

Client-side scripting

- Web pages can only update by reloading
 - HTTP requests take too long
 - Interactive systems must respond in $\leq 10\text{ms}$
- What updates can be made without retrieving new data?
 - Scrolling a select box, for example, has always been client-side
 - Client-side scripting lets the developer add functionality to the page without reloading it

Java (ECMA) Script

- Introduced by Netscape Navigator 2.0 w/ the original DOM
 - Only related to Java via marketing
 - Quickly “embrace & extended” by Microsoft (Jscript)
- ECMAScript is the actual standard
 - Everyone just says javascript, as in “stupid, lousy javascript”
 - Bad rep for being amateur-friendly
 - Revitalized by the AJAX paradigm shift

Java (ECMA) Script

- Loaded with a web page
 - Inline with `<script></script>` tags (no!)
 - `<script src="myscript.js"></script>`
 - Link separate .js file in head
 - All functions defined to call later
 - All statements not in a function are immediately executed
 - JavaScript console provides interactive programming
 - You can evaluate statements in the scope of the page
 - E.g. call functions from the script file
 - Fails silently! Fails silently! Fails silently!
 - (Just stops running)

Java (ECMA) Script

- C-style syntax
 - But actually more lisp/scheme based in its semantics
 - Familiar conditions, loops, functions
- Dynamic (loose) typing
 - No variable type declarations
 - (“var” is optional)

```
var someVar = 7;
```

```
someVar = “89”;
```

```
someVar + 17 = ?
```

Java (ECMA) Script

- Function declaration (no return type)

```
function bob() {  
    console.log( "hi there" );  
}
```

- Variable function arguments, return values
 - Extra formal arguments just set to null
 - Extra parameters just ignored
 - Return value is *null* if no return

Java (ECMA) Script

- Objects

- No class declaration

```
person = new Object;
```

```
person.name = "Emmett";
```

```
person.number = 7;
```

- Objects are associative arrays

```
person = { "name" : "Emmett", "number" : 7 };
```

- Member access with . or []

```
person['number'] = 6;
```

```
person.number = 7;
```

Java (ECMA) Script

- First-class functions

- Functions are data!

```
function bob() {  
    console.log( "hi there" );  
}
```

```
fn = bob;
```

- Note no parens! (Not calling bob, referring to it)

- Methods can be set at run time

```
obj = new Object;
```

```
obj.print = bob;
```

```
obj.print();
```

Java (ECMA) Script

- Anonymous functions (no name given)

```
fn = function () {  
    console.log( "hi there" );  
}
```

- Anonymous function and array notation

```
obj = { "name" : "Emmett",  
        "fn" : function() { console.log( "wow" ); }  
};  
obj.fn();
```