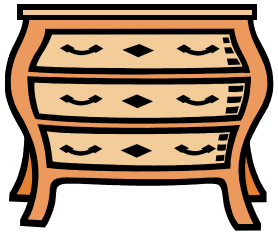
The University Of Sheffield.

COM1008: Web and Internet Technology

Lecture 8: JavaScript

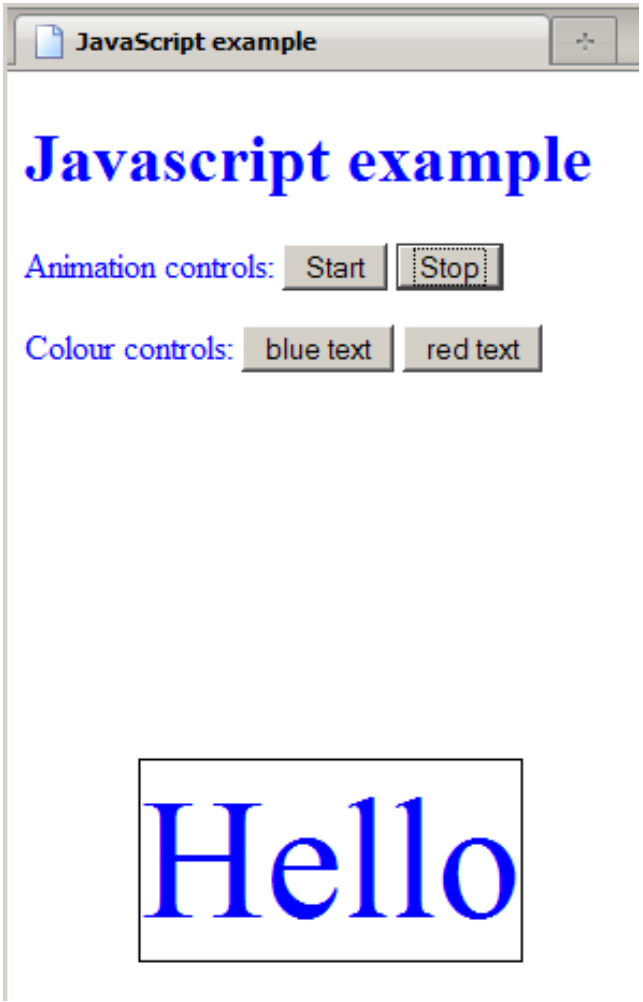


Dr. Steve Maddock
s.maddock@sheffield.ac.uk

1. Introduction

For a Web site:

- Structure using HTML
- Appearance using CSS
- Behaviour using **JavaScript**
  - Although, see recent CSS3 features, e.g. animation
- JavaScript can be used to:
  - Interact with the user
  - Control the web browser
  - Alter the document content
  - Examples: Gmail, Twitter, Firefox

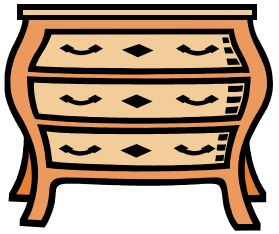


1. Introduction

- We'll start with the basics of the JavaScript language

> "First learn stand, then learn fly."
>
> *(Karate Kid, 1984)*

- Subsequent weeks:
  - Document Object Model
  - Event handling
  - HTML5 Canvas



2. JavaScript vs. Java

- Wikipedia:

> "**Java** is a computer programming language that is concurrent, class-based, **object-oriented**"

> "**JavaScript** is a[n interpreted,] **prototype-based** scripting language with dynamic typing and has first-class functions."

