

Javascript

Estructuras
Condicionales y
Repetitivas

GUIA DEL
ALUMNO

Contenidos

Estructuras Condicionales 3

 Estructuras Condicionales Simples 3

 Estructuras condicionales compuestas 5

 Estructuras Condicionales Anidadas 6

 Estructuras switch 11

Estructuras Repetitivas 13

 Acumuladores 13

 WHILE 14

 DO/WHILE 17

 FOR 18

Estructuras Condicionales

Estructuras Condicionales Simples

No todos los problemas pueden resolverse empleando estructuras secuenciales. Cuando hay que tomar una decisión aparecen las estructuras condicionales. En nuestra vida diaria se nos presentan situaciones donde debemos decidir.

- ¿Elijo la carrera A o la carrera B ?
- ¿Me pongo este pantalón ?
- ¿Entro al sitio A o al sitio B ?
- Para ir al trabajo, ¿elijo el camino A o el camino B ?
- Al cursar una carrera, ¿elijo el turno mañana, tarde o noche ?

Por supuesto que en un problema se combinan estructuras secuenciales y condicionales.

Cuando se presenta la elección tenemos la opción de realizar una actividad o no realizarla.

En una estructura CONDICIONAL SIMPLE por el camino del verdadero hay actividades y por el camino del falso no hay actividades. Por el camino del verdadero pueden existir varias operaciones, entradas y salidas, inclusive ya veremos que puede haber otras estructuras condicionales.

Ejemplo: Realizar la carga de una nota de un alumno. Mostrar un mensaje que aprobó si tiene una nota mayor o igual a 4:

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  var nombre;
  var nota;
  nombre=prompt('Ingrese nombre:','');
  nota=prompt('Ingrese su nota:','');
  if (nota>=4)
  {
    document.write(nombre+' esta aprobado con un '+nota);
  }
</script>
</body>
</html>
```

Aparece la instrucción if en el lenguaje JavaScript. La condición debe ir entre paréntesis. Si la condición se verifica verdadera se ejecuta todas las instrucciones que se encuentran encerradas entre las llaves de apertura y cerrado seguidas al if.

Para disponer condiciones en un if podemos utilizar alguno de los siguientes operadores relacionales:

- > mayor
- >= mayor o igual
- < menor
- <= menor o igual
- != distinto
- == igual

Siempre debemos tener en cuenta que en la condición del if deben intervenir una variable un operador relacional y otra variable o valor fijo. Otra cosa que hemos incorporado es el operador + para cadenas de caracteres:

```
document.write(nombre+' esta aprobado con un '+nota);
```

Con esto hacemos más corto la cantidad de líneas de nuestro programa, recordemos que veníamos haciéndolo de la siguiente forma:

```
document.write(nombre);  
document.write(' esta aprobado con un ');  
document.write(nota);
```

Estructuras condicionales compuestas

Cuando se presenta la elección tenemos la opción de realizar una actividad u otra. Es decir tenemos actividades por el verdadero y por el falso de la condición. Lo más importante que hay que tener en cuenta es que se realizan las actividades de la rama del verdadero o las del falso, NUNCA se realizan las actividades de las dos ramas.

En una estructura condicional compuesta tenemos entradas, salidas, operaciones, tanto por la rama del verdadero como por la rama del falso.

Ejemplo: Realizar un programa que lea dos números distintos y muestre el mayor de ellos:

