Estructuras de control de flujo

Los programas que se pueden realizar utilizando solamente variables y operadores son una simple sucesión lineal de instrucciones básicas.

Sin embargo, no se pueden realizar programas que muestren un mensaje si el valor de una variable es igual a un valor determinado y no muestren el mensaje en el resto de casos. Tampoco se puede repetir de forma eficiente una misma instrucción, como por ejemplo sumar un determinado valor a todos los elementos de un array.

Para realizar este tipo de programas son necesarias las **estructuras de control de flujo**, que son instrucciones del tipo *"si se cumple esta condición, hazlo; si no se cumple, haz esto otro"*.

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.

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(condicion) {

  ...
}
```

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.

Ejemplo:

```
var mostrarMensaje = true;

if(mostrarMensaje) {
  alert("Hola Mundo");
}
```

En el ejemplo anterior, el mensaje sí que se muestra al usuario ya que la variable `mostrarMensaje` tiene un valor de `true` y por tanto, el programa entra dentro del bloque de instrucciones del `if`.

El ejemplo se podría reescribir también como:

```
var mostrarMensaje = true;

if(mostrarMensaje == true) {
  alert("Hola Mundo");
}
```

En este caso, la condición es una comparación entre el valor de la variable `mostrarMensaje` y el valor `true`. Como los dos valores coinciden, la igualdad se cumple y por tanto la condición es cierta, su valor es `true` y se ejecutan las instrucciones contenidas en ese bloque del `if`.

La comparación del ejemplo anterior suele ser el origen de muchos errores de programación, al confundir los operadores `==` y `=`. Las comparaciones siempre se realizan con el operador `==`, ya que el operador `=` solamente asigna valores:

```
var mostrarMensaje = true;

// Se comparan los dos valores
if(mostrarMensaje == false) {
  ...
}

// Error - Se asigna el valor "false" a la variable
if(mostrarMensaje = false) {
  ...
}
```

La condición que controla el `if()` puede combinar los diferentes operadores lógicos y relacionales mostrados anteriormente:

```
var mostrado = false;

if(!mostrado) {
  alert("Es la primera vez que se muestra el mensaje");
}
```

Los operadores `AND` y `OR` permiten encadenar varias condiciones simples para construir condiciones complejas:

```
var mostrado = false;
var usuarioPermiteMensajes = true;

if(!mostrado && usuarioPermiteMensajes) {
  alert("Es la primera vez que se muestra el mensaje");
}
```

La condición anterior está formada por una operación `AND` sobre dos variables. A su vez, a la primera variable se le aplica el operador de negación antes de realizar la operación `AND`. De esta forma, como el valor de `mostrado` es `false`, el valor `!mostrado` sería `true`. Como la variable `usuarioPermiteMensajes` vale `true`, el resultado de `!mostrado && usuarioPermiteMensajes` sería igual a `true && true`, por lo que el resultado final de la condición del `if()` sería `true` y por tanto, se ejecutan las instrucciones que se encuentran dentro del bloque del `if()`.

Ejercicio

Completar las condiciones de los `if` del siguiente script para que los mensajes de los `alert()` se muestren siempre de forma correcta:

```
var numero1 = 5;
var numero2 = 8;

if(...) {
  alert("numero1 no es mayor que numero2");
}
if(...) {
  alert("numero2 es positivo");
}
if(...) {
  alert("numero1 es negativo o distinto de cero");
}
if(...) {
  alert("Incrementar en 1 unidad el valor de numero1 no lo hace mayor o igual que
numero2");
}
```

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 definición formal es la siguiente:

```
if(condicion) {
  ...
}
else {
  ...
}
```

Si la condición se cumple (es decir, si su valor es `true`) se ejecutan todas las instrucciones que se encuentran dentro del `if()`. Si la condición no se cumple (es decir, si su valor es `false`) se ejecutan todas las instrucciones contenidas en `else { }`. Ejemplo:

```
var edad = 18;

if(edad >= 18) {
  alert("Eres mayor de edad");
}
else {
  alert("Todavía eres menor de edad");
}
```

Si el valor de la variable `edad` es mayor o igual que el valor numérico `18`, la condición del `if()` se cumple y por tanto, se ejecutan sus instrucciones y se muestra el mensaje `"Eres mayor de edad"`. Sin embargo, cuando el valor de la variable `edad` no es igual o mayor que `18`, la condición del `if()` no se cumple, por lo que automáticamente se ejecutan todas las instrucciones del bloque `else { }`. En este caso, se mostraría el mensaje `"Todavía eres menor de edad"`.

El siguiente ejemplo compara variables de tipo cadena de texto:

```
var nombre = "";

if(nombre == "") {
  alert("Aún no nos has dicho tu nombre");
}
else {
  alert("Hemos guardado tu nombre");
}
```

La condición del `if()` anterior se construye mediante el operador `==`, que es el que se emplea para comparar dos valores (no confundir con el operador `=` que se utiliza para asignar valores). En el ejemplo anterior, si la cadena de texto almacenada en la variable `nombre` es vacía (es decir, es igual a `""`) se muestra el mensaje definido en el `if()`. En otro caso, se muestra el mensaje definido en el bloque `else { }`.

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

```
if(edad < 12) {
  alert("Todavía eres muy pequeño");
}
else if(edad < 19) {
  alert("Eres un adolescente");
}
else if(edad < 35) {
  alert("Aun sigues siendo joven");
}
else {
  alert("Piensa en cuidarte un poco más");
}
```
