ESTRUCTURAS DEFINIDAS POR EL USUARIO EN JAVASCRIPT

Unidad didáctica 3 - Parte 3

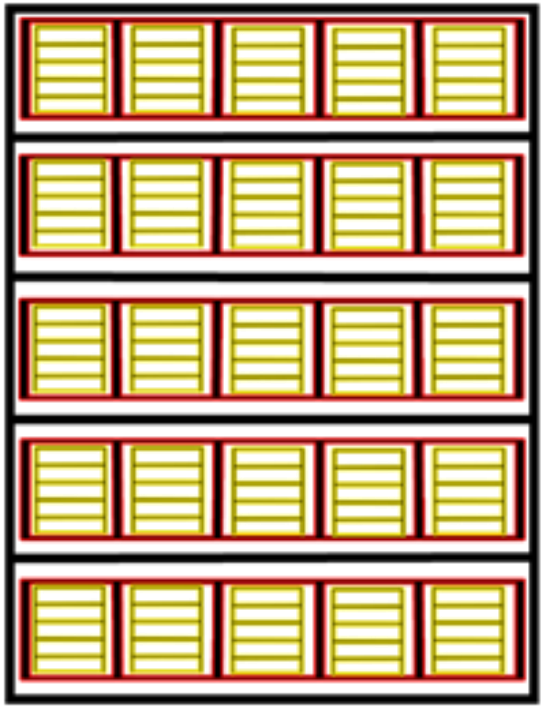
Arrays paralelos

- Dos o más arrays que utilizan el mismo índice para referirse a datos distintos pero que guardan una relación entre ellos.

```
var profesores = ["Cristina","Catalina","Vieites","Benjamin"];
var modulos=["Seguridad","Bases de Datos","Sistemas Informáticos","Redes"];
var alumnos=[24,17,28,26];
```

```
for (i=0;i<profesores.length;i++) {
    alert( profesores[indice]+" imparte el módulo de "+
           modulos[indice]+" que tiene "+
           alumnos[indice]+" alumnos en clase.” );
}
```

Arrays multidimensionables



- En JS los arrays son uni-dimensionales.
- Cada elemento puede ser otro array.

```
var datos = new Array();
datos[0] = new Array("Cristina","Seguridad",24);
datos[1] = new Array("Catalina","Bases de Datos",17);
datos[2] = new Array("Vieites","Sistemas Informáticos",28);
datos[3] = new Array("Benjamin","Redes",26);
```

```
var datos = [ ["Cristina","Seguridad",24],
              ["Catalina","Bases de Datos",17],
              ["Vieites","Sistemas Informáticos",28],
              ["Benjamin","Redes",26] ];
```

¿¿¿ datos[3][2] ???

Funciones

- Es la definición de un conjunto de acciones preprogramadas.
- Reutilizables.
- Específicas.
- Nombre adecuado (suelen incluir un verbo).
- No hay procedimientos.

CREACIÓN

```
function nombreFunción ( [parámetro1] .... [, parámetroN] ) {
   // conjunto de instrucciones
   [ return valor; ]
}
```

Funciones

LLAMADA

```
nombreFuncion([argumento1]…[,argumentoN]);                  // 1. sin valor de retorno
variable=nombreFuncion([argumento1]…[,argumentoN]);  // 2. con valor de retorno
If (nombreFuncion(…)) ….
```

Parámetros

- *Argumentos de la función.*
- *Son variables locales a la función que se inicializan automáticamente en el momento de llamar a la función con los valores que se le pasan en la llamada (~~var~~)*
- El nº y posición debe ser el mismo en creación y llamada.

Recomendaciones

- No reutilices un nombre de variable global como local en una función.
- No declares una variable global dentro de una función.

Funciones anidadas

- Recogen secuencias de instrucciones que necesitan ser llamadas desde múltiples sitios dentro de una función pero que sólo tienen significado dentro del contexto de esa función principal.
- Son funciones locales.

```
function  global() {
    function  local1() {
        // instrucciones
    }
    function  local2() {
        // instrucciones
    }
    // instrucciones
}
```

Funciones anidadas: ejemplo

```
function  totalVentaArticulo(articulo,cantidad) {
      function  compruebaPrecio(articulo) {
          // instrucciones
          return precio;
      }
      function  compruebaStock(articulo) {
          // instrucciones
          return stock;
      }
      function  compruebaIva(articulo) {
         // instrucciones
         return iva;
      }
      if (compruebaStock(articulo)>=cantidad) {
         return compruebaPrecio(articulo)*cantidad*(1+compruebaIva(articulo));
      } else {
         return 0;
      }
 }
//.............instrucciones del programa principal
if  (totalVentaArticulo("000123",20) >0) {
      alert("El precio total de la venta es" + totalVentaArticulo("000123",20)+"€");
} else {
     alert("No hay suficiente stock del articulo 000123");
}
//.....más instrucciones
```

Funciones predefinidas del lenguaje

- Todos los métodos de los objetos son funciones.
- Hay propiedades y métodos que se pueden utilizar a nivel global en cualquier parte del código de JavaScript.

Métodos globales de JavaScript	
Método	**Descripción**
decodeURI()	Decodifica los caracteres especiales de una URL excepto: , / ? : @ & = + $ #
decodeURIComponent()	Decodifica todos los caracteres especiales de una URL.
encodeURI()	Codifica los caracteres especiales de una URL excepto: , / ? : @ & = + $ #
encodeURIComponent()	Codifica todos los caracteres especiales de una URL.
eval()	Evalúa una cadena y la ejecuta si contiene código u operaciones.
isFinite()	Determina si un valor es un número finito válido.
isNaN()	Determina cuándo un valor no es un número.
Number()	Convierte el valor de un objeto a un número.
parseFloat()	Convierte una cadena a un número real.
parseInt()	Convierte una cadena a un entero.
String()	Convierte el valor de un objeto en string