```
<html>  
<head>  
</head>  
<body>  
<script type="text/javascript">  
  var num1,num2;  
  num1=prompt('Ingrese el primer número:');  
  num2=prompt('Ingrese el segundo número:');  
  num1=parseInt(num1);  
  num2=parseInt(num2);  
  if (num1>num2)  
  {  
    document.write('el mayor es '+num1);  
  }  
  else  
  {  
    document.write('el mayor es '+num2);  
  }  
</script>  
</body>  
</html>
```

La función prompt retorna un string por lo que debemos convertirlo a entero cuando queremos saber cual de los dos valores es mayor numéricamente. En el lenguaje JavaScript una variable puede ir cambiando el tipo de dato que almacena a lo largo de la

ejecución del programa.

Más adelante veremos qué sucede cuando preguntamos cuál de dos string es mayor.

Estamos en presencia de una ESTRUCTURA CONDICIONAL COMPUESTA ya que tenemos actividades por la rama del verdadero y del falso.

La estructura condicional compuesta tiene la siguiente codificación:

```
if (<condición>
{
    <Instruccion(es)>
}
else
{
    <Instruccion(es)>
}
```

Es igual que la estructura condicional simple salvo que aparece la palabra clave “else” y posteriormente un bloque { } con una o varias instrucciones.

Si la condición del if es verdadera se ejecuta el bloque que aparece después de la condición, en caso que la condición resulte falsa se ejecuta la instrucción o bloque de instrucciones que indicamos después del else.

Estructuras Condicionales Anidadas

Decimos que una estructura condicional es anidada cuando por la rama del verdadero o el falso de una estructura condicional hay otra estructura condicional.

Ejemplo: Confeccionar un programa que pida por teclado tres notas de un alumno, calcule el promedio e imprima alguno de estos mensajes:

Si el promedio es ≥ 7 mostrar "Promocionado".

Si el promedio es ≥ 4 y < 7 mostrar "Regular".

Si el promedio es < 4 mostrar "Reprobado".

Solución:

```
<html>
<head>
</head>
<body>
<script type="text/javascript"> var
    nota1,nota2,nota3;
    nota1=prompt('Ingrese 1ra. nota:');
    nota2=prompt('Ingrese 2da. nota:');
    nota3=prompt('Ingrese 3ra. nota:');
    //Convertimos los 3 string en enteros
    nota1=parseInt(nota1);
    nota2=parseInt(nota2);
    nota3=parseInt(nota3);
    var pro;
    pro=(nota1+nota2+nota3)/3;
    if (pro>=7)
    {
        document.write('promocionado');
    }
    else
    {
        if (pro>=4)
        {
            document.write('regular');
        }
        else
        {
            document.write('reprobado');
        }
    }
</script>
</body>
</html>
```

Analicemos el siguiente programa. Se ingresan tres string por teclado que representan las notas de un alumno, se transforman a variables enteras y se obtiene el promedio sumando los tres valores y dividiendo por 3 dicho resultado.

Primeramente preguntamos si el promedio es superior o igual a 7, en caso afirmativo por la rama del verdadero de la estructura condicional mostramos un mensaje que indique 'Promocionado' (con comillas indicamos un texto que debe imprimirse en pantalla).

En caso que la condición nos de falso, por la rama del falso aparece otra estructura condicional, porque todavía debemos averiguar si el promedio del alumno es superior o igual a cuatro o inferior a cuatro.

Los comentarios en JavaScript los hacemos disponiendo dos barras previas al comentario (los comentarios en tiempo de ejecución no son tenidos en cuenta y tienen por objetivos de documentar el programa para futuras modificaciones):

```
//Convertimos los 3 string en enteros
```

Si queremos disponer varias líneas de comentarios tenemos como alternativa:

```
/*  
linea de comentario 1.  
linea de comentario 2.  
etc.  
*/
```

Es decir encerramos el bloque con los caracteres `/* */`

El operador `&&`, traducido se lo lee como "Y". Se emplea cuando en una estructura condicional se disponen dos condiciones.

Cuando vinculamos dos o más condiciones con el operador `"&&"` las dos condiciones deben ser verdaderas para que el resultado de la condición compuesta de Verdadero y continúe por la rama del verdadero de la estructura condicional.

Recordemos que la condición debe ir entre paréntesis en forma obligatoria.

La utilización de operadores lógicos permiten en muchos casos, plantear algoritmos más cortos y comprensibles.

Ejemplo: Confeccionar un programa que lea por teclado tres números distintos y nos muestre el mayor de ellos.

