

UD4: Estructuras definidas por el usuario

Arrays

Funciones

Creación de objetos

Array, vector o matriz

- Conjunto de elementos, homogénea o no, ordenados en filas.
- Permite acceder a los elementos de manera aleatoria (directa), usando un **índice** (indexados). Cada elemento se referencia por la posición que ocupa dentro del array. La posición se llama índice y es correlativa, empezando en el cero.
- Si el índice es una cadena hablamos de un **array asociativo**.
- **Matriz:** Array bidimensional (usamos varios índices para acceder a los elementos, tantos como dimensiones tenga el array).

Creación de Arrays indexados

```
/* Array vacío */  
var miArray= new Array();  
  
/* Array con número fijo de elementos */  
var miArray=new Array(40);  
  
/* Array inicializado al instanciar */  
var diasLaborables=new  
Array("Lunes","Martes","Miercoles","Jueves","Viernes");  
  
/* Definición literal, notación de [] */  
var diasFestivos=["Sabado","Domingo"];
```

Creación de Arrays asociativos

```
var traducciones={  
    "Lunes": "Monday",  
    "Martes": "Tuesday",  
    "Miércoles": "Wednesday",  
    "Jueves": "Thursday",  
    "Viernes": "Friday",  
    "Sabado": "Saturday",  
    "Domingo": "Sunday"};
```

Acceso al Array

Array indexado: de forma directa, a través del índice.

```
var lunes=diasLaborables[0];
```

Array asociativo: a través de sus propiedades.

```
var traduccionLunes=traducciones["Lunes"];
```

```
var traduccionLunes=traducciones.Lunes;
```

Acceso al Array

Matriz: a través de los índices de la matriz, tantos como dimensiones.

```
var traducciones=[[“lunes”,“Monday”],  
                  [“Martes”,“Tuesday”],  
                  [“Miércoles”,“Wednesday”],  
                  [“Jueves”,“Thursday”],  
                  [“Viernes”,“Friday”],  
                  [“Sabado”,“Saturday”],  
                  [“Domingo”,“Sunday”]];
```

```
var traduccionLunes= traducciones[0][1];
```

Recorriendo un Array: bucle for

Array indexado

Un único índice

```
for(i=0;i<diasLaborables.length;i++)  
    console.log(i,diasLaborables[i]);
```

Dos índices:

```
var semana=[["Lunes","Martes","Miércoles","Jueves","Viernes"],  
            ["sábado","domingo"]];  
for (let i=0; i< semana.length;i++){  
    for (let j=0; j<semana[i].length;j++){  
        console.log(semana[i][j]);  
    }  
}
```

Recorriendo un Array: bucle for..in

Ejercicio: comprueba el siguiente código y extrae tus conclusiones sobre el bucle for..in en un array indexado.

```
var a = [];  
a[5] = 3;  
for (var x in a) {  
    console.log(x);  
}
```

Ejercicio: comprueba el funcionamiento en un array asociativo:

```
var traducciones=[["lunes","Monday"],["Martes","Tuesday"], ["Miércoles","Wednesday"],  
["Jueves","Thursday"],["Viernes","Friday"],["Sabado","Saturday"],["Domingo","Sunday"]];  
  
for(var clave in traducciones)  
    console.log("Clave "+clave+" Valor:"+traducciones[clave]);
```


Recorriendo un Array: bucle for..of

Ejercicio: for...in vs for...of

```
var diasSemana=["lunes","martes","miércoles","jueves",  
               "viernes","sábado","domingo"];  
for (let dia in diasSemana) {  
  console.log(dia);  
}  
  
for (let dia of diasSemana) {  
  console.log(dia);  
}
```

Comprueba la diferencia en arrays simples y arrays asociativos.

Recorriendo un Array: método `forEach()` (ES6)

Sintaxis: *`forEach(funcion)`*

Permite recorrer los elementos de un array, ejecutando la función indicada una vez por cada elemento del mismo.

```
var diasSemana=["lunes","martes","miércoles","jueves","viernes","sábado","domingo"];
diasSemana.forEach(function (dia, contador) {
    console.log(dia, contador);
});
```

Recorriendo un Array: método `forEach()` (ES6)

Ventaja:

- Permite aplicar una función sobre cada elemento del array porque cada elemento del array se pasa sucesivamente como parámetro de la función asociada al método `forEach()`.

Pero hay que tener en cuenta lo siguiente:

- No se puede hacer `break` ni `continue`.
- Utilizar `return` dentro del bucle no evitará que se siga recorriendo hasta el final.

Recorriendo un Array

Ejercicios:

1.- A partir del siguiente array:

```
var numeros=[0,1,2,3,4,5,6,7,8,9,10,11,12];
```

- a) Muestra los números pares, utilizando for y forEach().
- b) Crea una función que sume todos los números del array, utilizando forEach().

Objeto Array: Propiedades y métodos

Propiedades

length	Longitud del array.
---------------	---------------------

Métodos

concat(array2)	Concatena con array2 y devuelve una copia de los arrays unidos.
join(separador)	Une todos los elementos del array en una cadena, separados por separador .
reverse()	Invierte el orden de los elementos del array.

Objeto Array: Métodos

reverse()	Invierte el orden de los elementos del array.
slice([inicio[,fin]])	Devuelve una copia de una parte del array empezando por inicio y acabando en fin .
sort()	Ordena los elementos de un array.
toString()	Convierte el array a cadena y devuelve el resultado. Con join se obtiene también un string y además se puede indicar el separador.
splice(i,n,e1,e2...)	Cambia el contenido de un array eliminando o añadiendo contenido. i indica a partir de donde se modifica el contenido. n indica el número de elementos a eliminar. En caso de ser 0, e1, e2... indica los elementos a añadir.

