

Toma de decisiones o bifurcación

Realiza una u otra acción dependiendo del valor de la variable que se evalúe, en esta categoría se encuentran las estructuras: IF y SWITCH.

Estructura IF

La estructura más sencilla es la estructura IF y se utiliza de la siguiente manera:

```
If (expresión) {  
    Acciones a realizar en caso positivo  
} else {  
    Acciones a realizar en caso negativo  
}
```

Ejemplo;

En el siguiente ejemplo se solicita al usuario una respuesta, mediante el uso de una ventana de confirmación.

Debe tener en cuenta que: if (conf) significa, si la variable conf posee el valor true (verdadero), entonces escriba en el documento con la orden document.write: “Presionó Aceptar”, de otro modo (else) escriba en el documento, document.write: “Presionó Cancelar”

Un detalle que debe tener en cuenta es que si dentro de la rama if posee mas de una orden, estas deben encerrarse entre llaves: { }, lo mismo sucede con el else, siendo de buena costumbre usar las llaves siempre tanto en if como en else.

Recuerde que el else es opcional.



```
<html>
<body>
<script type="text/javascript">
<!--
conf = confirm("Desea Seguir?");
document.write(conf+"<br>");
if (conf) {
document.write("Presionó Aceptar");
}
else {
document.write("Presionó Cancelar");
}
-->
</script>
</body>
</html>
```

Bucles

Se usan para realizar una tarea repetidamente, de acuerdo a una condición determinada.
Se utilizan: FOR – WHILE – DO WHILE.

Bucle FOR

Se usa para realizar tareas repetitivas un número determinado de veces:

Sintaxis:

for (desde (inicio);hasta (condición);cómo_se_actualiza (incremento)) { Tareas a realizar }

Ejemplo

```
<html>
<body>
<SCRIPT>
for (i=1;i<6;i++){
document.write("<H" + i + ">Encabezado "+ i+
"</H><br>");
}
</SCRIPT>
</body>
</html>
```



Bucle WHILE

Se usa para repetir una tarea un número indefinido de veces, MIENTRAS se cumpla una condición.

Sintaxis:

```
while (condición){  
tarea a ejecutar  
}
```

Bucle while: primero se evalúa la condición y luego realiza la tarea.

Bucle do..while

Este bucle se ejecuta por lo menos una vez

Sintaxis:

```
do {  
tareas  
} while (condición)
```

Funciones

¿Qué es una función?

Una función es pequeño programa (subprograma) que realiza una o varias tareas determinadas.

- Conj. de instrucciones englobadas en un mismo proceso: este proceso es una función y tiene un nombre.
- Pueden ser convocados por su nombre desde diferentes lugares. De modo que sirven para evitar la repetición del código

Sintaxis

Las funciones poseen la siguiente estructura:



```
Function nombre(parámetro_1, parámetro_2,
....,parámetro_n) {
Código de tareas que realizará la función
}
```

Se escribe la palabra function, luego el nombre de la función y entre paréntesis los parámetros, si no los tiene, igual se incorporan los paréntesis() .

```
function alarma(){
Window.alert("ERROR!!!!")
}
```

Formas de ejecutar una función:

solo necesita invocar su nombre entre tags de script de la siguiente manera:

1. Luego de la función invocando su nombre: alarma()
2. En otra parte de la página
3. Desde un evento

Realice los siguientes ejemplos:

```
<html>
<body>
  <script type="text/javascript">
    function alarma(){
      alert("ERRORRRR!") }
    alarma()
  </script>
</body>
</html>
```

Otra forma de llamar a la función:

```
<html><body>
<script type="text/javascript">
function alarma(){
alert("ERRORRRR!")
}
</script>
<h1>veamos que pasa</h1>
<script>
alarma()
</script>
</body></html>
```

