



JavaScript

by
Conor Twomey



History

- Written in 1995 by Brendan Eich
- Took only 2 weeks to write!
- Originally called LiveScript
- Renamed to JavaScript to take advantage of the popular release of Java
- JavaScript is now standardised as ECMAScript

History

JavaScript is NOT Java

JavaScript

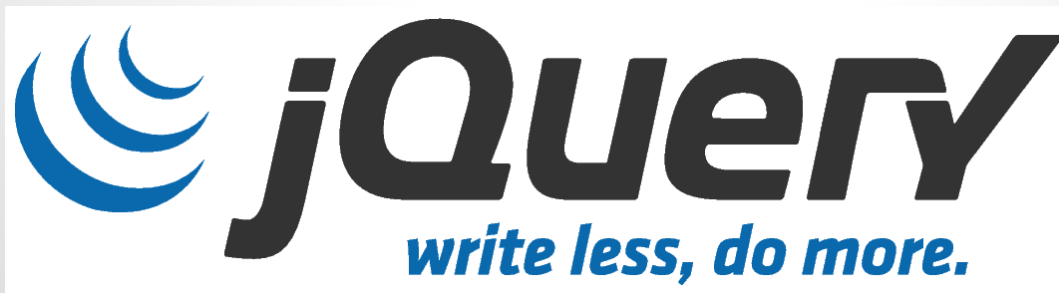
- Scripting Language
- Scripts in Browsers
- Weakly Typed
- Prototype

Java

- Compiled Language
- Applets in Browsers
- Strongly Typed

Do NOT Confuse them!

Current / Future



Let's Code!

HTML Template

```
<!DOCTYPE html>  
<html>  
<head>  
  <title></title>  
</head>  
<body>  
</body>  
</html>
```

Hello World!

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello World</title>
</head>
<body>
<script type="text/javascript">
  function helloWorld() {
    var message = "Hello World!";
    alert(message);
  }
  helloWorld();
</script>
</body>
</html>
```

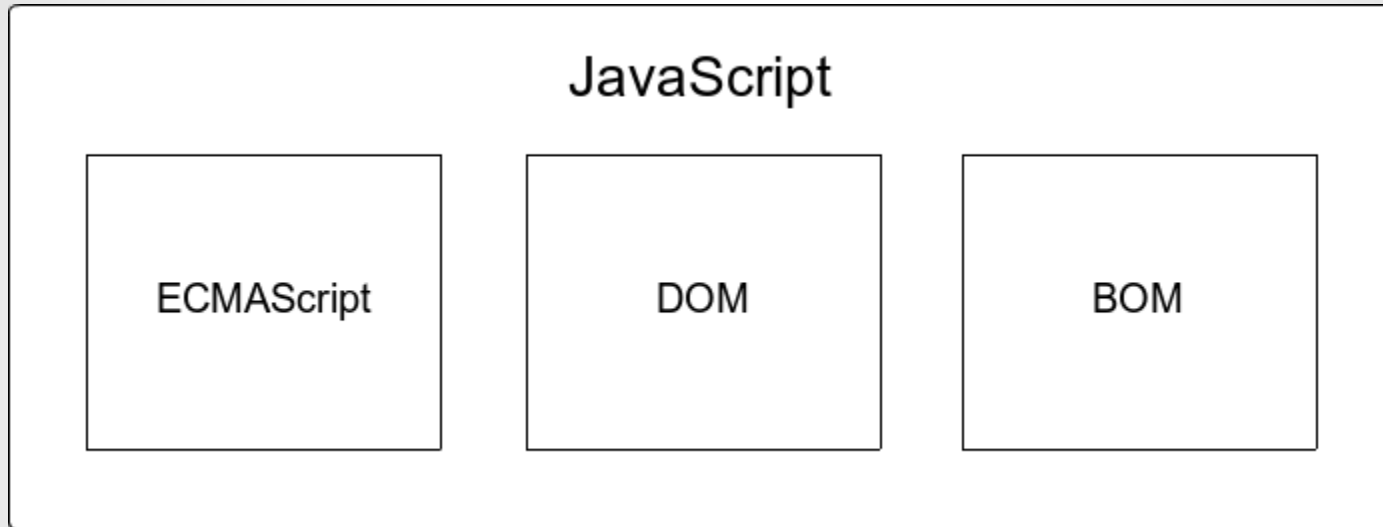
<script> Tag

- The `<script>` and `</script>` Tags are used to mark where the JavaScript starts and ends
- JavaScript can be local or in an external file

