



MERN – ES6 + React

Module 01

Outline

Module 08

- ▶ Understanding concepts of Functions
- ▶ Understanding concepts of Arrays
- ▶ Understanding concepts of Objects
- ▶ Understanding concepts of For Loops

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.

JavaScript Function Syntax

A JavaScript function is defined with the `function` keyword, followed by a **name**, followed by parentheses `()`.

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include parameter names separated by commas:

(parameter1, parameter2, ...)

The code to be executed, by the function, is placed inside curly brackets: `{ }`

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.

Function Return

When JavaScript reaches a `return` statement, the function will stop executing.

If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a **return value**. The return value is "returned" back to the "caller":

Example

Calculate the product of two numbers, and return the result:

```
let x = myFunction(4, 3); // Function is called, return value will end up in x

function myFunction(a, b) {
  return a * b;          // Function returns the product of a and b
}
```

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.

Arrow Functions

Arrow functions allows a short syntax for writing function expressions.

You don't need the `function` keyword, the `return` keyword, and the **curly brackets**.

Example

```
// ES5
var x = function(x, y) {
  return x * y;
}

// ES6
const x = (x, y) => x * y;
```

JavaScript Arrays

JavaScript basics

JavaScript Arrays

JavaScript arrays are written with square brackets.

Array items are separated by commas.

The following code declares (creates) an array called `cars`, containing three items (car names):

Example

```
const cars = ["Saab", "Volvo", "BMW"];
```

Array indexes are zero-based, which means the first item is [0], second is [1], and so on.

JavaScript Objects

JavaScript basics

JavaScript Objects

JavaScript objects are written with curly braces `{}`.

Object properties are written as name:value pairs, separated by commas.

Example

```
const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

The object (person) in the example above has 4 properties: firstName, lastName, age, and eyeColor.

JavaScript Loops

JavaScript basics

Different Kinds of Loops

JavaScript supports different kinds of loops:

- `for` - loops through a block of code a number of times
- `for/in` - loops through the properties of an object
- `for/of` - loops through the values of an iterable object
- `while` - loops through a block of code while a specified condition is true
- `do/while` - also loops through a block of code while a specified condition is true

JavaScript Loops

JavaScript basics

The For Loop

The `for` loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {  
  // code block to be executed  
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

Example

```
for (let i = 0; i < 5; i++) {  
  text += "The number is " + i + "<br>";  
}
```

JavaScript Loops

JavaScript basics

The For In Loop

The JavaScript `for in` statement loops through the properties of an Object:

Syntax

```
for (key in object) {  
  // code block to be executed  
}
```

Example

```
const person = {fname:"John", lname:"Doe", age:25};  
  
let text = "";  
for (let x in person) {  
  text += person[x];  
}
```

JavaScript Loops

JavaScript basics

The For Of Loop

The JavaScript `for of` statement loops through the values of an iterable object.

It lets you loop over iterable data structures such as Arrays, Strings, Maps, NodeLists, and more:

Syntax

```
for (variable of iterable) {  
  // code block to be executed  
}
```

Example

```
let language = "JavaScript";  
  
let text = "";  
for (let x of language) {  
  text += x;  
}
```

JavaScript Loops

JavaScript basics

The While Loop

The `while` loop loops through a block of code as long as a specified condition is true.

Syntax

```
while (condition) {  
    // code block to be executed  
}
```

Example

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

Summary

Module 08

- ▶ Understanding concepts of Functions
- ▶ Understanding concepts of Arrays
- ▶ Understanding concepts of Objects
- ▶ Understanding concepts of For Loops