```
<html>
<head>
</head>
<body>
<script type="text/javascript"> var
  num1,num2,num3; num1=prompt('Ingrese
  primer número:'); num2=prompt('Ingrese
  segundo número:'); num3=prompt('Ingrese
  tercer número:'); num1=parseInt(num1);
  num2=parseInt(num2);
  num3=parseInt(num3);
  if (num1>num2 && num1>num3)
  {
    document.write('el mayor es el '+num1);
  }
  else
  {
    if (num2>num3)
    {
      document.write('el mayor es el '+num2);
    }
    else
    {
      document.write('el mayor es el '+num3);
    }
  }
</script>
</body>
</html>
```

Podemos leerla de la siguiente forma:

Si el contenido de la variable num1 es mayor al contenido de la variable num2 Y si el contenido de la variable num1 es mayor al contenido de la variable num3 entonces la CONDICION COMPUESTA resulta Verdadera.

Si una de las condiciones simples da falso, la CONDICION COMPUESTA da Falso y continúa por la rama del falso.

Es decir que se mostrará el contenido de num1 si y sólo si num1>num2 y num1>num3.

En caso de ser Falsa la condición de la rama del falso, analizamos el contenido de num2 y num3 para ver cual tiene un valor mayor.

En esta segunda estructura condicional, al haber una condición simple, no se requieren operadores lógicos.

Traducido se lo lee como "O". Si la condición 1 es Verdadera o la condición 2 es Verdadera, luego ejecutar la rama del Verdadero.

Cuando vinculamos dos o más condiciones con el operador "O", con que una de las dos condiciones sea Verdadera alcanza para que el resultado de la condición compuesta sea Verdadero.

Ejemplo: Se carga una fecha (día, mes y año) por teclado. Mostrar un mensaje si corresponde al primer trimestre del año (enero, febrero o marzo).

Cargar por teclado el valor numérico del día, mes y año por separado.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  var dia,mes,año;
  dia=prompt('Ingrese día:');
  mes=prompt('Ingrese mes:');
  año=prompt('Ingrese año:');
  dia=parseInt(dia);
  mes=parseInt(mes);
  año=parseInt(año);
  if (mes==1 || mes==2 || mes==3)
  {
    document.write('corresponde al primer trimestre del año.');
```

La carga de una fecha se hace por partes, ingresamos las variables dia, mes y año

Estructuras switch

La instrucción switch es una alternativa para remplazar los if/else if.

De todos modos se puede aplicar en ciertas situaciones donde la condición se verifica si es igual a cierto valor. No podemos preguntar por mayor o menor.

Ejemplo: Confeccionar un programa que solicite que ingrese un valor entre 1 y 5. Luego mostrar en castellano el valor ingresado. Mostrar un mensaje de error en caso de haber ingresado un valor que no se encuentre en dicho rango.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  var valor;
  valor=prompt('Ingrese un valor comprendido entre 1 y 5:');
  //Convertimos a entero
  valor=parseInt(valor);
  switch (valor) {
    case 1: document.write('uno');
            break;
    case 2: document.write('dos');
            break;
    case 3: document.write('tres');
            break;
    case 4: document.write('cuatro');
            break;
    case 5: document.write('cinco');
            break;
    default:document.write('debe ingresar un valor comprendido entre 1
y 5.');
```

