

Website Design 1

Professor Danne Woo

webdesign.dannewoo.com

ARTS 214 – 02

Spring 2016

Thursday 2:00 pm – 5:50 pm

I-Building 213

What is JavaScript?

- Object-Orientated Programming Language
- Can manipulate DOM elements
- Works on all modern browsers and devices
- Not related to Java

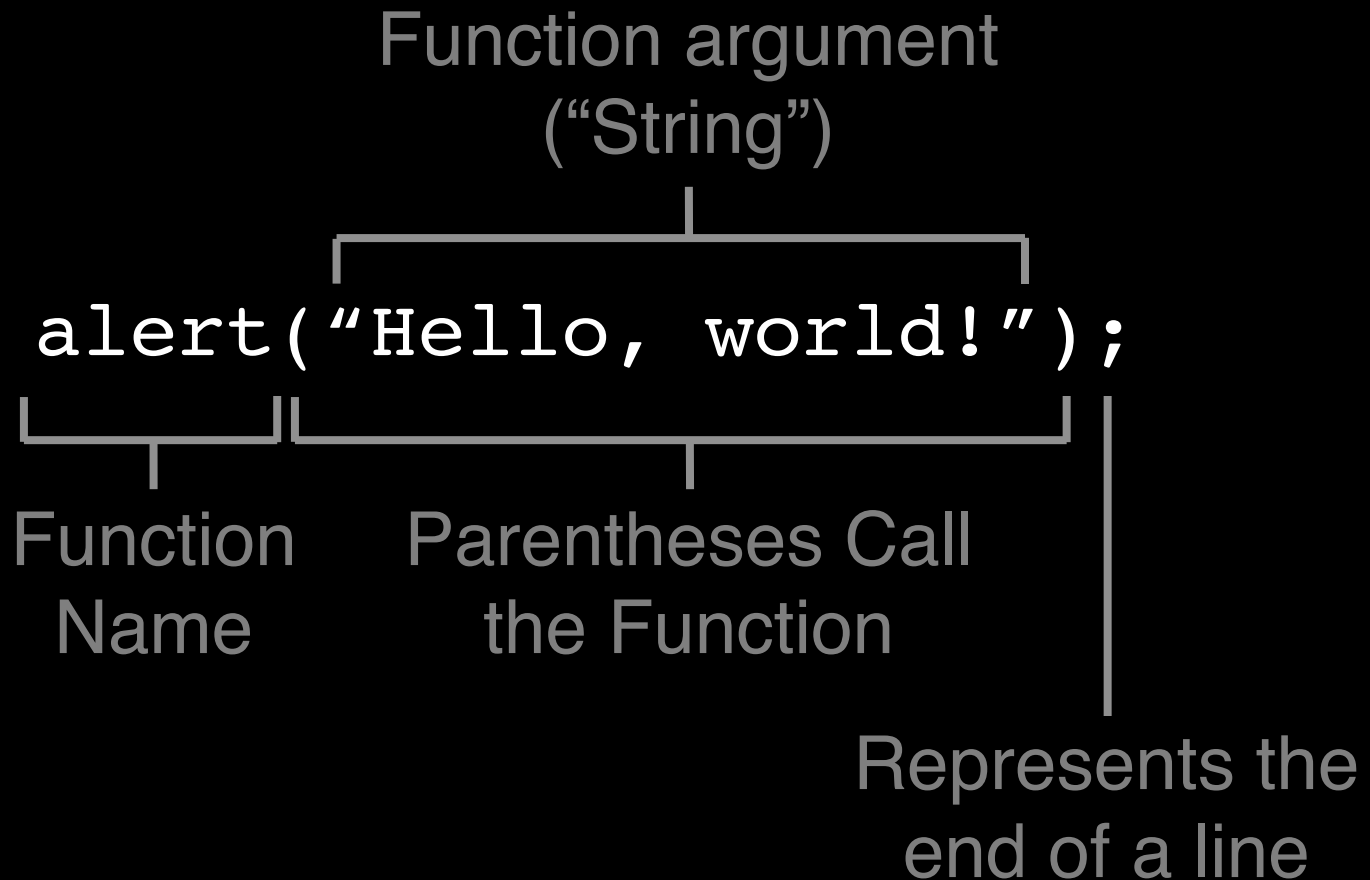
What are the possibilities?

- Add interactivity to HTML code
- Can react to events (i.e. onLoad, onClick)
- Browser detection
- Complex algorithms

What we will cover

- Types
- Operators
- Conditional Code
- Loops
- Arrays
- Objects
- Functions
- Scope

Simple Line of JS



External JS

```
<html>
  <head>
    <title>Using External JS</title>
    <script src="/path/to/example.js">
    </script>
  </head>
  <body>
  </body>
</html>
```

Internal JS

```
<html>
  <head>
    <title>Using Internal JS</title>
    <script>
      alert( "Hello World!" );
    </script>
  </head>
  <body>
  </body>
</html>
```

Inline JS

```
<html>
  <head>
    <title>Using Inline JS</title>
  </head>
  <body>
    <button onClick="alert('Hello');">
      Click me</button>
    <a href="javascript:alert('Hello');">
      Click me too</a>
  </body>
</html>
```


Writing code in the Console

The image shows a web browser window with the address bar at `learn.jquery.com/javascript-101/types/`. The page content includes a link to the [Objects](#) section and a heading for **Array**. The text explains that an array is an ordered collection of items, indexed from zero, and has a `.length` property. It also mentions that arrays can be created using the `Array` constructor or array literals.

```
1 | // Creating an array with the constructor:
```

The browser's developer console is open, showing the command `> alert("Hello world");` and the output `undefined`. The console toolbar at the bottom indicates 10 errors or warnings.