Objeto Array: Métodos

pop()	Elimina el último elemento del array y devuelve dicho elemento.
push(elemento)	Añade elementos al final del array y devuelve el nuevo tamaño.
shift()	Elimina el primer elemento del array y lo devuelve
unshift(elemento)	Añade un elemento al comienzo del array, devolviendo el nuevo tamaño.

Borrado de elementos en un Array

Podemos borrar un elemento:

- Asignando el valor del elemento a borrar a null o cadena vacía.
- Mediante el operador delete:

Sintaxis: delete array[i]

```
var diasClase = ["Lunes", "Martes", "Miércoles","Jueves","Viernes"];  
delete diasClase[0];
```

```
< [undefined × 1, "Martes", "Miercoles", "Jueves", "Viernes"]
```


Borrado de elementos en un Array

Al borrar un elemento con los métodos anteriores no se reduce la longitud del array .

Podemos eliminar un elemento o una secuencia de elementos ajustando el número de elementos usando el método `splice(pos,cantidad)`:

```
var diasClase = ["Lunes", "Martes", "Miércoles","Jueves","Viernes"];  
diasClase.splice(0,2);
```

```
["Miercoles", "Jueves", "Viernes"]
```

Objeto Array: Propiedades y métodos

Ejemplos:

- Concatenar dos arrays: **concat()**

```
var diasSemana=[];  
var diasClase = ["Lunes", "Martes", "Miércoles","Jueves","Viernes"];  
var diasFinDeSemana = ["Sábado","Domingo"];  
diasSemana=diasClase.concat(diasFinDeSemana);
```

- Unir los elementos de un array en una cadena: **join()**

```
var diasClase = ["Lunes", "Martes", "Miércoles","Jueves","Viernes"];  
var semana = diasClase.join(":");
```

Objeto Array: Propiedades y métodos

Ejemplos:

- Tomar elementos según su posición (inicial y final): **slice()**
var semana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
var diasClase = semana.slice(0, 5);
var diasFinDeSemana = semana.slice(-2);
- Eliminar elementos según su posición (inicio, número de elementos): **splice()**
var semana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
var diasFinDeSemana = semana.splice(5, 2);
var diasClase = semana.splice(-5);

Objeto Array: Propiedades y métodos

Ejemplos:

Partimos del array

```
meses["Enero", "Febrero","Marzo"];
```

a) Obtener una copia del array:

```
var copiaMeses = meses.slice();
```

b) Obtener la posición de un elemento del array:

```
var posicion= meses.valueOf("Febrero");
```

// Devuelve 1, porque Febrero está en la 2ª posición, primer índice es 0.

c) Añadir al final del array:

```
var nuevoTamano = meses.push("Abril");
```

//["Enero", "Febrero","Marzo","Abril"] y devuelve el tamaño del array

d) Eliminar un elemento al final del array:

```
var nuevoArray= meses.pop(); // borra Abril y devuelve el array ["Enero","Febrero","Marzo"];
```

Objeto Array: Propiedades y métodos

Ejemplos:

e) Añadir al principio del array:

```
var nuevoTamano = meses.unshift("Diciembre");// ["Diciembre", "Enero", "Febrero", "Marzo"];
```

f) Eliminar del principio del array:

```
var primero = meses.shift(); // ["Enero", "Febrero", "Marzo"];
```

g) Ordena según el orden lexicográfico: *sort()*

```
var nombres = ["Juan", "Ana", "Javier", "Ramón", "Isabel", "Luis", "Rosa"];  
var nombresOrdenado = nombres.sort ();
```

h) Invertir el orden de los elementos del array: *reverse()*

```
var nombres = ["Juan", "Ana", "Javier", "Ramón", "Isabel", "Luis", "Rosa"];  
var nombresReves = nombres.reverse();
```

Objeto Array: Propiedades y métodos

Ejercicios:

1.- Son equivalentes los siguientes códigos:

```
var semana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

```
var diasFinDeSemana= semana.splice(5, 2);  
var diasClase = semana.splice(-5);
```

```
var diasClase = semana.splice(-5);var  
diasFinDeSemana= semana.splice(5, 2);
```

2.- A partir de la variable

```
var semana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

Utilizar el método splice() para sustituir “Sábado” y “Domingo” por “No lectivo”, “No lectivo”.

Objeto Array: Propiedades y métodos

Ejercicios:

3.- Gestionar una lista de países haciendo uso de Arrays y funciones para cada uno de los siguientes apartados:

- a) *Mostrar el número de elementos del array.*
- b) *Mostrar todos los elementos del array.*
- c) *Mostrar los elementos del array en sentido inverso.*
- d) *Mostrar los elementos del array ordenados alfabéticamente (pero sin modificar el array).*
- e) *Añadir un elemento al principio del array.*
- f) *Añadir un elemento al final del array.*
- g) *Borrar un elemento al principio del array (y decir cuál se ha borrado).*
- h) *Borrar un elemento al final del array (y decir cuál se ha borrado).*
- i) *Mostrar el elemento que se encuentra en una posición que el usuario indica.*
- j) *Mostrar la posición en la que se encuentra un elemento que le indica el usuario.*
- k) *Mostrar los elementos que se encuentran en un intervalo que el usuario indica.*