

13 Estructuras de control de flujo

Las estructuras de control de flujo son bloques de código que permiten que los programas dejen de ser una sucesión lineal de instrucciones para convertirse en programas inteligentes que pueden tomar decisiones en función del valor de las variables.

Estructura if

La estructura más utilizada en JavaScript y en la mayoría de lenguajes de programación es la estructura if. Se emplea para tomar decisiones en función de una condición. Su definición formal es:

```
if(expresión condicional) {  
    // código a ejecutar si se cumple la condición  
}
```

Si la condición se cumple (es decir, si su valor es true) se ejecutan todas las instrucciones que se encuentran dentro de {...}. Si la condición no se cumple (es decir, si su valor es false) no se ejecuta ninguna instrucción contenida en {...} y el programa continúa ejecutando el resto de instrucciones del script.

Estructura if...else

En ocasiones, las decisiones que se deben realizar no son del tipo "si se cumple la condición, hazlo; si no se cumple, no hagas nada". Normalmente las condiciones suelen ser del tipo "si se cumple esta condición, hazlo; si no se cumple, haz esto otro".

Para este segundo tipo de decisiones, existe una variante de la estructura if llamada if...else. Su sintaxis formal es la siguiente:

```
if(expresión condicional) {  
    ...  
}  
else {  
    ...  
}
```

La estructura if...else se puede encadenar para realizar varias comprobaciones seguidas.

Estructura switch (case)

Una estructura switch permite a un programa evaluar una expresión e intentar igualar el valor de dicha expresión a una etiqueta de caso (case). Si se encuentra una coincidencia, el programa ejecuta la sentencia asociada. La sintaxis se describe como se muestra a continuación:

```
switch (expresión) {  
  case etiqueta_1:  
    sentencias_1  
    [break;]  
  case etiqueta_2:  
    sentencias_2  
    [break;]  
  ...  
  default:  
    sentencias_por_defecto  
    [break;]  
}
```

El programa primero busca una cláusula case con una etiqueta que coincida con el valor de la expresión y, entonces, transfiere el control a esa cláusula, ejecutando las sentencias asociadas a ella. Si no se encuentran etiquetas coincidentes, el programa busca la cláusula opcional default y, si se encuentra, transfiere el control a esa cláusula, ejecutando las sentencias asociadas. Si no se encuentra la cláusula default, el programa continúa su ejecución por la siguiente sentencia al final del switch. Por convención, la cláusula por defecto es la última cláusula, aunque no es necesario que sea así.

La sentencia opcional break asociada con cada cláusula case asegura que el programa finaliza la sentencia switch una vez que la sentencia asociada a la etiqueta coincidente es ejecutada y continúa la ejecución por las sentencias siguientes a la sentencia switch. Si se omite la sentencia break, el programa continúa su ejecución por la siguiente sentencia que haya en la sentencia switch.

Estructura o ciclo for

La estructura for permite realizar bucles de una forma muy sencilla. Su sintaxis formal es la siguiente:

```
for(inicialización; condición; actualización) {  
    // código a ejecutar mientras se cumpla la condición  
}
```

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".

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 continúa 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.

Estructura o ciclo while

La estructura while ejecuta un simple bucle, mientras se cumpla la condición. Su sintaxis formal es la siguiente:

```
while(condición) {  
    // código a ejecutar mientras se cumpla la condición  
}
```

La idea del funcionamiento de un bucle while es la siguiente: "mientras la condición indicada se siga cumpliendo, repite la ejecución de las instrucciones definidas dentro del while. Es importante modificar los valores de las variables incluidas dentro de la condición, ya que otra manera, el bucle se repetiría de manera indefinida, perjudicando la ejecución de la página y bloqueando la ejecución del resto del script.

Estructura do... while

Similar a while, con el ciclo de tipo do.. while primero ejecutamos el bloque de instrucciones y posteriormente hacemos la validación de la condición. La sintaxis es:

```
do {  
    // código a ejecutar mientras se cumpla la condición  
} while (condición)
```

Al igual que en el ciclo de tipo while, es responsabilidad del programador controlar la actualización de las variables y manejar la condición, por lo que podemos caer en un ciclo infinito que bloquee el navegador.

Estructura try... catch

La estructura try...catch define un bloque de código a intentar ejecutar en try, y especifica una respuesta si se produce una excepción en catch. Su sintaxis es:

```
try {  
    // Código a ejecutar  
}  
catch (error) {  
    // Código a ejecutar si se produce una excepción en el bloque anterior  
}
```

Puede añadirse a la estructura un bloque en finally que se ejecutará con independencia de si se produce excepción o no en el bloque try.