Syntax Basics

- Comments `//` or `/* ... */`
- Whitespace is also ignored
- Reserved words (i.e. `true`, `false`, `if`, `else`, `for`)
- Can create variables
 - `var someNumber = 4;`
 - `var someString = "abc";`
 - `var someFunction = function() {...};`

Primitive Types

- Strings (i.e. "Hello World!")
- Numbers (i.e. 1.618 or 10)
- Booleans (true or false)
- null (the absence of a value)
- undefined (no value assigned)

Object Types

- Object
- Array
- Functions

Basic Operators

```
var foo = "Hello";  
var bar = "world";  
console.log(foo + " " + bar);  
// Hello world
```

```
2 * 3 // 6  
2 / 3 // 0.6666666666
```

```
var i = 1;  
console.log( i++ ); // 1  
console.log( i ); // 2
```

Comparison Operators

```
var foo = 1, bar = 0, baz = '1';

foo == bar; // does equal-false
foo != bar; // doesn't equal-true

foo > bar; // greater than-true
foo < bar; // less than-false
foo <= bar; // less than or equal to-true

foo == baz; // does equal-true
foo === baz; // absolutely equals-false
```

Conditional Code

```
// AND (&&) OR (||)
var foo = 10;
var bar = 15;

if (foo == 10 && bar >= 20) {
    alert("First"); // Will never run
} else if (foo == 10 && bar <= 20) {
    alert("Second"); // Will run
}
```

Loops

```
for (var i = 0; i < 5; i++) {  
    // Logs "try 0", "try 1", ... "try 4"  
    console.log("try " + i);  
}
```

```
var i = 0;  
while (i < 100) {  
    // Will execute 100 times  
    console.log("Currently at " + i);  
    i++; // Increases by 1 every time  
}
```


Object using a Constructor

JS

```
var person1 = new Object;  
  
person1.firstName = "John";  
person1.lastName = "Doe";  
  
alert( person1.firstName + " " +  
person1.lastName ); // John Doe
```

Object using Object Literal Syntax

JS

```
var person1 = {  
    firstName: "Jane",  
    lastName: "Doe"  
};  
  
alert( person1.firstName + " " +  
person1.lastName ); // John Doe
```

Arrays

JS

```
var foo = [100, 80, 38];  
alert( foo[ 0 ] ); // 100  
alert( foo[ 2 ] ); // 38  
alert( foo.length ); // 3  
  
var bar = new Array(100);  
alert( bar[ 0 ] ); // undefined  
alert( bar.length ); // 100
```

Functions

JS

```
function foo() {  
    // Do something.  
};
```

```
var foo = function() {  
    // Do something.  
};
```

Using Functions

```
var greet = function( person, greeting ) {  
    var text = greeting + ", " + person;  
    return text;  
};  
  
console.log(greet("Danne", "Hi class"));  
// "Hi class, Danne"
```

Function Returning Another Function

JS

```
var greet = function( person, greeting ) {  
    var text = greeting + ", " + person;  
    return text;  
};  
  
var greeting = greet("Danne", "Hi class");  
  
greeting(); // "Hi class, Danne"
```

Global Scope

JS

```
var x;  
  
function myFunc() {  
    x = 5;  
};  
  
myFunc();  
  
console.log(x); // 5
```

Local Scope

JS

```
function myFunc() {  
    var x = 5;  
};  
  
myFunc();  
  
console.log(x);  
// ReferenceError: x is not defined
```


JavaScript Frameworks



prototype

The logo for the Prototype JavaScript framework, featuring the word "prototype" in a blue, lowercase, sans-serif font. The letter "o" is stylized with a red dot above it. The logo is set against a white rounded rectangular background with a subtle gradient and a reflection effect below it.



dōjō
toolkit

The logo for the Dojo Toolkit, consisting of the word "dōjō" in a white, lowercase, sans-serif font with a macron over the "o", and the word "toolkit" in a smaller, white, lowercase, sans-serif font below it. The logo is set against a dark gray rectangular background.



THREE.JS

The logo for Three.js, featuring the text "THREE.JS" in a bold, yellow, uppercase, sans-serif font. The letters have a slight 3D effect with a dark shadow underneath. The logo is set against a dark gray rectangular background.



mootools

The logo for the Mootools JavaScript framework, featuring the word "mootools" in a white, lowercase, sans-serif font. The letter "o" is stylized with a white dot above it. The logo is set against a dark gray rectangular background.

The image shows a screenshot of the jQuery website homepage as it appeared in a browser window. The browser's address bar shows 'jquery.com'. The website features a blue header with the jQuery logo and tagline 'write less, do more.'. A navigation bar includes links for 'Download', 'API Documentation', 'Blog', 'Plugins', and 'Browser Support', along with a search box. The main content area highlights three key features: 'Lightweight Footprint' (32kB minified and gzipped), 'CSS3 Compliant' (supports CSS3 selectors), and 'Cross-Browser' (works on IE, Firefox, Safari, Opera, Chrome, etc.). A prominent orange button encourages users to 'Download jQuery v1.10.2 or v2.0.3'. Below this, there are sections for 'What is jQuery?' and 'Resources', which lists links to core API documentation, learning center, blog, and contribution information.

jQuery
write less, do more.

jQuery conference
SAN DIEGO, CA
FEB 12 - 13, 2014

Download API Documentation Blog Plugins Browser Support Search

Lightweight Footprint
Only 32kB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant
Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser
[IE, Firefox, Safari, Opera, Chrome, and more](#)

Download jQuery
v1.10.2 or v2.0.3

[View Source on GitHub](#) →
[How jQuery Works](#) →

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)
- [Contribute to jQuery](#)
- [About the jQuery Foundation](#)

Corporate Members