```
  }
</script>
</body>
</html>
```

Debemos tener en cuenta que la variable que analizamos debe ir después de la instrucción switch entre paréntesis. Cada valor que se analiza debe ir luego de la palabra clave 'case' y seguido a los dos puntos, las instrucciones a ejecutar, en caso de verificar dicho valor la variable que analiza el switch.

Es importante disponer la palabra clave 'break' al finalizar cada caso. Las instrucciones que hay después de la palabra clave 'default' se ejecutan en caso que la variable no se verifique en algún case. De todos modos el default es opcional en esta instrucción.

Plantaremos un segundo problema para ver que podemos utilizar variables de tipo cadena con la instrucción switch.

Ingresar por teclado el nombre de un color (rojo, verde o azul), luego pintar el fondo de la ventana con dicho color:

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  var col;
  col=prompt('Ingrese el color con que se quiere pintar el fondo de la
  ventana (rojo, verde, azul)', '');
  switch (col) {
    case 'rojo': document.bgColor='#ff0000';
                break;
    case 'verde': document.bgColor='#00ff00';
                break;
    case 'azul': document.bgColor='#0000ff';
                break;
  }
</script>
</body>
</html>
```

Cuando verificamos cadenas debemos encerrarlas entre comillas el valor a analizar:

```
case 'rojo': document.bgColor='#ff0000';
```

```
    break;
```

Para cambiar el color de fondo de la ventana debemos asignarle a la propiedad bgColor del objeto document el color a asignar (el color está formado por tres valores hexadecimales que representan la cantidad de rojo, verde y azul), en este caso al valor de rojo le asignamos ff (255 en decimal) es decir el valor máximo posible, luego 00 para verde y azul (podemos utilizar algún software de graficación para que nos genere los tres valores).

Estructuras Repetitivas

Acumuladores

Explicaremos el concepto de un acumulador con un ejemplo.

Problema: Desarrollar un programa que permita la carga de 5 valores por teclado y nos muestre posteriormente la suma.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  var x=1;
  var suma=0;
  var valor;
  while (x<=5)
  {
    valor=prompt('Ingrese valor:','');
    valor=parseInt(valor);
    suma=suma+valor;
    x=x+1;
  }
  document.write("La suma de los valores es "+suma+"<br>");
</script>
</body>
</html>
```

En este problema, a semejanza de los anteriores, llevamos un CONTADOR llamado x que nos sirve para contar las vueltas que debe repetir el while.

También aparece el concepto de ACUMULADOR (un acumulador es un tipo especial de variable que se incrementa o decrementa con valores variables durante la ejecución del programa).

Hemos dado el nombre de suma a nuestro acumulador. Cada ciclo que se repita la estructura repetitiva, la variable suma se incrementa con el contenido ingresado en la variable valor.

La prueba del diagrama se realiza dándole valores a las variables:

valor	suma	x
0	0	
(Antes de entrar a la estructura repetitiva estos son los valores).		
5	5	1
16	21	2
7	28	3
10	38	4
2	40	5

Este es un seguimiento del programa planteado. Los números que toma la variable valor dependerá de qué cifras cargue el operador durante la ejecución del programa.

Hay que tener en cuenta que cuando en la variable valor se carga el primer valor (en este ejemplo es el valor 5), al cargarse el segundo valor (16), el valor anterior 5 se pierde, por ello la necesidad de ir almacenando en la variable suma el valor acumulado de los valores ingresados.

WHILE

Hasta ahora hemos empleado estructuras SECUENCIALES y CONDICIONALES. Existe otro tipo de estructuras tan importantes como las anteriores que son las estructuras REPETITIVAS.

Una estructura repetitiva permite ejecutar una instrucción o un conjunto de instrucciones varias veces.

Una ejecución repetitiva de sentencias se caracteriza por:

- ✓ La o las sentencias que se repiten.
- ✓ El test o prueba de condición antes de cada repetición, que motivará que se repitan o no las sentencias.

Funcionamiento del while: En primer lugar se verifica la condición, si la misma resulta verdadera se ejecutan las operaciones que indicamos entre las llaves que le siguen al while.

En caso que la condición sea Falsa continúa con la instrucción siguiente al bloque de llaves.

El bloque se repite MIENTRAS la condición sea Verdadera.

Importante: Si la condición siempre retorna verdadero estamos en presencia de un ciclo repetitivo infinito. Dicha situación es un error de programación, nunca finalizará el programa.

Ejemplo: Realizar un programa que imprima en pantalla los números del 1 al 100.

Sin conocer las estructuras repetitivas podemos resolver el problema empleando una estructura secuencial. Inicializamos una variable con el valor 1, luego imprimimos la variable, incrementamos nuevamente la variable y así sucesivamente. Pero esta solución es muy larga.

La mejor forma de resolver este problema es emplear una estructura repetitiva:

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  var x;
  x=1;
  while (x<=100)
  {
    document.write(x);
    document.write('<br>');
    x=x+1;
  }
</script>
</body>
</html>
```

