ARRAYS

INTRODUCCIÓN

Una estructura típica en todos los lenguajes es el Array, que es como una variable donde podemos introducir varios valores, en lugar de solamente uno como ocurre con la variables normales.

Los arrays nos permiten guardar varias variables y acceder a ellas de manera independiente, es como tener una variable con distintos compartimentos donde podemos introducir datos distintos. Para ello utilizamos un índice que nos permite especificar el compartimiento o posición a la que nos estamos refiriendo.

Más sobre arrays de JavaScript en: http://www.desarrolloweb.com/articulos/630.php

Each

```
Array.each(function(elemento, indice)    {
        alert("Indice" + indice + " Elemento " + elemento);
    });
```

Se utiliza para recorrer arrays, por cada posición de nuestro array se llama a la función que le pasamos por parámetro a each. Dicha función recibe por parámetros el valor de la posición actual del array y su indice.

Se puede ver mas claro con el siguiente ejemplo:

```
    window.onload = function() {
        var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];

        nombres.each(function(elemento, indice)      {
             escribe("nombres[" + indice + "] = '" + elemento);
        });
    };

//nombres[0] = 'Juan'
//nombres[1] = 'Pedro'
//. . .
```

Filter

```
var nuevoArreglo = Vector.filter(function(elemento, indice) {
        return elemento != "algo"; //condición
    }, otrovector);
```

Con filter podemos filtrar un array mediante una condición evitando así aquellos elementos que no necesitamos. Devuelve un array con los elementos que cumplen con la condición. Para realizar el filtro podemos utilizar el valor de cada elemento, el indice o una combinación de ambos.

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];

        var algunos = nombres.filter(function(elemento, indice)   {
            //Utilizamos el valor de cada elemento para filtrar
            return elemento != "Pedro" && elemento != "Federico";
        });

escribe("algunos " + algunos);   //Algunos Juan,Martin,Alejandro
```

```
var algunos = nombres.filter(function(elemento, indice)    {
      //Utilizamos el indice
      return indice > 2;
});

escribe(algunos);   //Martin, Alejandro
```

map

```
      var nuevoVector = Vector.map(function(elemento, indice)
          {        return elemento + "algo"; //operación      });
```

map nos permite aplicar una operación a cada una de las posiciones de un vector, retornando un nuevo array con los cambios realizados.

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];
var procesados = nombres.map(function(elemento, indice) {
      return indice + ".-" + elemento;
});

escribe("procesados " + procesados); //0.-Juan, 1.-Pedro, 2.-Federico, ...
```

every

```
var resultado = Vector.every(function(elemento, indice)    {
             return elemento != "algo"; //condicion    });
```

every ejecuta la función pasada por parámetro mientras la condición se cumple. Si la condición se cumple para todos los elementos, every retorna true. Ahora, si existe un solo elemento que no cumple la condición, retorna false.

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];

var resultado = nombres.every(function(elemento, indice)   {
      escribe(elemento);              //Juan, Pedro y se sale
      return elemento != "Pedro";
});

escribe("Res " + resultado);          //Res false
```

some

```
var resultado = Vector.some(function(elemento, indice)    {
       return elemento != "algo"; //condicion
});
```

some es parecida a every. Recorre el vector hasta que encuentra un elemento que cumple la condición e inmediatamente devuelve true. Si no encuentra elementos que cumplan la condición, devuelve false.

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];

var resultado = nombres.some(function(elemento, indice)    {
      escribe(elemento);              //Juan
      return elemento != "Pedro";
});

escribe("Res " + resultado);          //Res true
```

indexOf

```
var resultado = Vector.indexOf(valor);
```

indexOf nos permite buscar un elemento dentro de un vector, retornando su indice en caso de encontrarlo o -1 en caso contrario (comienza en 0).

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];
var resultado = nombres.indexOf("Pedro");
escribe(resultado);      //1
```

erase (ant conocido como remove)

```
Vector = Vector.erase("valor");
```

Con erase borramos un elemento del array.

Contains

```
resultado = Vector.contains("valor", comienzo);
```

Con contains podemos buscar un elemento dentro de un vector, devuelve true en caso de encontrarlo o false en caso contrario. Como parámetro opcional podemos indicar que comience a buscar desde una posición en particular, si no se especifica una posición, comenzará desde 0.

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];
var resultado = nombres.contains("Pedro");
escribe(resultado);       //true

resultado = nombres.contains("Pedro", 2);
escribe(resultado);       //false
```

associate

```
var vectorAsociativo = vectorValores.associate(vectorLlaves);
```

Con associate podemos construir fácilmente un vector asociativo del tipo llave => valor a partir de dos vectores. Se puede ver claramente en el siguiente ejemplo:

```
var valores = ['valor 1', 'valor 2', 'valor 3', 'valor 4'];
var llaves = ['llave1', 'llave2', 'llave3', 'llave4'];

var vectorAsociativo = valores.associate(llaves);

escribe (vectorAsociativo['llave1']);         //valor 1
escribe (vectorAsociativo['llave2']);         //valor 2
```

extend

```
vector.extend(otroVector);
```

extend nos permite extender un vector con otro.

```
var nombres = new Array("Juan", "Pedro", "Federico", "Martin", "Alejandro");
nombres.extend(new Array("Martin", "Joaquin"));

escribe(nombres);  //Juan, Pedro, Federico, Martin, Alejandro, Martin, Joaquin
```

combine (ant conocido como merge)

```
vector.cobine(otroVector);
```

merge nos permite fusionar dos vectores, es decir, unir dos en uno dejando fuera los elementos repetidos.