- JavaScript is not Java

Java	JavaScript
Static typing	Dynamic typing – a variable can hold an object of any type
Loaded from compiled bytecode	Loaded as human-readable text; Interpreted programming language

- JavaScript is like Java in some ways
  - Both have a structured C-like syntax (e.g. if, while, switch)
  - Both are case-sensitive

3. Beware: JavaScript issues

- Different versions of browsers have differing support when JavaScript is used to control Web page elements
- Solutions:
  - Detect browser and change script accordingly
  - Use a library or toolkit (e.g. jQuery) which handles browser differences
- Development
  - Progressive enhancement – start basic for all, then enhance
  - Degrade gracefully – Web page should remain usable if a feature is not supported

4. A first program

- Write a program to calculate the area of a room
- To make such a program flexible, we need:
  - Input – ask user the size of the room
  - Variables – store values used in calculations
  - Calculation – area of a room
  - Output – display results to a user

4. A first program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title>
</head>

<body>

<h1>Example</h1>
<script>
  var length = prompt("Rectangle length in cm?");
  var width = prompt("Rectangle width in cm?");
  document.write("Area = " + length*width);
  alert("Area = " + length*width);
</script>

</body>

</html>
```

The **script** element identifies and contains the JavaScript

demo

4.1 Input

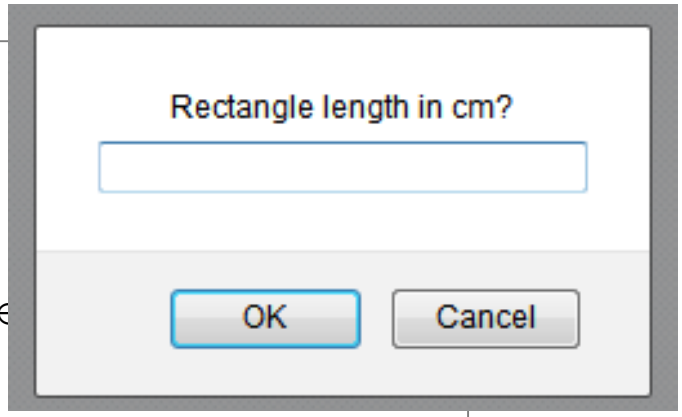
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title
</head>

<body>

<h1>Example</h1>
<script>
  var length = prompt("Rectangle length in cm?");
  var width = prompt("Rectangle width in cm?");
  document.write("Area = " + length*width);
  alert("Area = " + length*width);
</script>

</body>

</html>
```



prompt() produces a popup box to get input

Note: Return type of a call to prompt() is a ***string***

Alternative to prompt(): create a form with a series of fields – see a later lecture

demo

4.2 Variables

demo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title>
</head>

<body>

<h1>Example</h1>
<script>
  var length = prompt("Rectangle length in cm?");
  var width = prompt("Rectangle width in cm?");
  document.write("Area = " + length*width);
  alert("Area = " + length*width);
</script>

</body>

</html>
```

The variables are named **length** and **width**

Note: There is no type given in the variable definition

4.2 Variables

JavaScript

```
var score = 3;
score = 4.2;
score = 'not enough';
```

Java

```
int score = 3;
double scoreB = 4.2;
String message = "not enough";
```

- JavaScript data types
  - Number: 3, -3.1, 2.456
  - String: “hello world”, ‘hello world’, “That’s all folks”
  - Boolean: true, false
- Don’t use keywords as variable names

Duck typing

- A style of dynamic typing in which current properties determine valid semantics.
- *"when I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck.“ (attributed to James Whitcomb Riley – see http://en.wikipedia.org/wiki/Duck_typing)*

4.3 Calculation

demo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title>
</head>

<body>

<h1>Example</h1>
<script>
  var length = prompt("Rectangle length in cm?");
  var width = prompt("Rectangle width in cm?");
  document.write("Area = " + length*width);
  alert("Area = " + length*width);
</script>

</body>

</html>
```

Note: Return type of a call to prompt() is a *string*

String concatenation

Consider: Automatic type conversion

4.3 Calculation

- Same arithmetic operators as Java
  - + , - , * , / , % , = , +=, -=, *=, /=, %=

https://developer.mozilla.org/en/JavaScript/Reference/Operators/Arithmetic_Operators

- The + operator is overloaded…

- Same precedence rules as Java for complicated expressions
  - Use brackets to clarify meaning

