JavaScript

25th August 2012

Angad Singh

```
function foo() {
   alert(“Hello World!”)
}
```

JavaScript

The world's most misunderstood language

http://javascript.crockford.com/javascript.html

No it has nothing to do with Java

Except some syntactical similarities

It is a scripting language – it is not compiled, but interpreted and executed on-the-fly.

The Web Browser: Execution Sequence

- Downloads the HTML document
- Starts Parsing the document

```
<html>
<head>
    <script> … </script>
    <style> … </style>
</head>
<body> … </body>
</html>
```

The Web Browser: Execution Sequence

- DOM is rendered, CSS is rendered in parallel
- Javascript is executed as it is seen. The browser may stop parsing the HTML when it is executing the Javascript.
- Resources such as images, embeds and iframes are requested as and when the parser encounters them. They are requested asynchronously.

The Web Browser

V8

SpiderMonkey

Chakra

Why do we need JavaScript?

You need to *dynamically* add or change elements on your page

All this happens on the *client-side*

Though, there is server-side JS, but that's out of scope here.

DOM – Document Object Model

Provides an object-oriented programming interface between HTML/CSS and JavaScript

Lets Dive!

In-line JavaScript

```
<script type = ‘text/javascript’>
alert(‘Hello World!’);
</script>
```

External JavaScript

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

Some basics

- **JavaScript is case-sensitive**

```
getElementById != getElementByid
```

- **Semicolons in JavaScript are optional**

(but, semicolons is my hobby)

```
var a = 1
var b = 2
```

- **Blocks do not have scope – only functions have scope**

Basics

- Variables

```
var a = 1;
var me = “angad”;
```

Say no to Global Variables! They are shared among all code in your JS application!

Always declare variables with `var` Any variable not declared with var, ends up in your “nearest” (mostly, global) namespace.

Basics

- Operators - +, -, *, / and %
- Assignment =, +=, -= etc.
- Increment and Decrement ++, --
- String Concatenation “hello” + “world”
- Comparisons <, >, <= and >=
- == performs type coercion with different types
- Use === to prevent type coercion

Control Structures

```
if (a === 1) {
    a++;
}

while (true) { }

for(var i = 0; i < 5; i++)

switch(action) {
    case ‘draw’: drawit();
                 break;
    case ‘eat’ : eatit();
                 break;
    default    : donothing();
}
```

Arrays

```
var a = new Array();
a[0] = “car”;
a[1] = “bike”;
var a = [“car”, “bike”];
```

Quick Tip: Length of the array is one more than the highest index.

```
a[100] = “truck”;
a.length = 101;
```

Objects

JS Objects are collection of name-value pairs.

```
var obj = new Object();
var obj = {};
//Similar

obj.name = “Angad”;
```

Is similar to

```
obj[“name”] = “Angad”;
```

Functions

```
function sayHello() {
   alert(“Hello!”);
}
```

Inner Functions:

```
function a() {
   function b(){
      return 1;
   }
   return b();
}
```

jQuery

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

“jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.”

Basics

```
$(document).ready(function(){
  // Your code here
});
```

$ - alias for for the jQuery “class”

$() – constructs a new jQuery object

DOM Manipulations

JavaScript

- `document.getElementById();`
- `document.getElementByClassName();`
- `document.getElementByTagName();`

jQuery

- `$('#id');`
- `$('.class');`
- `$('tagname');`
- `$('tagname#id.class');`

Events

JavaScript

- `onclick`
- `onfocus`
- `onkeydown`
- `onmouseover`
- …

jQuery

- `$(elem).click();`
- `$(elem).focus();`
- …

Animation

```
jQuery(element).fadeIn().fadeOut().show().
hide().slideDown().slideUp().toggle();
```

AJAX

```
$.ajax();
$.get();
$.post();
```

//Don’t do it by yourself!

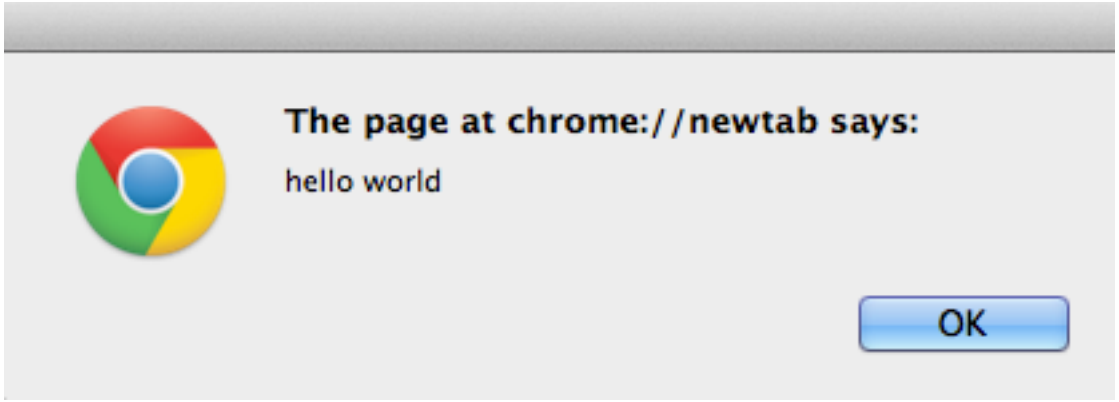
http://api.jquery.com/jQuery.ajax/

Hands On!

Chrome JavaScript Console: Alt + Command + J

Firebug in Firefox

alert(“hello world”);



READ THIS!

- http://net.tutsplus.com/tutorials/javascript-ajax/principles-of-maintainable-javascript/
- http://addyosmani.com/resources/essentialjsdesignpatterns/book/
- http://net.tutsplus.com/tutorials/javascript-ajax/digging-into-design-patterns-in-javascript/
- http://net.tutsplus.com/tutorials/javascript-ajax/the-essentials-of-writing-high-quality-javascript/
- https://developer.mozilla.org/en/JavaScript/Introduction_to_Object-Oriented_JavaScript
- https://developer.mozilla.org/en-US/docs/JavaScript/A_re-introduction_to_JavaScript

Some general rules

- Document your code!
- Use Version Control (Git!)
- Re-use code!