Para que se impriman los números, uno en cada línea, agregamos la marca HTML de
.

Es muy importante analizar este programa:

La primera operación inicializa la variable x en 1, seguidamente comienza la estructura repetitiva while y disponemos la siguiente condición (x <= 100), se lee MIENTRAS la variable x sea menor o igual a 100.

Al ejecutarse la condición, retorna VERDADERO, porque el contenido de x (1) es menor o igual a 100.

Al ser la condición verdadera se ejecuta el bloque de instrucciones que contiene la estructura while. El bloque de instrucciones contiene dos salidas al documento y una operación. Se imprime el contenido de x y seguidamente se incrementa la variable x en uno.

La operación $x = x + 1$ se lee como "en la variable x se guarda el contenido de x más 1". Es decir, si x contiene 1 luego de ejecutarse esta operación se almacenará en x un 2.

Al finalizar el bloque de instrucciones que contiene la estructura repetitiva, se verifica nuevamente la condición de la estructura repetitiva y se repite el proceso explicado anteriormente.

Mientras la condición retorne verdadero, se ejecuta el bloque de instrucciones; al retornar falso la verificación de la condición, se sale de la estructura repetitiva y continúa el algoritmo, en este caso, finaliza el programa.

Lo más difícil es la definición de la condición de la estructura while y qué bloque de instrucciones se va a repetir. Observar que si, por ejemplo, disponemos la condición $x \geq 100$ (si x es mayor o igual a 100) no provoca ningún error sintáctico pero estamos en presencia de un error lógico porque al evaluarse por primera vez la condición retorna falso y no se ejecuta el bloque de instrucciones que queríamos repetir 100 veces.

No existe una RECETA para definir una condición de una estructura repetitiva, sino que se logra con una práctica continua, solucionando problemas.

Una vez planteado el programa debemos verificar si el mismo es una solución válida al problema (en este caso se deben imprimir los números del 1 al 100 en la página), para ello podemos hacer un seguimiento del flujo del diagrama y los valores que toman las variables a lo largo de la ejecución:

X	Cuando x vale 101 la condición de la estructura repetitiva retorna falso, en este caso finaliza el diagrama.
1	
2	
3	
4	
.	
.	
100	
101	

La variable x recibe el nombre de CONTADOR. Un contador es un tipo especial de variable que se incrementa o decrementa con valores constantes durante la ejecución del programa. El contador x nos indica en cada momento la cantidad de valores impresos en la página.

Importante: Podemos observar que el bloque repetitivo puede no ejecutarse si la condición retorna falso la primera vez.

La variable x debe estar inicializada con algún valor antes que se ejecute la operación $x = x + 1$.

Probemos algunas modificaciones de este programa y veamos qué cambios se deberían hacer para:

- 1 - Imprimir los números del 1 al 500.
- 2 - Imprimir los números del 50 al 100.
- 3 - Imprimir los números del -50 al 0.
- 4 - Imprimir los números del 2 al 100 pero de 2 en 2 (2,4,6,8100).

DO/WHILE

La sentencia do/while es otra estructura repetitiva, la cual ejecuta al menos una vez su bloque repetitivo, a diferencia del while que puede no ejecutar el bloque.

