

# Week1: Overview



# Week 1

- What is JavaScript
- How to run JavaScript
- Where to put JavaScript
- Semicolons
- Brackets
- Braces
- Parentheses
- Comments
- Data Types
- Variables

# HTML5 with JavaScript

- Javascript is ubiquitous
- This course will focus on Javascript and its manipulation of web pages and html5
- You will need to know some html5 and css3
- You will need to produce a project

# What is JavaScript?

- Javascript is NOT supported identically by all browsers
- NOT related to Java other than by name and some other syntactic similarities
- Built as a front end language, however there are frameworks such as node that have some embedded server sided functionality

# JavaScript vs Java

- interpreted, not compiled
- more relaxed syntax and rules
  - fewer and "looser" data types
  - variables don't need to be declared
  - errors often silent (few exceptions)
- key construct is the function rather than the class
  - "first-class" functions are used in many situations
- contained within a web page and integrates with its HTML/CSS content



# So what does JavaScript do?

- Controls appearance
- Forms
- Stores data locally
- Asynchronously connects to server (via ajax)
- Can present data from an API
- Has several built-in objects that can utilize helpful things like the date or math

# Why use client-side programming?

- client-side scripting (JavaScript) benefits:
  - **usability**: can modify a page without having to post back to the server (faster UI)
  - **efficiency**: can make small, quick changes to page without waiting for server
  - **event-driven**: can respond to user actions like clicks and key presses

# Where to put Javascript

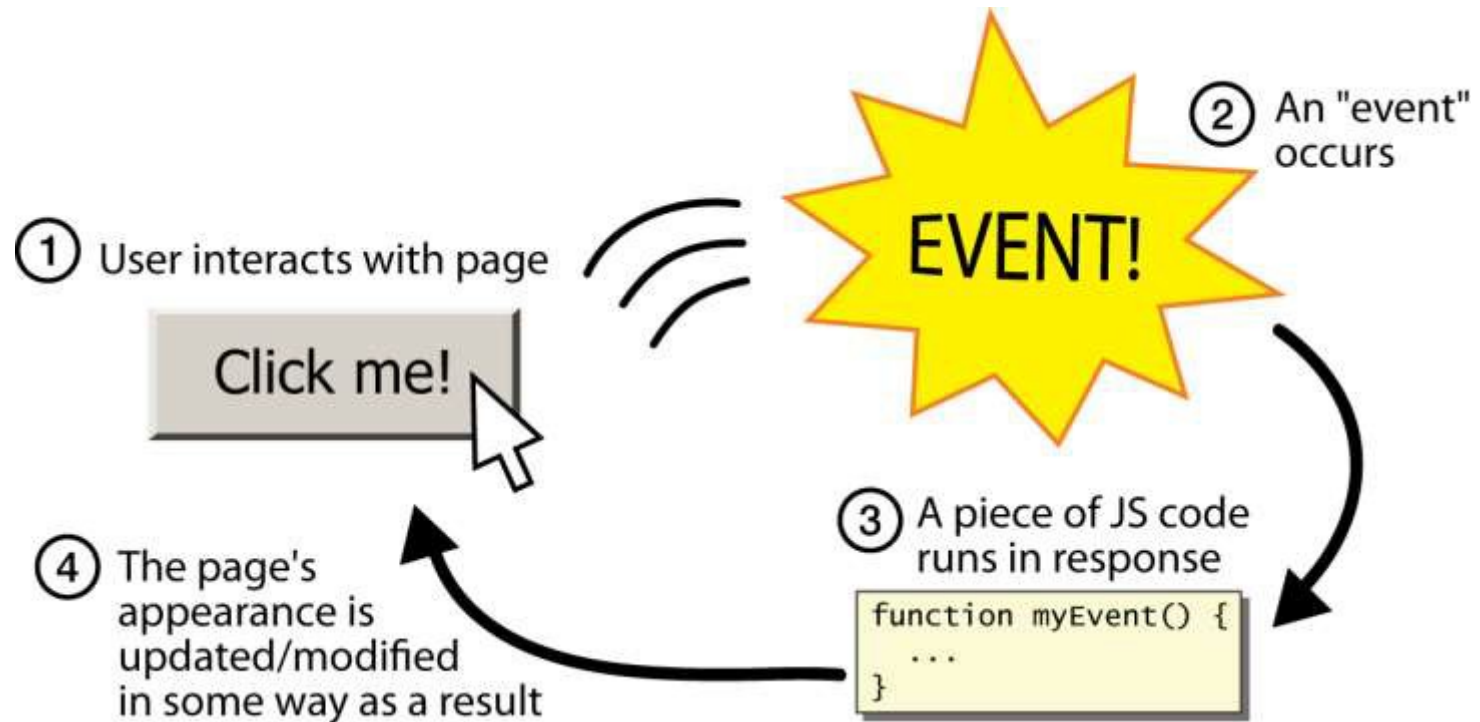
- Script tag should be placed in HTML page's head:

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

- Code is stored in a separate .js file
- JS code can be placed directly in the HTML file's body or head (like CSS)
  - but this is bad practice (should separate content, presentation, and behavior)

