Web Design and Programming (0107558)
Internet Programming (0107571)[1]

Interactivity with JavaScript

Dr. Naeem Odat

College of Engineering
Department of Computer and Communications Engineering

[1] Michael Mendez. *The Missing Link: An Introduction to Web Development and Programming*. Open SUNY Textbooks 2014.

What is JavaScript?

JavaScript

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

JavaScript

JavaScript is the most widely used scripting language on Earth. And it has the largest library ecosystem of any programming language. The best text editor is the one right for you.

Demo

- 0-numbergussinggame.html
- 1-htmlcssjavascript.html

JavaScript

<script> tag

- The script tags allow us to link to a script file or mark off a section of our code for our JavaScript to reside in.
- We can place script tags in any section of our HTML. Traditionally, JavaScript is placed in the head of the page, especially when the code consists of functions to be called or actions that are to occur on page load.

```
<script type="text/javascript" src="js/js.js"></script>
```

JavaScript

Including JavaScript

- Inline with document inside script tag
- As part of an event in an HTML tag.
- From a file within script start tag.

JavaScript

Variables

- A variable can be declared in 4 ways:
  - Automatically
  - Using *var*
  - Using *let*
  - Using *const*
- A variable can take on many different types (dynamically typed)
- A variable name needs to begin with a letter, $ or _.
- A variable is case sensitive.

JavaScript

Functions

```
function fname(){...}
```

```
var fname = function(){...}
```

- Each one is executed (invoked) as:

```
fname();
```

JavaScript

Functions with arguments

```
function fname(x,y){...}
```

- Can be called as (Legal):

```
fname(4,5);
fname(4,"a");
fname();
```

JavaScript

Scope

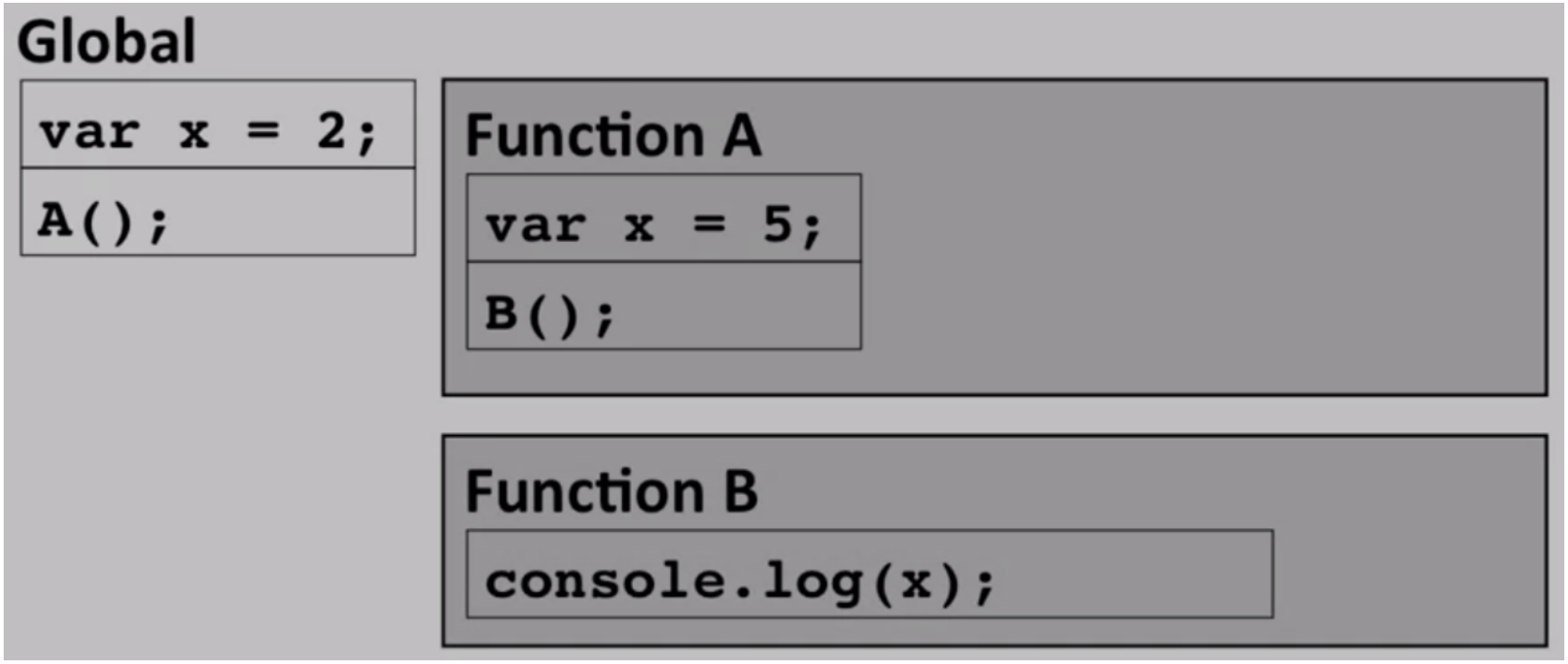
- Global. Variables and functions defined global are available everywhere.
- Function (lexical). Variables and functions defined in a function are available only within that function.
- Block. If defined using *let*.