```
<script type="text/javascript">  
    ...  
</script>
```

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


Browsers, DOM, BOM

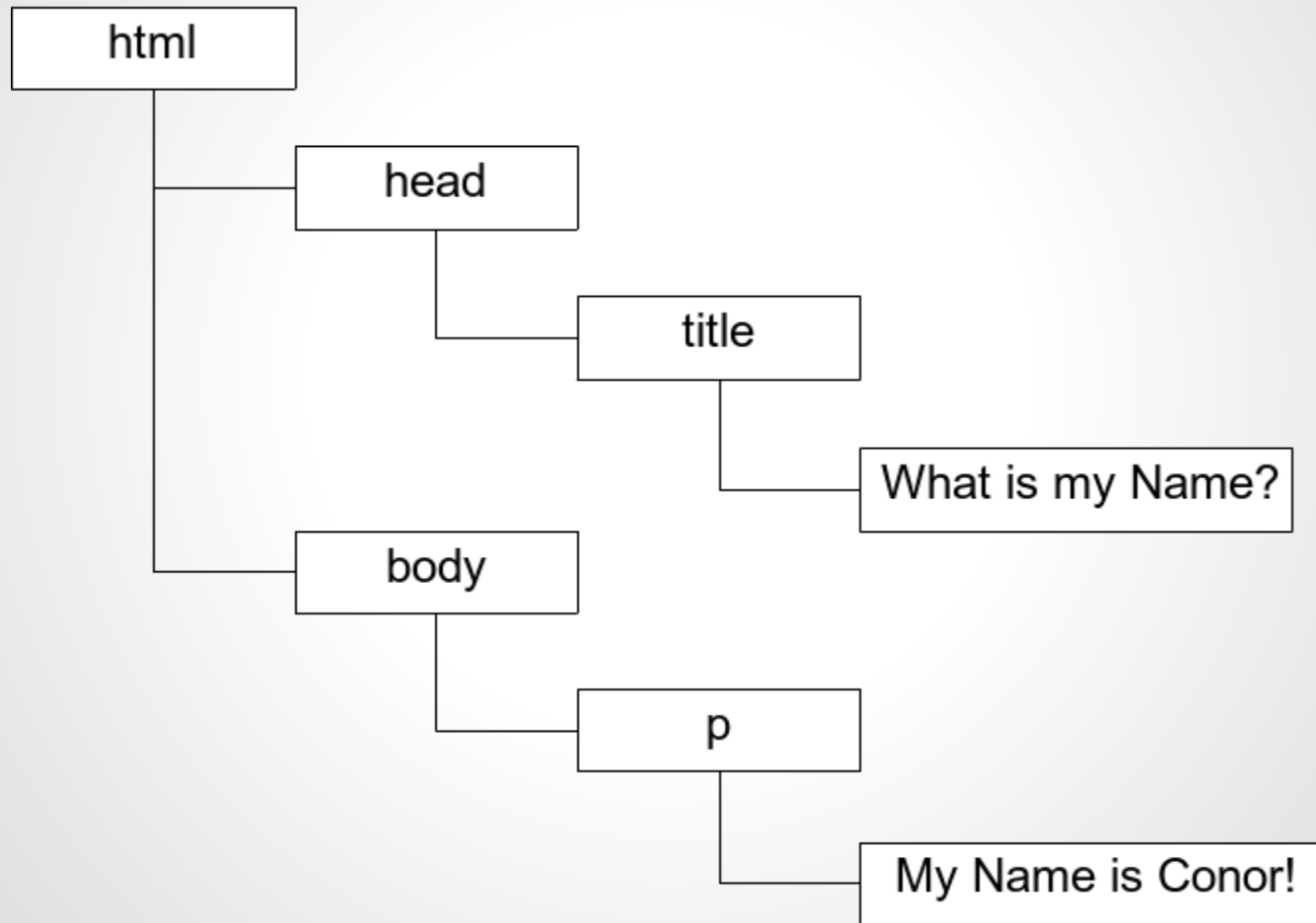


- Document Object Model (DOM)
- Browser Object Model (BOM)

What is my Name?

