

JAVASCRIPT – ARRAYS

by JS - Juha Söderqvist

Arrays

- special type of variable
- object

Creating an array

Different techniques can be used to create arrays.

- array literal
- array constructor

array literal

1. declare array variable

```
var colors;
```

2. square brackets []

assign values to the array with a square brackets []
and separated with a comma

NOTE: values can be different of data types

```
colors = ['white', 'black', 'red'];
```

3. final code (d8.1) filename: arrayliteral.js

```
var colors;  
colors = ['white', 'black', 'red'];
```

```
var ee = document.getElementById('colors');  
ee.textContent = colors[0];
```

array constructor

1. declare array variable

```
var colors = new Array
```

2. assign values to the array with () parentheses

and separated with a comma

NOTE: values can be different of data types

3. final code (d8.1) filename: arrayconstructor.js

```
var colors = new Array('white', 'black', 'red');
```

```
var ee = document.getElementById('colors');  
ee.textContent = colors[0];
```

Array properties

Array property length

```
var colors;  
colors = ['white', 'black', 'red'];
```

```
var howmany;  
howmany = colors.length;
```

Change values in an array

```
var colors;  
colors = ['white', 'black', 'red'];
```

```
// change value of array index 1  
colors[1] = 'green'
```

Output Array item to HTML

```
// get the element with an id of colors
var elementx = document.getElementById('colors');

// add to page
elementx.textContent = colors[2];
```

Functions

Function return single value

```
function calcarea(width, height) {  
  var area = width * height;  
  return area;  
}
```

```
// call the function  
var area1 = calcarea(2,3);  
var area2 = calcarea(4,4);
```

Function return multiple values

```
function getsize(width, height, depth) {  
  var area = width * height;  
  var volyme = width * height * depth;  
  var sizes = [area, volume];  
  return sizes;  
}
```

```
// get area from array id 0  
// get vol from array id 1  
var areaone = getsize(3, 2, 3)[0];  
var areaone = getsize(3, 2, 3)[1];
```

IIFE

Immediately invoked function expression "IIFE"

- Functions are not given a name
- executed once
- protect variables

```
var area= (function() {  
  var width = 3;  
  var height = 2;  
  return width * height;  
})();
```

IIFE (javascript design pattern)

```
(function(){ /* code */ }()); // Crockford recommends this one
```

```
(function(){ /* code */ })(); // But this one works just as well
```

```
(function () {
```

```
})();
```