Scope chain

Referenced variable will be searched for in its current scope first. If not found it will be searched for in its outer reference. If not found outer reference of its outer reference will be searched until the global scope. If not found it is declared undefined.

JavaScript

Scope chain



Variables, functions and scope Demo

2-scope.html

JavaScript

Global execution context

this

JavaScript

Quiz

1. JavaScript is always executed sequentially:
   - True
   - False
2. Given the following HTML code snippet:

   ```
   <script src="js/script.js"/>
   ```

   Is this HTML valid?
   - No, it's missing a closing 'script' tag. It must always be there.
   - Yes, the closing 'script' tag only needs to be there if javascript code is written in between the opening and closing 'script' tags.

JavaScript

Quiz

1. Identify valid JavaScript code snippet(s):
   - `<script src="js/script.js"/>`
   - ```
     var x = function () {
         console.log("Hello World");
         return;
     };
     ```
   - ```
     var x = function () {
         console.log("Hello World");
         return;
     };
     x();
     ```
   - ```
     var x = function () {
         console.log("Hello World");
         return;
     };
     x = "hello, I am a new value!";
     ```
   - ```
     var x = function ()
     [
         console.log("Hello World");
         return;
     ];
     ```

JavaScript

JavaScript Types

- Has 8 data types, 7 are primitive and 1 object type.

Object types

An Object is a collection of name/value pairs.
Person Object:

```
firstName: "Naeem",
lastName: "Odat",
info:{
    email:"naeemodat@ttu.edu.jo",
    office: "engineering college"
}
```

JavaScript

Primitive Types

- Primitive types represent a single value.
  1. Boolean. var x=true;
  2. Undefined. no value is set to it. Lack of definition (declared but not defined).
  3. Null. It is set to null, i.e., no value is assigned to it. Lack of value.
  4. Number. 64 bit double precision floating point number. (Accurate to 15 digits)
  5. Bigint. Store big integer. Too big for Number.
  6. String. Sequence of characters within single or double quotes.
  7. Symbol.

Demo

3-datatypes.html

JavaScript

Quiz

1. An object in Javascript is simply a collection of name/value pairs:
   - True
   - False
2. In Javascript, a primitive type can store only 1 name/value pair at a time:
   - True
   - False

JavaScript

Common Language Constructs

4-constructs.html

When you are unsure about anything if it is true or false, use Boolean wrapper to check on the console.

Handling default values

When a user provide an argument to the function the function will respond to it. What about when the user did not provide an argument to a function that takes argument from the user. The function should have a default value in the absence of argument.

Demo

5-default.html

JavaScript

Quiz

1. Strict equality operator (===) differs from regular equality operator (==) in that it checks if both values on its right and left are of the same type first. If they are not, it doesn't try to coerce them to be the same value and just returns false.
   - True
   - False
2. Given the following Javascript code:

```
var x = 10;
if ( (null) || (console.log("Hello")) || x > 5 ) {
    console.log("Hello");
}
```

   How many times will the word 'Hello' be printed to the console?
   - 1
   - 3
   - 2
   - 0

JavaScript

Creating objects

An object is a collection of name/value pairs.

Demo

6-createobjects.html

Quiz

Defining a variable as an Object Literal accomplishes pretty much the same thing as defining a variable equal to 'new Object()'. However, it's faster and easier to type up an object literal.