```
<!DOCTYPE html>
<html>
<head>
  <title>What is my Name?</title>
</head>
<body>
  <p id="name">My Name is Conor!</p>
</body>
</html>
```

Document Object Model (DOM)



What is my Name?

```
<!DOCTYPE html>
<html>
<head>
  <title>What is my Name?</title>
</head>
<body>
  <p id="name">My Name is Conor!</p>
  <button type="button" onclick="changeName()">Change Name</button>
<script type="text/javascript">
  function changeName() {
    document.getElementById("name").innerHTML="My Name is Darragh!";
  }
</script>
</body>
</html>
```

Variables

- Loosely Typed
- Use **var** for local variables

```
function test() {  
    var message1 = "hi";    // local variable  
    message2 = "bye";      // global variable  
    message3;              // Undefined  
}  
test();  
alert(message1);          // error  
alert(message2);         // "bye"
```

Data Types

- 5 x Primitive Data Types - **Undefined**, **Null**, **Boolean**, **Number** and **String**
- 1 x Complex Data Type - **Object**

```
var myString = "hi"; // String
var myNumber = 123; // Number
var finished = true; // Boolean
var dontKnow; // Undefined
var empty = null; // Null
```

Data Types

- Use the **typeof** Operator to identify which one a variable is. Returns ("undefined", "boolean", "string", "number", "object", "null", "function")

```
var message = "My Name";  
var dontKnow;  
alert(typeof message);    // "string"  
alert(typeof 2013);       // "number"  
alert(typeof dontKnow);   // "undefined"
```

Boolean Operators

NOT	!
AND	&&
OR	

```
var found = true;
var a = false;
var b = true;
alert(!found);           // false
alert(a || b);          // true
alert(a && b);           // false
```


Multiplicative Operators

MULTIPLY	*
DIVIDE	/
MODULUS	%

```
var result1 = 10 * 10;  
alert(result1);           // 100  
var result2 = 25 / 5;  
alert(result2);          // 5  
var result3 = 26 % 5;  
alert(result3);          // 1
```

Additive Operators

ADD / CONCAT	+
SUBTRACT	-

```
var result1 = 5 + 6;  
alert(result1); // 11  
var result2 = 12 - 2;  
alert(result2); // 10  
  
var x = 5, y = 2;  
alert("x and y = " + x + y); // ???
```

Relational Operators

LESS THAN	<
GREATER THAN	>
LESS THAN OR EQUAL	<=
GREATER THAN OR EQUAL	>=

```
var result1 = 5 > 3;           // true
var result2 = 5 < 3;           // false
var result3 = "a" <= "b";     // true
var result4 = 3 <= 3;         // true
var result5 = "23" < "3";     // ???
```

Equality Operators

EQUAL	==
NOT EQUAL	!=
IDENTICALLY EQUAL	===
IDENTICALLY NOT EQUAL	!==

```
var result1 = ("55" == 55);  
alert(result1);           // true
```

```
var result2 = ("55" === 55);  
alert(result2);          // false
```

Statements

- Statements control flow in JavaScript
- Common in most Languages
- Statements cover

Condition	<code>if</code> Statement
Post-Test Loop	<code>do-while</code> Statement
Pre-Test Loop	<code>while</code> Statement
Iterative	<code>for</code> Statement
Break and Continue	<code>break</code> , <code>continue</code>

`if` Statement (Condition)

```
if (condition) {  
    statement1  
}  
else if {  
    statement2  
}  
else {  
    statement3  
}
```

if Statement

```
<script type="text/javascript">
function test () {
    var i = document.myForm.inputText.value;

    if (i > 25) {
        alert("Greater than 25.");
    }
    else {
        alert("Less than or equal to 25.");
    }
}
</script>
```

if Statement

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
<form name="myForm" onSubmit="return test()">
  <input type="text" name="inputText"/>
  <input name="Test" type="submit"/>
</form>
</body>
</html>
```


do-while Statement

```
do {  
    statement  
}  
while (expression)
```

do-while Statement

```
<script type="text/javascript">
function test () {
    var i = document.myForm.inputText.value;

    do {
        alert("Number of Alerts Left: " + i);
        i = i - 1;
    }
    while (i > 0)
}
</script>
```