

JavaScript: JS30 I

Week 1: An Introduction to JavaScript

Introductions

- A little about your instructor
 - <http://blog.vicmetcalfe.com/>
- A little about you
 - Experience? Expectations?

Fuzzy Syllabus

- Intro and Basics
- Closures and Scope
- The Document Object Model (DOM)
- jQuery
- Object Oriented JavaScript
- Best Practices
- node.js? ajax? Windows 8?

What is JavaScript?

- JavaScript is a scripting language built into web browsers
- It is not Java - it is ECMAScript
- It powers web apps, mobile apps, server apps and desktop apps
- It is asynchronous, fast and powerful as hell
- JavaScript is fun!

No Server Required

- JavaScript can run on a server but most commonly runs on the client or browser
- Consoles in most browsers:
 - Chrome JavaScript Console
 - Firebug for Firefox
 - Safari Error Console
 - Internet Explorer Developer Tools

I'll be Using Chrome

- Use any tool you like
- I can help you best if you use Chrome too

Great JavaScript Resources

- W3 Schools: <http://www.w3schools.com/js/>
- YUI Theater: <http://yuilibrary.com/theater/>
- O'Reilly Books:
 - JavaScript: The Definitive Guide
 - JavaScript Patterns

Getting to the Console

- In Chrome right click and choose “Inspect Element”, then go to the Console tab.
- Type:
`document.write("Hello World");`
- Then press enter

What just Happened?

- In web browsers, document is an object which represents the current HTML document.
- It has a method called write which writes content to the HTML document.
- We invoked that method with the text we wanted to write as a parameter.
- The method returned the value 'undefined'.

How to break it

- Try these variations:
 - `Document.write("Hello World");`
 - `document.write('Hello World');`
 - `document.write("Hello World")`
 - `{document.write("Hello World");}`

Variables

- Variables hold values:
 - > `greeting = "Hello World";`
`"Hello World"`
 - > `document.write(greeting);`
`undefined`
- This also sends “Hello World” to the browser.
- Variable assignments return the assignment value.

Operators

- You can perform operations on variables:
 - > apples = "Apples";
"Apples"
 - > bananas = "Bananas";
"Bananas"
 - > ilike = apples + " & " + bananas;
"Apples & Bananas"
- What other operations do you think you could apply to variables?

Types

- Variables have types:

String:

```
> one = "1";  
"1"  
> two = "2";  
"2"  
> one + two  
"12"
```

Number:

```
> one = 1;  
1  
> two = 2;  
2  
> one + two  
3
```

Boolean:

```
> yup = true;  
true  
> nope = false;  
false  
> 5 < 2;  
false  
> !nope  
true
```

Arrays

- Array literals:

```
> fibonacci = [0,1,1,2,3,5,8,13];  
[0, 1, 1, 2, 3, 5, 8, 13]  
> fibonacci.push(21);  
9  
> fibonacci  
[0, 1, 1, 2, 3, 5, 8, 13, 21]  
> fibonacci.length;  
9  
> fibonacci[2];  
1  
> fibonacci[6];  
8
```

Objects

- Object Literals:

- > `Rectangle = { width: 10, height: 5 };`

- ▶ Object

- > `rectangle.height`
5

- > `rectangle.length`
undefined

More Types

- Functions
- Undefined

More on these later.

if conditional

```
> age = 13;  
13  
> if (age >= 16) console.log("Old enough to vote!");  
undefined  
> age = 25;  
25  
> if (age >= 16) console.log("Old enough to vote!");  
Old enough to vote!  
undefined
```

if conditional

```
> age = 15;  
15  
> if ((age > 12) && (age < 20)) console.log("Teen");  
Teen  
undefined  
> if ((age > 12) && (age < 20)) {  
    console.log("Teen");  
}  
Teen  
undefined
```

if conditional

```
> age = 15;  
15  
> if (age >= 16) {  
    console.log("Old enough to drive");  
} else {  
    console.log("Old enough in " +  
        (16 - age) + " year(s)");  
}  
Old enough in 1 year(s)  
undefined
```

JSFIDDLE

- We're outgrowing the JavaScript console!
- JSFIDDLE is another great way to play with JavaScript.
- Check out the next demo there:
<http://jsfiddle.net/zymsys/J88N5/>

Loops: for

- For loops are the most common type of loop in JavaScript:

```
for (i = 1; i <=10; i++) {  
    document.write("<div>" + i + "</div>");  
}
```

<http://jsfiddle.net/zymsys/awXGn/>

Loops: while

- While loops run until a condition is met

```
i = 1;
while (i <=10) {
    document.write("<div>" + i + "</div>");
    i += 1;
}
```

<http://jsfiddle.net/zymsys/AShuA/>

Loops: break & continue

- Break jumps out of a loop
- Continue jumps back to the start of the loop
- <http://jsfiddle.net/zymSYS/8uHr6/>

Loops: for .. in

- Loops over the items in an array or object
- <http://jsfiddle.net/zymsys/s8ABu/>

Functions

- Functions group code into re-usable chunks
- Can take parameters
- Can return values
- <http://jsfiddle.net/zymSYS/heKGP/>
- <http://jsfiddle.net/zymSYS/DwMBE/>

Exception Handling

- Instead of returning error codes you can “throw” errors up the call stack to error handlers
- <http://jsfiddle.net/zymsys/xskxy/>

Next Week...

- Advanced Functions and Closures
- Variable Scope
- The Document Object Model
- jQuery