



WEB TECHNOLOGIES

JavaScript – Arrays and Functions

Vinay Joshi

Department of
Computer Science and Engineering

- Arrays are lists of elements indexed by a numerical value starting with 0 to (length of the array – 1)
- Arrays can be created using
 - The new Array method
 - `let arr = new Array(100)` – creates an array of 100 elements
 - `let arr = new Array(10, 20)` – creates an array of 2 elements
 - Literal arrays using square brackets
 - `var alist = [1, "ii", "gamma", "4"];`

- Array length property can be modified at runtime
- Hence, the length property does not necessarily indicate the number of defined values in the array

```
const arr = [1, 2];  
console.log(arr);  
// [ 1, 2 ]
```

```
arr.length = 7;  
// set array length to 7  
console.log(arr);  
// [ 1, 2, <5 empty items> ]
```

```
console.log(arr[4]);  
//undefined
```

```
for (i in arr)  
    console.log(typeof i + i));  
// string 0  
// string 1
```

- Whenever you have a relatively complex piece of code that is likely to be reused, you have a candidate for a function.
- The general syntax for a function is:

```
function function_name([parameter [, ...]])  
{  
    statements  
    //optional return statement  
}
```

- The general syntax for calling a function is:

```
[retval =] function_name([argument [,...]])
```

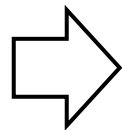
- Argument list and parameter list mismatch does not give errors.
- Parameter that is not passed a value in arguments list is treated as **undefined**
- To access additional arguments, use the **arguments** object or the **args** array to access the values passed.

```
function func() {  
    // access passed values  
    // using arguments object  
}
```

```
function func(...args) {  
    // access passed values  
    // using args array  
}
```

- Hoisting is JavaScript's default behavior of moving all variable and function declarations to the top of the current scope (to the top of the current <script> or the current function).
- Only declarations are hoisted not initializations
- Variables and constants declared with **let** or **const** are not hoisted!

```
num = 6;  
console.log(num);  
var num = 8;  
console.log(num);
```



```
var num;  
num = 6;  
console.log(num);  
num = 8;  
console.log(num);
```



THANK YOU

Vinay Joshi

Department of Computer Science and Engineering

vinayj@pes.edu

+91 80 2672 6622