
Lesson 14: Functions and Objects

 GENERAL ASSEMBLY

FRONT-END WEB DEVELOPMENT

Emerson Taymor
co-founder, philosophie

- **Review**
- **Functions**
 - Understand how functions take input & create output
 - Write a Function
 - Call a Function
- **JavaScript Objects**
 - Creating Objects and attributes.
 - JSON: Saving objects in strings.

REVIEW

Is there homework or topics you would like to review?

FUNCTIONS

FUNCTIONS

A function is a block of code that is executed when it is referred to.

Functions allow us to reuse snippets of code so we don't have to write them every time.

We can pass dynamic data to functions to make them more useful.

`$.html()`, `$.css()`, ATM withdraw

FUNCTIONS – SYNTAX

“*function*” Keyword. Like the “var” keyword.

The name of your function



```
function functionName(arg1, arg2) {  
    //Body of function  
}
```

FUNCTIONS – SYNTAX

Arguments let you pass data into the function

```
function functionName(arg1, arg2) {  
    //Body of function  
}
```

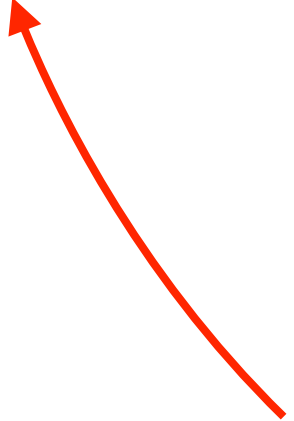


The functions executed code goes between the { } brackets. Much like an “if” statement.

FUNCTIONS – EXAMPLE

```
function helloWorld() {  
    console.log("Hello Functions");  
}
```

helloWorld(); //Prints "Hello Functions to the console.



The brackets execute the function.
Try calling the function without
them to see what happens.

FUNCTIONS – EXAMPLE

```
function addAndPrint(num1, num2) {  
    var sum = num1 + num2;  
    console.log(sum);  
}
```

addAndPrint(1, 2); // Result is 3

addAndPrint(8, 2); // Result is 10

CODE ALONG

Code_Along_Functions

RETURNING DATA FROM FUNCTIONS

What if we want to use the data the function creates?
The “return” method allows us to do that.

FUNCTIONS – EXAMPLE

```
function add(num1, num2) {  
    var sum = num1 + num2;  
    return sum;  
}
```

add(1, 2); // 3 is the result

var answer = **add**(20, 10);

ORGANIZING FUNCTIONS

Ideally, move all of your functions that are going to be reused in multiple places to a global.js file.

If you want to use functions and a document(ready), keep the functions outside of document(ready)

```
function callMe() {  
    // Do Something  
}
```

```
$(document).ready(function() {  
    callMe();  
});
```

LAB TIME

Available Exercises:

- Exercise_Return_Functions
- ATM_Functions

JAVASCRIPT

OBJECTS

OBJECTS

Technically every data type in JavaScript is an object

An object is a special type of data that has **properties** and **methods**

We create custom objects like we would for functions.

Address Book filled with contacts

OBJECTS – FORMAL DEFINITION

An “object” in computer science is a collection of data and functions that work with that data.

Objects allow us to organize similar data effectively

EXAMPLE OF AN “OBJECT”

Person

- Has a name, age, location
- Can speak, eat.

Lightbulb

- Number of watts, brand
- Can turn on and off

JAVASCRIPT OBJECTS – SYNTAX

Empty object `{}`

Person Object `var person = {
 age: 20,
 name: “Kevin Bacon”
};`

SYNTAX OF THE “KEY: VALUE” PAIRS

{key: “Value”}

SYNTAX OF THE “KEY: VALUE” PAIRS

```
{  
  age: 20,  
  name: “Kevin Bacon”  
  Profession: “Actor”  
}
```

SYNTAX OF THE “KEY: VALUE” PAIRS

Objects start and end with
{ } brackets



{name: “Kevin Bacon”}

SYNTAX OF THE “KEY: VALUE” PAIRS

The key is similar to
a variable name.




{name: “Beyonce Knowles”}

SYNTAX OF THE “KEY: VALUE” PAIRS

{name: “Beyonce Knowles”}


The value can be anything!
String, number, boolean..
even function or another
object!



SYNTAX OF THE “KEY: VALUE” PAIRS

{name: “Beyonce Knowles”, age: 20}

Multiple key/value pairs
are separated by commas,
like array values.



INDEX METHOD - ACCESSING DATA

Creating

```
var test = {a: "hi"};
```

Accessing

```
test["a"];  
// returns "hi"
```

Assigning

```
test["a"] = "bye";  
// test["a"] now  
stores "bye"
```

DOT METHOD – ACCESSING OBJECTS

Creating

```
var test = {a: “hi”};
```

Accessing

```
test.a;  
// returns “hi”
```

Assigning

```
test.a = “bye”;  
// test[“a”] now  
stores “bye”
```

CAR EXAMPLE

```
var car = {  
  make: “Ford”,  
  model: “Focus”,  
  year: 2013,  
  mileage: 89000  
}
```

CODE ALONG

Code_Along_Objects

JSON

- JavaScript Object Notation
- Way of representing objects we create as “Strings”
- Safe way of saving objects, and passing them between places

Things like Facebook and Twitter use JSON to pass data around.

- When you tweet, JSON is sent to the twitter servers

SYNTAX – JSON

Normal Object

{name: “John Smith”}

JSON

‘{"name":"John Smith"}’



It's a string!

JSON – TURING OBJECTS INTO JSON

JSON.stringify(obj);
// turns objects into JSON strings

JSON.parse(string);
// turns JSON strings into objects

LAB TIME

Exercise JSON

