

Lección 18

Estructuras Repetitivas

Loops



Bucle (Loop) y las Iteraciones

Un **bucle (Loop)** es utilizado para hacer una acción repetida sin tener que escribir varias veces el mismo código,

Un bucle está compuesto por (n) iteraciones.



Loops (Bucles) e Iteraciones

Imagina un programa que tenga que imprimir todos los números del 0 al 10:

```
console.log(0);  
console.log(1);  
console.log(2);  
console.log(3);  
console.log(4);  
...  
console.log(10);
```

Funciona!

Pero, ¿qué pasa si quiero imprimir los números del 0 al 100?

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.

While

- ❑ El *while loop* ejecuta una sentencia mientras la condición sea verdadera.
- ❑ La condición es evaluada antes de evaluar la sentencia.
- ❑ Permite crear bucles que se ejecutan ninguna o más veces, dependiendo de la condición indicada.

```
while (condición) {  
    sentencia  
}
```

```
var numero = 0;  
while (numero <= 10) {  
    console.log(numero);  
    numero = numero + 1;  
}
```

Ejemplo While

"Mientras se cumpla la condición indicada, repite indefinidamente las instrucciones incluidas dentro del bucle".

```
1  var resultado = 0;
2  var numero = 100;
3  var i = 0;
4
5  while(i <= numero) {
6      resultado += i;
7      i++;
8  }
9
10 alert(resultado);
```

El programa debe sumar todos los números menores o igual que otro dado. Por ejemplo si el número es 5, se debe calcular: $1 + 2 + 3 + 4 + 5 = 15$

For

La estructura **for** permite realizar repeticiones (también llamadas bucles) de una forma muy sencilla. No obstante, su definición formal no es tan sencilla como la de **if()**:

```
1 for(inicializacion; condicion; actualizacion) {  
2     ...  
3 }
```

"Mientras la condición indicada se siga cumpliendo, repite la ejecución de las instrucciones definidas dentro del for. Además, después de cada repetición, actualiza el valor de las variables que se utilizan en la condición".

For

El *for loop* está formado por tres expresiones que se escriben dentro de un paréntesis y separadas por punto y coma ";", seguidas por una sentencia:

```
for ([expresión-inicial]; [condición]; [expresión-incremento]){  
    sentencia  
}
```

Expresión-inicial:

Utilizada para inicializar una variable.

Condición:

Expresión para ser evaluada antes de cada iteración del loop.

Si la expresión es **verdadera**, la sentencia se ejecuta.

Expresión-incremento

Expresión evaluada al finalizar el loop. Ocurre antes de evaluar la siguiente condición.

Generalmente usado para actualizar o incrementar el contador.

Ejemplo For

```
for (var numero = 0; numero <= 100; numero++) {  
    console.log(numero);  
}  
  
for (var i = 0; i < 100; i++) {  
    console.log(i);  
};
```

CUIDADO

INFINITE LOOP

Otras estructuras



Do While

El bucle de tipo `do...while` es muy similar al bucle `while`, salvo que en este caso **siempre** se ejecutan las instrucciones del bucle al menos la primera vez. Su definición formal es:

```
1  do {  
2    ...  
3  } while(condicion);  
4
```

Ejemplo Do While

Utilizando este bucle se puede calcular fácilmente el factorial de un número:

```
1  var resultado = 1;
2  var numero = 5;
3
4  do {
5      resultado *= numero; // resultado = resultado * numero
6      numero--;
7  } while(numero > 0);
8
9  alert(resultado);|
```