ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Auditory exercises 6

Internet programming

Ivan Kitanovski

Bojan Ilijoski

Arrays in JavaScript

- JavaScirpt Arrays – JARRAY
- Serve to store multiple values in one variable.
- Creating arrays

```
var cars = ["Saab", "Volvo", "BMW"];
or
var cars = new Array("Saab", "Volvo", "BMW");
```

Arrays in JavaScript (2)

- Access to elements
  - □ `var cars = ["Saab", "Volvo", "BMW"];`
  - □ console.log(cars[0]);
- The elements can be objects
  - □ `myArray[0] = Date.now;`
  - □ `myArray[1] = myFunction;`
  - □ `myArray[2] = myCars;`
- Properties and methods
  - □ `var x = cars.length;    //number of elements`
  - □ `var y = cars.sort();    //sort the array`

Arrays in JavaScript (3)

- Adding elements
  - `var fruits = ["Banana", "Orange", "Apple", "Mango“да];`
  - `fruits.push("Lemon");`
- Array recognition
  - `var fruits = ["Banana", "Orange", "Apple", "Mango"];`
  - `fruits instanceof Array     // returns true`

Example 1

```
var days = ["Sunday", "Monday"];
alert(days.length); // => 2
var cars = [];
cars[1] = "Honda";
cars[3] = "Fiat";
alert(cars.length); // => 4
cars["six"] = "Volkswagen";
alert(cars.length); // => still returns 4
```

Example 1 (2)

```
for (var i = 0; i < cars.length; i++) {
    if (cars[i] === undefined) // skip
         undefined elements
         continue;
    alert(cars[i]); // => Ford, BMW
}
```

Example 2

```
var twoDim = [];
for (var row = 0; row < 5; row++) {
        var oneDim = [];
        for (var col = 0; col < 5; col++) {
                oneDim[col] = (row === col) ? 1 : 0; // 0 or 1 (diag)
        }
        twoDim[row] = oneDim;
}
console.log(twoDim[4][2]); // => 0
console.log(twoDim[3][3]); // => 1
console.log(twoDim); // =>[Array(5), Array(5), Array(5), Array(5), Array(5)]
console.log(twoDim[0]); //=> [1, 0, 0, 0, 0]
```

Example 3

```
var days = ["Sunday", "Monday", "Tuesday", "Wednesday"];
delete days[2]; // => delete the element at index 2
alert(days[2]); // => undefined
alert(days.length); // => still 4
alert(days); // =>["Sunday", "Monday", empty, "Wednesday"]
```

Functions with arrays

Method & Description
concat()Returns a new array comprised of this array joined with other array(s) and/or value(s).
every()Returns true if every element in this array satisfies the provided testing function.
filter()Creates a new array with all of the elements of this array for which the provided filtering function returns true.
forEach()Calls a function for each element in the array.
indexOf()Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found.
join()Joins all elements of an array into a string.
lastIndexOf()Returns the last (greatest) index of an element within the array equal to the specified value, or -1 if none is found.
map()Creates a new array with the results of calling a provided function on every element in this array.
pop()Removes the last element from an array and returns that element.
push()Adds one or more elements to the end of an array and returns the new length of the array.
reduce()Apply a function simultaneously against two values of the array (from left-to-right) as to reduce it to a single value.
reduceRight()Apply a function simultaneously against two values of the array (from right-to-left) as to reduce it to a single value.
reverse()Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first.
shift()Removes the first element from an array and returns that element.
slice()Extracts a section of an array and returns a new array.
some()Returns true if at least one element in this array satisfies the provided testing function.
toSource()Represents the source code of an object
sort()Sorts the elements of an array
splice()Adds and/or removes elements from an array.
toString()Returns a string representing the array and its elements.
unshift()Adds one or more elements to the front of an array and returns the new length of the array.

Problem 1

- Write a JavaScript function that receives an array and parameter n, as an argument and for result return first n elements of the array.

Input :

```
console.log(first([],3));
console.log(first([7, 9, 0, -2],3));
console.log(first([7, 9, 0, -2],6));
console.log(first([7, 9, 0, -2],-3));
```

Output :

```
[]
[7, 9, 0]
[7, 9, 0, -2]
[]
```

Solution

```
first =  function(array, n) {
    if (array == null)
      return void 0;
    if (n == null)
      return array[0];
    if (n < 0)
      return [];
    return array.slice(0, n);
  };

console.log(first([7, 9, 0, -2]));
console.log(first([],3));
console.log(first([7, 9, 0, -2],3));
console.log(first([7, 9, 0, -2],6));
console.log(first([7, 9, 0, -2],-3));
```

Problem 2

- Write a JavaScript code that will merge all elements in one string.

```
Input :
myColor = ["Red", "Green", "White", "Black"];
Output :
"Red,Green,White,Black"
"Red,Green,White,Black"
"Red+Green+White+Black"
```

Solution

```
myColor = ["Red", "Green", "White", "Black"];
console.log(myColor.toString());
console.log(myColor.join());
console.log(myColor.join('+'));
```

Problem 3

- Write a JavaScript code that receives integer as entrance and insert "-" between two pair numbers.

```
Input :
025468
Output :
0-254-6-8
```

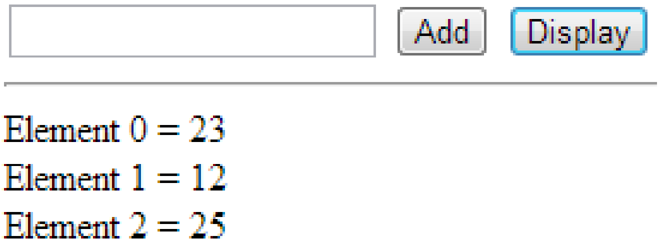
Solution

```
var num=window.prompt();
var str = num.toString();
var result = [str[0]];

for(var x=1; x<str.length; x++) {
  if((str[x-1]%2 === 0)&&(str[x]%2 === 0)) {
      result.push('-', str[x]);
  } else {
      result.push(str[x]);
  }
}
console.log(result.join(''));
```

Problem 4

- Make a web page where through text box and a button “Add” is entering elements in the array. With click on button “Display”, the array will be print on the page.