- True
- False

JavaScript

Functions

- Functions are objects.
- Functions can be used as a function factory.
- Functions can be passed as arguments to other functions.

Demo

7-functions.html

JavaScript

Quiz

What is the output for the following Javascript code:

```
function makeMultiplier(multiplier) {
    var myFunFunc = function (x) {
        return multiplier * x;
    };

    return myFunFunc;
}

var operation = makeMultiplier(10);
console.log(operation(10));
```

- 100
- 10
- 0
- 5

JavaScript

Passing variables by value or by reference

- Primitive variables are passed by value.
- Objects are passed by reference.
- Primitives:

```
var a = 8;
var b = a;
```

- Objects:

```
var a = {x: 7};
var b = a;
```

Demo

8-byvalueorreference.html

JavaScript

Quiz

1. Javascript doesn't allow anything to be passed by reference
   - False
   - True
2. What is the output of the following Javascript code:

```
var x = 5;
var y = x;
x = 10;
console.log(y);
```

   - 5
   - 10
   - undefined
   - null

Function Constructors

A function constructor is a blueprint (class) that can be used to generate new objects of the same type.

```
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}
```

this keyword in function

Refers to the global object.

Demo

9-functionconstructors.html

JavaScript

Object literals

```
var person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
```

this keyword in a method

In a method, this refers to the owner object.

Demo

10-objectliterals.html

JavaScript

Quiz

What is the output of the following Javascript code?

```
function Dog(name) {
    this.name = name;
}
Dog.prototype.bark = function () {
    console.log(this.name + " likes barking! Bark!");
}
var max = new Dog("Max", "Buddy");
max.bark();
```

- Max likes barking! Bark!
- Buddy likes barking! Bark!
- It will produce an error
- likes barking! Bark!

JavaScript

Arrays

Array is a collection of data.

Demo

11-arrays.html

JavaScript

Quiz

What is the output of the following Javascript code:

```
var counter = 0;
var myArray = ["Naeem", 2, {handle: "odatnaeem"}];
for (var i = 0; i < myArray.length; i++) {
    counter++;
}
console.log(counter);
```

- 3
- 0
- 4
- Nothing, there will be an error.

JavaScript

Clousers

A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a closure gives you access to an outer function's scope from an inner function.

Demo

12-clousers.html

JavaScript

DOM Manipulation

- Document object represents the DOM (i.e., the whole html page).
- (document.getElementById("title");) can be used on the console to display ($<$ *h*1 id="title"$>$ *DOM Manipulation* $<$ /*h*1 $>$).
- (document.querySelector("#title").textContent) can be used to read or write the content of the element with title id.

Demo

13-dommanipulation.html

JavaScript

Handling Events

- Events occur when the user or the browser manipulates a page.
- Events are a part of the DOM and every HTML element contains a set of events which can trigger JavaScript Code.
- Examples of events:
  - onclick: Triggers on a mouse click.
  - onchange: Triggers when an element changes.
  - onkeypress: Triggers when a key is pressed and released.

Demo

14-handlingevents.html

JavaScript

Handling Events

JavaScript inclusion can be added in the head tag when using (addEventListener) in the code.

Demo

15-handlingeventslistener.html

JavaScript

Quiz

- Given the following HTML code snippet:

  ```
  <input id="name" type="text" value="Hi">
  ```

  and the following Javascript snippets:
  Snippet 1:

  ```
  console.log(document.getElementById("name").value
   + " Coursera!");
  ```

  Snippet 2:

  ```
  console.log(document.querySelector("#name").value
  + " Coursera!");
  ```

  What will be the difference in the output between the 2 Javascript code snippets?

  - No difference
  - First one will not display the word 'Hi'
  - Second one will throw a Javascript error.
  - First one will throw a Javascript error.

JavaScript

Quiz

- What will the following code snippet accomplish?

```
...
<head>
  <script>
    document.addEventListener("DOMContentLoaded", function(event) {
      console.log("Yey!");
    });
  </script>
</head>
<body>
  ...
  <h1>Hello!</h1>
</body>
</html>
```

  - It will output Yey! when the web page is loaded, but before external resources like images are downloaded to the user's machine.
  - It will output Yey! when the user hits the Reload browser button.
  - It will never output Yey!
  - It will output Yey! concurrently as the browser displays the 'h1' tag.