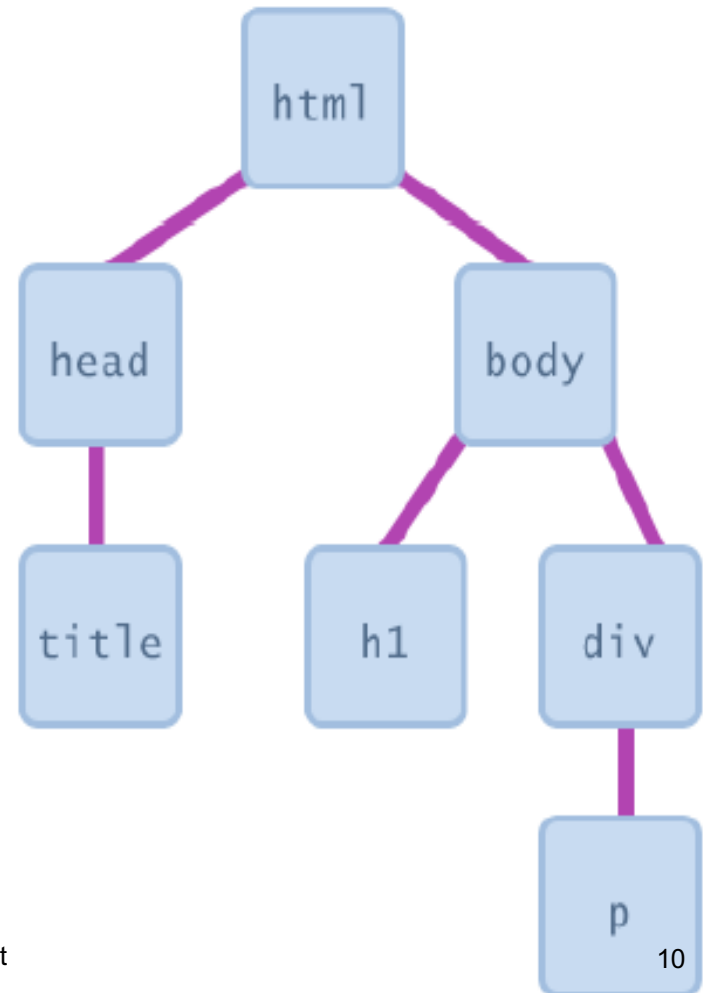


# Event-driven programming



# Document Object Model (DOM)

- Most JS code manipulates elements on an HTML page
- We can examine elements' state
  - e.g. see whether a box is checked
- We can change state
  - e.g. insert some new text into a div
- We can change styles
  - e.g. make a paragraph red



# JavaScript Syntax

- JavaScript programs are created with basic text editors just like your HTML web pages.
- The JavaScript interpreter that will be running your code will look line-by-line through the code to see what to do.
- In order for the interpreter to understand what you've written, there are some rules you must follow, i.e., syntax.

# Semi-Colons

- Semi-Colons end a statement
- Use semi-colons as good practice
- When you have statements on one line, the first semicolon is required
- You may minify your code later and semicolons will help more clearly separate statements
- Don't put a semicolon after a closing curly brace unless you are assigning a variable to a something that has braces:

```
var something = {}; //NO!!!
```

# Brackets

- `[]` are used to group elements or create objects
- `var someGroup =`  
`['student1','student2','student3'];`
- This will all be repeated during the array and object section of this course

# Braces

- Creates Object:

```
var someObject = { }
```

- Creates a group of statements:

```
var someFunction = function (){  
statement1;statement2;  
}
```

# Parantheses

- Supply parameters:

```
function callSomeCode ('arg1','arg2'){  
}
```

- Group Expressions:

```
var answer = (3*5) + 2;
```

- Execute Functions:

```
callSomeCode('arg1','arg2');
```

# Comments

- Unlike html: `<!--` →
- Single Line Comment `//`
- Multi-line comment `/* */`
- Comment your code based on two assumptions:

Assume you will get amnesia!

Assume someone else will use your code!



# Data-types

- types are not specified, but JS does have types ("loosely typed")
  - Number, Boolean, String, Array, Object, Function, Null, Undefined
  - can find out a variable's type by calling  
`var someNumber = "5";`  
`typeof(someNumber) //would equal String`

# Variables

- `var name = expression;`
- `var clientName = "Some Student";`
- `var age = 32;`
- `var weight = 127.4;`
- variables are declared with the `var` keyword (case sensitive)

# Good Habits

- Come up with your own style and stick to it. (Camel casing, Pascal casing, All Lower casing)
- Comment out code before you code to keep yourself on track
- Declare Variables on top
- Move your code to an external page based on function before you go live

# Bad Habits

- Not ending each statement with a semicolon
- Not declaring variables where necessary
- Not grouping your code for reuse
- Not putting your code in an external file
- More tips in a couple of weeks

# The most important things!

- Use Chrome or Firefox and get to know the console.
- `console.log`
- `typeof`
- Now please see my screencast demo of all the aforementioned functionalities in the Lesson 1 folder