

Por ejemplo, si me piden crear un programa para imprimir en pantalla los diez primeros números, puedo hacerlo de las dos siguientes maneras, una sin y otra con bucle. Cuál te parece más rápida?.

SIN BUCLE	CON BUCLE
document.write("1");	for (i=1; i<=10;i++)
document.write("2");	{
document.write("3");	document.write(i);
document.write("4");	}
document.write("5");	
document.write("6");	
document.write("7");	
document.write("8");	
document.write("9");	
document.write("10");	

Sintaxis del bucle FOR:

Siempre vamos a tener una variable numérica que se utilizará como índice del bucle.

for (inicio; condición parada; incremento)

{

Grupo de instrucciones que se ejecutan dentro del bucle for.

Si solo fuese una instrucción, nos podemos ahorrar las llaves.

}

Inicio: valor inicial del índice. En el ejemplo i=1.

Condicion parada: Expresa una condición que debe cumplirse para que el bucle siga ejecutándose. Cuando ya no se cumpla, el bucle parará. En el ejemplo: i<=10

Incremento: Es el valor con el que se incrementa el índice en cada paso del bucle. En nuestro ejemplo pone i++ (que equivale a incrementarla de uno en uno). Si el incremento fuese de dos en dos, podríamos poner i=i+2;

Vamos a poner un ejemplo explicado:

Programa que imprime los números pares del 1 al 100.

```
.....  
var i;  
for (i=1;i<=100;i++)  
{  
  if (i%2==0)  
    document.write (i);  
}
```

Definimos inicialmente la variable que utilizamos como índice. En este caso es i.

Comenzamos el bucle dándole el valor 1 a i en el inicio. El bucle estará ejecutándose mientras i sea menor o igual que 100. En ese momento, el bucle terminará. El incremento de i será de uno en cada paso. Esto ya sabemos que lo podemos expresar como i++ o como i=i+1.

Como me piden que solo imprima los pares, compruebo dentro del bucle el resto de dividir i entre dos. Si es cero, entonces el número i es par y lo imprimo. En caso contrario, no hago nada.

Contadores.

Un contador es una variable entera que la utilizamos para contar cuando ocurre un suceso concreto.

Un contador siempre ha de tener un valor inicial. Normalmente a cero.

Y dentro de un bucle se incrementará en el valor que nos interese cada vez que se produzca el evento.

En el ejemplo anterior, vamos a contar cuantos pares hay. (En javascript se utiliza la // para realizar comentarios en una línea o los caracteres /* ... */ para multilineas).

```
.....  
var i;  
var contadorpares=0; // Inicializamos a cero la variable que vamos a usar de contador.  
for (i=1;i<=100;i++)  
{  
  if (i%2==0)  
    document.write (i);  
    contadorpares=contadorpares+1; // Incrementamos en uno el valor del contador cada vez que se encuentre un número par  
}  
  
document.write ("Total Pares: "+contadorpares);
```

Acumuladores.

Un acumulador es una variable numérica que permite ir acumulando operaciones. Me permite ir haciendo operaciones parciales.

Un acumulador se **inicializa** a un valor inicial según la operación que se va a acumular: a 0 si es una suma o a 1 si es un producto (elemento neutro de la operación que va a realizar).

Vamos a repetir el ejercicio anterior, pero en esta ocasión sumando el total de números pares:

```
.....  
var i;  
var sumapares=0; // Inicializamos a cero la variable que vamos a usar de acumulador.  
for (i=1;i<=100;i++)  
{  
  if (i%2==0)  
    document.write (i);  
    sumapares=sumapares+i; // si el numero es par, se suma al acumulador  
}  
document.write ("Suma Pares: "+sumapares);
```

Bucles que van en orden inverso.

Hasta ahora hemos visto que el índice del bucle comienza en un número pequeño y termina en un número mayor. También se pueden definir bucles que comiencen en un numero grande y termine en uno más pequeño, es decir, cuentan hacia atrás.

El procedimiento es exactamente el mismo, con la diferencia de que el incremento es negativo, es decir, hay que ir restándole una cantidad al índice.

Por ejemplo, si queremos imprimir números del 25 al 1, haríamos el siguiente bucle:

```
.....  
var i;  
for (i=25; i>=1; i--)  
{  
  document.write (i);  
}
```

Fijaros que la condición que tiene que cumplirse para que el bucle siga ejecutándose es, en este caso, $i \geq 1$ (ahora es mayor o igual, y no menor o igual). Para decrementar una variable de uno en uno, podemos poner $i--$ (i menos menos) o también $i=i-1$.

EJERCICIOS con bucle FOR.

1. Realizar el ejercicio de contar pares y sumar pares en un mismo ejercicio.
2. Modificar el ejercicio de contar pares y sumar pares de manera que se hará desde 1 hasta un número que se pide por teclado. Si el número introducido es menor de 1, ha de imprimir un mensaje de que no se puede realizar.
3. Repetir el ejercicio anterior, pero realizando sumas y conteos de números pares por un sitio e impares por otro.
4. Crear un programa que pida dos números por teclado. Tendrá que devolver la suma de todos los comprendidos entre ellos dos (ambos incluidos).
5. Crear un programa en que pida 2 números por teclado, que serán la base y el exponente. Me tendrá que decir el resultado de elevar la base al exponente.
6. Programa que pida por teclado un número y me devuelva el factorial de ese número. El factorial de un número es el producto de todos los números comprendidos entre uno y el número. Por ej, el factorial de 5 es: $1 \times 2 \times 3 \times 4 \times 5 = 120$