Esta estructura repetitiva se utiliza cuando conocemos de antemano que por lo menos una vez se ejecutará el bloque repetitivo.

La condición de la estructura está abajo del bloque a repetir, a diferencia del while que está en la parte superior.

Finaliza la ejecución del bloque repetitivo cuando la condición retorna falso, es decir igual que el while.

Problema: Escribir un programa que solicite la carga de un número entre 0 y 999, y nos muestre un mensaje de cuántos dígitos tiene el mismo. Finalizar el programa cuando se cargue el valor 0.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  var valor;
  do {
    valor=prompt('Ingrese un valor entre 0 y 999:');
    valor=parseInt(valor);
    document.write('El valor '+valor+' tiene ');
    if (valor<10)
    {
      document.write("Tiene 1 digitos");
    }
    else
    {
      if (valor<100)
```

```
        {
            document.write("Tiene 2 digitos");
        }
        else
        {
            document.write("Tiene 3 digitos");
        }
    }
    document.write('<br>');
} while(valor!=0);
</script>
</body>
</html>
```

En este problema por lo menos se carga un valor. Si se carga un valor menor a 10 se trata de un número de una cifra, si es mayor a 10 pero menor a 100 se trata de un valor de dos dígitos, en caso contrario se trata de un valor de tres dígitos. Este bloque se repite mientras se ingresa en la variable 'valor' un número distinto a 0.

FOR

Cualquier problema que requiera una estructura repetitiva se puede resolver empleando la estructura while. Pero hay otra estructura repetitiva cuyo planteo es más sencillo en ciertas situaciones.

Esta estructura se emplea en aquellas situaciones en las cuales CONOCEMOS la cantidad de veces que queremos que se ejecute el bloque de instrucciones. Ejemplo: cargar 10 números, ingresar 5 notas de alumnos, etc. Conocemos de antemano la cantidad de veces que queremos que el bloque se repita.

Por último, hay que decir que la ejecución de la sentencia break dentro de cualquier parte del bucle provoca la salida inmediata del mismo.

Sintaxis:

```
for (<Iniciación> ; <Condición> ; <Incremento o Decremento>)
{
    <Instrucciones>
}
```

Esta estructura repetitiva tiene tres argumentos: variable de inicialización, condición y variable de incremento o decremento.

Funcionamiento:

- Primero se ejecuta por única vez el primer argumento .

Por lo general se inicializa una variable.

- El segundo paso es evaluar la (Condición), en caso de ser verdadera se ejecuta el bloque,

en caso contrario continúa el programa.

- El tercer paso es la ejecución de las instrucciones.
- El cuarto paso es ejecutar el tercer argumento (Incremento o Decremento).
- Luego se repiten sucesivamente del Segundo al Cuarto Paso.

Este tipo de estructura repetitiva se utiliza generalmente cuando sabemos la cantidad de veces que deseamos que se repita el bloque.

Ejemplo: Mostrar por pantalla los números del 1 al 10.

```
<html>
<head>
</head>
<body>

<script type="text/javascript">
  var f;
  for(f=1;f<=10;f++)
  {
    document.write(f+" ");
  }
</script>

</body>
</html>
```

Inicialmente f se la inicializa con 1. Como la condición se verifica como verdadera se ejecuta el bloque del for (en este caso mostramos el contenido de la variable f y un espacio en blanco). Luego de ejecutar el bloque pasa al tercer argumento del for (en este caso con el operador ++ se incrementa en uno el contenido de la variable f, existe otro operador -- que decrementa en uno una variable), hubiera sido lo mismo poner $f=f+1$ pero este otro operador matemático nos simplifica las cosas.

Importante: Tener en cuenta que no lleva punto y coma al final de los tres argumentos del for. El disponer un punto y coma provoca un error lógico y no sintáctico, por lo que el navegador no avisará.