Element 0 = 23
Element 1 = 12
Element 2 = 25

Solution

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>JS</title>
<style>
body {padding-top:50px}
</style>
</head>
<body>
<input type="text" id="text1"></input>
<input type="button" id="button1" value="Add" onclick="
add_element_to_array();"></input>
<input type="button" id="button2" value="Display" oncli
ck="display_array();"></input>
<div id="Result"></div>
</body>
</html>
```

Solution (2)

```
var x = 0;
var array = Array();

function add_element_to_array()
{
 array[x] = document.getElementById("text1").value;
 alert("Element: " + array[x] + " Added at index " + x);
 x++;
 document.getElementById("text1").value = "";
}

function display_array()
{
   var e = "<hr/>";

   for (var y=0; y<array.length; y++)
   {
     e += "Element " + y + " = " + array[y] + "<br/>";
   }
   document.getElementById("Result").innerHTML = e;
}
```

Problem 5

- Write a function in JavaScript that take two arrays for arguments and for result return union of the two arrays.

Example:

```
console.log(union([1, 2, 3], [100, 2, 1, 10]));
[1, 2, 3, 10, 100]
```

Solution

```
function union(arra1, arra2) {

  if ((arra1 == null) || (arra2==null))
    return void 0;

  var obj = {};

  for (var i = arra1.length-1; i >= 0; -- i)
     obj[arra1[i]] = arra1[i];

  for (var i = arra2.length-1; i >= 0; -- i)
     obj[arra2[i]] = arra2[i];

  var res = [];

  for (var n in obj) {

    if (obj.hasOwnProperty(n))
      res.push(obj[n]);
  }

  return res;
}
console.log(union([1, 2, 3], [100, 2, 1, 10]));
```

Problem 6

- Write a function in JavaScript that for arguments receives two arrays and for result return the difference between the arrays (the elements that are different).

```
Examples:
console.log(difference([1, 2, 3], [100, 2, 1, 10]));
["3", "10", "100"]

console.log(difference([1, 2, 3, 4, 5], [1, [2], [3, [[4]]],[5,6]]));
["6"]

console.log(difference([1, 2, 3], [100, 2, 1, 10]));
["3", "10", "100"]
```

Solution

```
function difference(arr1,arr2) {
 var a1= flatten(arr1,true);
 var a2= flatten(arr2,true);
 var a=[], diff=[];
 for(var i=0;i<a1.length;i++)
            a[a1[i]]=false;
 for(i=0;i<a2.length;i++)
            if(a[a2[i]]===false) {
                        delete a[a2[i]];
            } else a[a2[i]]=true;
 for(var k in a)
  diff.push(k);
 return diff;
}
var flatten = function(a, shallow, r){
  if(!r){ r = [];}
  if (shallow) {
            return r.concat.apply(r,a);
  }
  for(i=0; i<a.length; i++){
            if(a[i].constructor == Array){
              flatten(a[i],shallow,r);
            }else{
              r.push(a[i]);
            }
  }
  return r;
};
console.log(difference([1, 2, 3], [100, 2, 1, 10]));
console.log(difference([1, 2, 3, 4, 5], [1, [2], [3, [[4]]],[5,6]]));
console.log(difference([1, 2, 3], [100, 2, 1, 10]));
```

Problem 7

- Write a function that will fill the array with value (integer and character) for given parameters.
- Example

```
console.log(num_string_range('a',  "z", 2));
["a", "c", "e", "g", "i", "k", "m", "o", "q", "s", "u", "w", "y"]
```

Solution

```
function num_string_range(start, end, step) {
 var range = [];
 if ((step === 0) || (typeof start == "undefined" || typeof end == "undefined") || (typeof start != typeof end))
  return false;
 if (end < start) {
  step = -step;
 }
 if (typeof start == "number") {
  while (step > 0 ? end >= start : end <= start) {
   range.push(start);
   start += step;
  }
 } else if (typeof start == "string") {
  if (start.length != 1 || end.length != 1) {
   throw TypeError("Strings with one character are supported.");
  }
  start = start.charCodeAt(0);
  end = end.charCodeAt(0);
  while (step > 0 ? end >= start : end <= start) {
   range.push(String.fromCharCode(start));
   start += step;
  }
 } else {
  throw TypeError("Only string and number are supported");
 }
 return range;
}

console.log(num_string_range('a', "z", 2)); //=> ["a", "c", "e", "g", "i", "k", "m", "o", "q", "s", "u", "w", "y"]
console.log(num_string_range("Z", "A", 2)); //=> ["Z", "X", "V", "T", "R", "P", "N", "L", "J", "H", "F", "D", "B"]
console.log(num_string_range(0, -5, 1)); //=> [0, -1, -2, -3, -4, -5]
console.log(num_string_range(0, 25, 5)); //=> [0, 5, 10, 15, 20, 25]
console.log(num_string_range(20, 5, 5)); //=> [20, 15, 10, 5]
```