Arrays

```
var arr = []; // Creates an array of no particular length
console.log(arr.length); // .length is one more than the last index
console.log(arr);

for (var value = 3, i = 0; value <= 66; value += 3, i++) {
   arr[i] = value;
}

console.log(arr.length);
console.log(arr);
```

```
0
[]
22
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66]
```

```
arr = new Array(21); // Creates an array of a specific length
console.log(arr.length);
console.log(arr);

for (var value = 3, i = 0; value <= 66; value += 3, i++) {
   arr[i] = value;
}

console.log(arr.length);
console.log(arr);
```

```
21
[undefined, undefined, undefined, undefined, undefined, undefined, undefined, undefined, undefined,
undefined, undefined, undefined, undefined, undefined, undefined, undefined, undefined, undefined, undefined,
undefined, undefined]
22
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66]
```

JavaScript Array Methods

```
var arr = [2, 6, 9];
var arr2 = arr.concat(12, [3, 5, 7]);
console.log(arr);
console.log(arr2);
```

```
[2, 6, 9]
[2, 6, 9, 12, 3, 5, 7]
```

```
var arr3 = [2, 6, 9];
var length = arr3.push(12, [3, 5, 7]);
console.log(arr3);
console.log(length);
```

```
[2, 6, 9, 12, [3, 5, 7]]
5
```

```
var arr4 = [2, 6, 9];
var length = arr4.unshift(12, [3, 5, 7]);
console.log(arr4);
console.log(length);
```

```
[12, [3, 5, 7], 2, 6, 9]
5
```

```
var arr5 = [2, 6, 9];
var value = arr5.pop();
console.log(arr5);
console.log(value);
```

```
[2, 6]
9
```

```
var arr6 = [2, 6, 9];
var value = arr6.shift();
console.log(arr6);
console.log(value);
```

```
[6, 9]
2
```

```
var arr7 = [2, 6, 9, 23];
var arr8 = arr7.slice(1, 3); // start index, end index (exclusive)
console.log(arr7);
console.log(arr8);
```

```
[2, 6, 9, 23]
[6, 9]
```

```
var arr9 = [2, 6, 9, 23];
var arr10 = arr9.splice(1); // start index
console.log(arr9);
console.log(arr10);
```

```
[2]
[6, 9, 23]
```

```
var arr11= [2, 6, 9, 23];
var arr12 = arr11.splice(1, 2, "a", "b", "c"); // start index, amount to delete, what to insert
console.log(arr11);
console.log(arr12);
```

```
[2, "a", "b", "c", 23]
[6, 9]
```

```
var arr13 = [2, 6, 9, 23];
var arr14 = arr13.reverse(); // Reference to the same array
console.log(arr13);
console.log(arr14);
```

```
[23, 9, 6, 2]
[23, 9, 6, 2]
```

```
var arr15 = [2, 16, 19, 43, 22];
var arr16 = arr15.sort(); // sorts in Unicode order, returns reference to same array
console.log(arr15);
console.log(arr16);
```

```
[16, 19, 2, 22, 43]
[16, 19, 2, 22, 43]
```

```
var arr17 = [2, 16, 19, 43, 22];
var arr18 = arr17.sort(function(a, b) {
   return a - b; // if a < b return -ve, if a > b return +ve, else return 0
});
```

```
console.log(arr17);
console.log(arr18);
```

```
[2, 16, 19, 22, 43]
[2, 16, 19, 22, 43]
```

```
var arr19 = [2, 16, 19, 43, 22];
var str = arr19.join();
console.log(arr19);
console.log(str);
```

```
[2, 16, 19, 43, 22]
2,16,19,43,22
```

```
var arr20 = [2, 16, 19, 43, 22];
var str2 = arr20.join(""); // empty string as the delimiter
console.log(arr20);
console.log(str2);
```

```
[2, 16, 19, 43, 22]
216194322
```

```
var arr21 = [2, 16, 19, 43, 22];
var str3 = arr20.toString();
console.log(arr21);
console.log(str3);
```

```
[2, 16, 19, 43, 22]
2,16,19,43,22
```

JavaScript Functions

Self-Invoking

```
(function() {
   console.log("Self-invoking function");
})();
```

```
Self-invoking function
```

Arguments

```
function func1(a, b, c) {
   console.log(a);
   console.log(b);
   console.log(c);
}

func1("value1", "value2"); // Note: 3rd argument missing
```

```
value1
value2
undefined
```

```
function func2(a) {
   console.log(a);
   console.log(arguments[1]);
   console.log(arguments[2]);
```

```
}

func2("value1", "value2", "value3"); // Note: 3 arguments
```

```
value1
value2
value3
```

Default Values

```
function func3(arg) {
   arg = arg || "This is the default";
   console.log(arg);
}

func3();
func3("jeff");
```

```
This is the default
jeff
```

Passing Objects

```
function func4(person) {
   console.log(person.name);
   console.log(person.age);
}

func4({ name: "Jeff", age: 34 });
```

```
Jeff
34
```

Return Values

```
function func5() {
}

console.log(typeof func5());

function func6() {
   return 1;
}

console.log(typeof func6());
```

```
undefined
number
```

Context Binding

```
function func7() {
   console.log(this.age);
}

func7();
var person = { name: "Jeff", age: 32 };
func7(person);
func7.call(person); // The context of the function is person
func7.apply(person); // The context of the function is person
```

```
undefined
undefined
```

```
32
32
```