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];
nombres.combine(new Array("Martin", "Joaquin"));

escribe(nombres);     //Juan, Pedro, Federico, Martin, Alejandro, Joaquin
```

include

```
      vector.include('elemento');
```

Con include podemos agregar un elemento siempre y cuando no sea un elemento ya existente.

getRandom

```
      vector.getRandom();
```

Con getRandom obtenemos un elemento aleatorio de nuestro vector.

getLast

```
      vector.getLast();
```

Con getLast obtenemos el ultimo elemento.

clean

Elimina aquellos elementos que están vacios (null) o están indefinidos (undefined).

Empty

Vacia un array.

```
var x = [1,2,3];
x.empty();
//lo mismo que
x = [];
```

flatten

Toma un vector de vectores y lo convierte a uno simple.

```
var myArray = [1,2,3,[4,5, [6,7]], [[[8]]]];
myArray.flatten(); //returns [1,2,3,4,5,6,7,8]
```

rgbToHex, hexToRgb

```
[99,100,101].rgbToHex() //returns "#636465"
['63','64','65'].hexToRgb() //returns "rgb(99,100,101)"
```

Links

Método especial que sirve para indicar el tipo de objeto. Sólo funciona si existen tipos diferentes.

```
var el = document.createElement('div');
var arr2 = [100, 'Hello', {foo: 'bar'}, el, false];
arr2.link({myNumber:    Number.type,    myElement:    Element.type,    myObject:
Object.type, myString: String.type, myBoolean: $defined});
//returns  {myNumber:  100,  myElement:  el,  myObject:  {foo:  'bar'},  myString:
'Hello', myBoolean: false}
```

$A (incluye ant conocido como copy)

Crea una copia de un Array. Útil para aplicar a:

- un array
- a objetos iterativos
    - una colección de DOM Node
    - el objeto de argumentos (arguments).

```
      var copia = $A(original);
```

Cuando nosotros intentamos copiar un vector de la siguiente manera:

```
      var vector = ["valor"];
      var nuevoVector = vector;
```

Lo que obtenemos en **nuevoVector** no es una copia sino una referencia, es decir, tanto nuevoVector como vector se refieren a los mismos elementos, por lo cual si modificamos nuevoVector también observaremos los mismos cambios en vector.

Copia de vector:

```
var nombres = ["Juan", "Pedro", "Federico", "Martin", "Alejandro"];

//Haciendo referencia
var nombres2 = nombres;
nombres2[0] = "Juancito";
escribe("Nombres " + nombres);  //Juancito, Pedro, Federico, Martin, Alejandro
escribe("Nombres2" + nombres2); //Juancito, Pedro, Federico, Martin, Alejandro

//Haciendo copia
var nombres3 = $A(nombres);
nombres3[0] = "Juancito";
escribe("Nombres3" + nombres3); //Juancito, Pedro, Federico, Martin, Alejandro
```

Copia de una colección DOM Node.

```
    copia = $A($$("span").get('text'));
    escribe (copia);
```

Con el objeto arguments:

```
function myFunction(){
  $A(arguments).each(function(argument){
    alert(argument);
  });
};
//Obendrá los argumentos pasados a la función myFunction.
```

Ejercicio

Objetivo:
Crear un programilla que ayude a apreder los días de la semana en otro idioma.

PARTE 1:
Crea un array con los nombres de los días de la semana. Crea un div por cada día con una clase compuesta por 2 nombres: “libre oculto”.

- Libre tendrá las propiedades que tu quieras.
- Oculto sólo tendrá como propiedad “visibility: hidden;”

De esta manera que NO se vean nada mas arrancar la página.

```
div id=”enero” class=”estilo_libre oculto” >Lunes</div>
```

Crea un input de texto.

Si el usuario escribe un día de forma correcta, este se enciende con la función ($ sirve para tomar la etiqueta html con id escrito entre paréntesis).

$(“dia_escrito”).removeClass(“oculto”)

Si no hay acierto se escribe (alert) error.

Si hay acierto, eliminas el día del array dias de la semana y lo pones en el array de aciertos.

Cuando estén todos acertados pinta un vector ordenado según fue acertando.

PARTE 2:
Crea otro array con los días de la semana en inglés
var ing = ["monday", "tuesday", "wednesday", "thursday", "friday", "saturday","sunday"];

Crea otro array asociativo del tipo traduce[“Monday”] = “lunes”

Modifica el ejercicio anterior de manera que cuando escriba “Monday” se encienda “lunes”.

Al completar la semana debe aparecer el orden de los días de la semana en inglés según los fui acertando.