

Desarrollo Web En Entorno Cliente

TEMA 2

Fundamentos de programación en JavaScript



Índice

- ▶ **Estructuras Condicionales.**
- ▶ **Estructuras Repetitivas.**
- ▶ **Funciones.**

Estructuras condicionales

- ▶ Las **estructuras condicionales** seleccionan unas acciones u otras dependiendo del estado de las variables.
- ▶ Vamos a ver las estructuras **if-elseif-else** y **switch-case**
- ▶ La principal diferencia entre una y otra, es que la **estructura if** comprueba si una condición es true o false.
- ▶ Mientras que la **estructura switch** permite definir situaciones de igualdad con varios valores concretos.

- Permite decidir si ejecutar o no un fragmento de código en función del resultado de una condición.

```
if ( condición ) {  
    /*Conjunto de instrucciones a ejecutar  
    | si la condición es verdadera*/  
  
}else if( condicion ){  
    /*Conjunto de instrucciones a ejecutar si no se cumple  
    | la condicion anterior y pero si la actual*/  
  
...  
}else{  
    /*Conjunto de instrucciones a ejecutar  
    | si no se cumple ninguna condicion anterior*/  
}
```

- El bloque if else y else son opcionales:

```
var miEdad = prompt("Introduce tu edad: ");  
if (miEdad > 30)  
{  
    alert("Ya eres una persona adulta");  
}
```

```
var x = prompt("Introduce la raiz cuadrada de -1: ");  
if (x=="i"){  
    alert ("¡Muy bien! Conoces perfectamente los números imaginarios");  
}else{  
    alert ("¡Recuerda estudiar los números imaginarios!");  
}
```

- Las condiciones se van ejecutando en orden

```
var resolucion = prompt("Introduce la resolucion maxima: ");  
if ( resolucion < 400) {  
    alert("Resolución móvil");  
}else if(resolucion < 800) {  
    alert("Resolución tablet");  
}else if(resolucion < 1280){  
    alert("resolución portatil");  
}else{  
    alert("resolución monitor");  
}
```


- Permite realizar diferentes acciones según el valor concreto de una variable:

```
var dia=prompt("Introduce el dia de la semana");
switch(dia) {
    case 1:
        alert("Lunes");
        break;
    case 2:
        alert("Martes");
        break;
    ...
    default:
        alert("No es un dia de la semana");
}
```


Estructuras repetitivas

- El Bucle For permite ejecutar un bloque de código un número fijo y conocido de veces.

```
for (expresión inicial; condición; incremento)
{
    // Instrucciones a ejecutar dentro del bucle.
}
```

```
var nombre;
for (var veces=1;veces<=10;veces++)
{
    nombre=prompt("Introduce un nombre");
    alert("<h4>El nombre numero"+veces+"es "+nombre+"</h4>");
}
```

```
var numero=prompt("Introduce la longitud de la cuenta atras");  
for (var countdown=numero;countdown>=0;countdown--)  
{  
    alert("Cuenta atras "+countdown);  
}
```

```
var x = "", i;  
for (i=1; i<=6; i++) {  
    x = x + "<h" + i + ">Cabecera " + i + "</h" + i + ">";  
}  
document.write(x);
```

```
for(var i = 0; i<dias.length; i++)  
{  
    document.write(dias[i]+"<br>");  
}
```

- Existe una variante del bucle for para recorrer arrays y objetos más cómodamente llamada **for in**:

```
for(i in dias)  
{  
    document.write(dias[i]+"<br>");  
}
```


- ▶ Con el **bucle for**, podemos repetir una tarea un número fijo de veces.
- ▶ El **bucle while** permite crear bucles que se ejecutan cero o más veces de manera indefinida.

```
while( condicion )  
{  
    instrucciones  
}
```

```
while (i < 10) {  
    document.write("<br>El numero es " + i);  
    i++;  
}
```

```
var i=1;  
var respuesta=confirm("¿Desea salir? Intento"+ i);  
while (respuesta!=true) {  
    respuesta=confirm("¿Desea salir? Intento"+ i);  
    i++;  
}
```

- ▶ Existe una variante del bucle while que es equivalente y permite escribir de forma más sencilla ciertos códigos
- ▶ Es el **bucle do while** que se ejecuta una o más (al menos una) veces mientras que se cumpla la condición de forma indefinida.

```
do  
{  
    instrucciones  
}while( condicion);
```

```
var i=1;
var respuesta;
do
{
    respuesta=confirm("¿Desea salir? Intento"+ i);
    i++;
}while (respuesta!=true);
```

```
var color;
do
{
    color=prompt("indica un color distinto de blanco");
}while(color=="white" || color=="#FFF" || color=="#FFFFFF");
```


Funciones

- ▶ Las **funciones son una herramienta** que nos permite definir una serie de instrucciones como si fuera una única instrucción evitando así duplicar código.
- ▶ Utilizar funciones permite **modularizar el código** y decidir que bloques de instrucciones se ejecutarán en cada momento.

```
function nombreFuncion(parametro1,parametro2,...)
{
    ...
}
```

```
function Compara(num1, num2){  
    if(num1>num2){  
        alert("El mayor es: "+num1);  
    }else{  
        alert("El mayor es: "+num2);  
    }  
}  
Compara(4,7);|
```

```
function Compara(num1, num2){  
    var mayor;  
    if(num1>num2){  
        mayor=num1;  
    }else{  
        mayor=num2;  
    }  
    return mayor;  
}  
alert(Compara(4,7));
```


- ▶ El **ámbito de una variable**, es la zona del programa donde existe la variable y se puede operar con ella.
- ▶ En **JavaScript** hay dos tipos de ámbitos: global y local.
- ▶ **Variables locales:** Son variables que sólo existen dentro de la función que las crea. No se puede acceder a ella desde fuera de la función.
- ▶ **Variables globales:** Son variables que no se han definido dentro de ninguna función y por tanto están disponibles en cualquier parte de la aplicación, incluso dentro de una función

► Ejemplo variables locales:

```
function Mostrar() {  
    var mensaje = "hola";  
    alert(mensaje);  
}  
Mostrar();  
alert(mensaje);
```

- ▶ Ejemplo variables globales:

```
var mensaje="hola";  
function Mostrar(){  
    alert(mensaje);  
}  
Mostrar();  
alert(mensaje);
```

- ▶ No se recomienda el uso de variables globales.
- ▶ En su lugar se utilizaran variables locales y los parámetros de las funciones.

Fin del Tema 2

Bibliografía

- ▶ Gauchat, Juan Diego: **“El gran libro de HTML5, CSS3 y Javascript”**. Editorial Marcombo. 2012
- ▶ Vara, J.M. y otros: **“Desarrollo Web en Entorno Cliente. CFGS”**. Editorial Ra-Ma. 2012
- ▶ **“Programación en Javascript”**. Colección de artículos disponibles en la url. <http://www.desarrolloweb.com/manuales/> Última visita: Septiembre 2017.
- ▶ **W3SCHOOL “Manual de referencia y Tutoriales”**
<http://www.w3schools.com/> Última visita Septiembre 2017
- ▶ **Comesaña, J.L. : Técnico en Desarrollo de Aplicaciones Web.**
<http://www.sitiolibre.com/daw.php> Última visita Septiembre 2017