JavaScript: Estructuras de control de flujo

Estructura for

Las estructuras **if** y **if...else** no son muy eficientes cuando se desea ejecutar de forma repetitiva una instrucción. Por ejemplo, si se quiere mostrar un mensaje cinco veces, se podría pensar en utilizar el siguiente **if**:

```
var veces = 0;
if(veces < 4) {
  alert("Mensaje");
  veces++;
}
```

JavaScript: Estructuras de control de flujo

Estructura for

Se comprueba si la variable veces es menor que 4. Si se cumple, se entra dentro del **if()**, se muestra el mensaje y se incrementa el valor de la variable veces. Así se debería seguir ejecutando hasta mostrar el mensaje las cinco veces deseadas.

Sin embargo, el funcionamiento real del script anterior es muy diferente al deseado, ya que solamente se muestra una vez el mensaje por pantalla.

JavaScript: Estructuras de control de flujo

Estructura for

La razón es que la ejecución de la estructura **if()** no se repite y la comprobación de la condición sólo se realiza una vez, independientemente de que dentro del **if()** se modifique el valor de la variable utilizada en la condición.

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

JavaScript: Estructuras de control de flujo

Estructura for

Sintaxis:

```
for (inicializacion; condicion; actualizacion) {
...
}
```

La idea del funcionamiento de un bucle **for** es la siguiente: "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".

JavaScript: Estructuras de control de flujo

Estructura for

```
for (inicializacion; condicion; actualizacion) {
...
}
```

- La "*inicialización*" es la zona en la que se establece los valores iniciales de las variables que controlan la repetición.
- La "*condición*" es el único elemento que decide si continua o se detiene la repetición.
- La "*actualización*" es el nuevo valor que se asigna después de cada repetición a las variables que controlan la repetición.

JavaScript: Estructuras de control de flujo

Estructura for

```
var mensaje = “Hola, estoy dentro de un bucle”;
for(var i = 0; i < 5; i++) {
alert("Mensaje");
}
```

La parte de la inicialización del bucle consiste en:

```
var i = 0;
```

Por tanto, en primer lugar se crea la variable **i** y se le asigna el valor de **0**. Esta zona de *inicialización* solamente se tiene en consideración justo antes de comenzar a ejecutar el bucle. Las siguientes repeticiones no tienen en cuenta esta parte de *inicialización*.

JavaScript: Estructuras de control de flujo

Estructura for

```
var mensaje = “Hola, estoy dentro de un bucle”;
for(var i = 0; i < 5; i++) {
alert("Mensaje");
}
```

La zona de condición del bucle es:

```
i < 5;
```

Los bucles se siguen ejecutando mientras se cumplan las condiciones y se dejan de ejecutar justo después de comprobar que la condición no se cumple. En este caso, mientras la variable **i** valga menos de 5 el bucle se ejecuta indefinidamente.

JavaScript: Estructuras de control de flujo

Estructura for

```
var mensaje = “Hola, estoy dentro de un bucle”;
for(var i = 0; i < 5; i++) {
alert("Mensaje");
}
```

Como la variable **i** se ha inicializado a un valor de **0** y la condición para salir del bucle es que **i** sea menor que **5**, si no se modifica el valor de i de alguna forma, el bucle se repetiría indefinidamente. Por ese motivo, es imprescindible indicar la zona de *actualización*, en la que se modifica el valor de las variables que controlan el bucle.

JavaScript: Estructuras de control de flujo

Estructura for

```
var mensaje = “Hola, estoy dentro de un bucle”;
for(var i = 0; i < 5; i++) {
alert("Mensaje");
}
```

La zona de actualización del bucle es:

```
i++;
```

En este caso, el valor de la variable **i** se incrementa en una unidad después de cada repetición. La zona de actualización se ejecuta después de la ejecución de las instrucciones que **i**ncluye el **for**.

JavaScript: Estructuras de control de flujo

Estructura for

```
var mensaje = “Hola, estoy dentro de un bucle”;
for(var i = 0; i < 5; i++) {
alert("Mensaje");
}
```

Así, durante la ejecución de la quinta repetición el valor de **i** será **4**. Después de la quinta ejecución, se actualiza el valor de i, que ahora valdrá **5**. Como la condición es que **i** sea menor que **5**, la condición ya no se cumple y las instrucciones del **for** no se ejecutan una sexta vez.

JavaScript: Estructuras de control de flujo

Estructura for

Normalmente, la variable que controla los bucles for se llama i, ya que recuerda a la palabra índice y su nombre tan corto ahorra mucho tiempo y espacio.
El ejemplo anterior que mostraba los días de la semana contenidos en un array se puede rehacer de forma más sencilla utilizando la estructura **for**:

JavaScript: Estructuras de control de flujo

Estructura for

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves",
"Viernes", "Sábado", "Domingo"];

for(var i=0; i<7; i++) {
  alert(dias[i]);
}
```

JavaScript: Estructuras de control de flujo

Ejercicio 7

El factorial de un número entero **n** es una operación matemática que consiste en multiplicar todos los factores n x (n-1) x (n-2) x ... x 1. Así, el factorial de 5 (escrito como 5!) es igual a: 5! = 5 x 4 x 3 x 2 x 1 = 120

Utilizando la estructura **for**, crear un script que calcule el factorial de un número entero.

JavaScript: Estructuras de control de flujo

Ejercicio 7

```
var numero = prompt("Introduce un número y se mostrará su factorial");
var resultado = 1;

for(var i=1; i<=numero; i++) {
  resultado *= i;
}
alert(resultado);
```