- A Math object exists with similar methods and properties to Java
  - E.g. abs(x), ceil(x), cos(x), exp(x), floor(x), pow(x,y), random(), round(x), sin(x), sqrt(x), tan(x), etc.
  - E.g. `var x = Math.random();` gives 0.0 <= x < 1.0

4.3.1 The '+' operator is overloaded

- "The '+' operator can be used in the following ways:
  - string concatenation
  - arithmetic addition
  - to convert strings to numbers
  - It also has special meaning when used in a 'regular expression'

http://en.wikipedia.org/wiki/JavaScript_syntax

```
var vatRate = 0.15;
var costWithoutVat = 3;
var vat = costWithoutVat*vatRate;
var costWithVat = costWithoutVat + vat;
var part1 = "Hello";
var part2 = "world!!";
var message = part1 + " " + part2;
```

4.3.2 Automatic type conversion

- Here, score is automatically converted to a string

```
var score = 8;
var message = "Score out of 10: " + score;
```

- But, this can cause problems with the + operator
(https://developer.mozilla.org/en/Core_JavaScript_1.5_Guide/Core_Language_Features#Values)

```
var value1 = "37";
var value2 = 7;
var unexpectedResult = value1 + value2; // returns "377"
var expectedResult = value1 - value2;   // returns 30
```

4.3.3 Conversion

```
var value1 = "37";
var value2 = 7;
var unexpectedResult = value1 + value2;   // returns "377"
var expectedResult = value1 - value2;     // returns 30
```

- Solution: conversion

```
var result = +value1 + value2;          // returns 44
var result = Number(value1) + value2; // returns 44
```

- However, if value1 contains letters rather than digits, the result is NaN

```
var length = parseFloat(prompt("Rectangle length in cm?"));
```

4.4 Output

demo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title>
</head>

<body>

<h1>Example</h1>
<script>
  var length = prompt("Rectangle length in cm?");
  var width = prompt("Rectangle width in cm?");
  document.write("Area = " + length*width);
  alert("Area = " + length*width);
</script>

</body>

</html>
```

document.write() writes to the Web page

alert() produces a popup box containing a message

4.4 Output

demo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title>
</head>

<body>

<h1>Example</h1>
<script>
  var length = prompt("Rectangle length in cm?");
  var width = prompt("Rectangle width in cm?");
  document.write("<p>Area = " + length*width + "<\/p>");
  alert("Area = " + length*width);
</script>

</body>

</html>
```

document.write() writes to the Web page – should include html tags

\/p – the '\' is required by the specification and validators, but browsers will understand /p without the '\'

4.5 Comments

```
<script>
  var length = prompt("Rectangle length in cm?");
  var width = prompt("Rectangle width in cm?");
  document.write("Area = " + length*width);
  alert("Area = " + length*width);

  // single line comment
  /* multiple-line comment
     multiple-line comment */
</script>
```

Comments similar to other programming languages

5. Debugging

- Open the console tab from the Web Developer tools available in browsers
  - A debugger is available
- Can use console.log() within the script

```
<script>
  var length = prompt("Rectangle length in cm?");
  console.log("length=" + length);
  var width = prompt("Rectangle width in cm?");
  console.log("width=" + width);
  document.write("Area = " + length*width);
  alert("Area = " + length*width);
</script>
```

6. Logic and control

- As with Java, there is:
  - if..else
  - switch
  - for
  - while
  - do..while

6.1 The if statement

```
if (a>1 && a<10) {
  // a is between 1 and 10
  if (a>5) {
    // a is between 5 and 10
  }
  // and so on...
}

if (key=='a' || key=='b') {
  // do something
}

if (!valid) {
  // display errors
}
```

```
var x = 100;
var y;
if (x>=50) {
  y = 0;
}
else {
  y = 1;
}
```

The ternary operator

```
(condition) ? A : B;
```

```
var x = 100;
var y = (x>=50) ? 0 : 1;
```

6.2 for loop

- Example: Write 6 heading styles

```
for (i = 1; i <= 6; i++) {
  document.write("<h" + i + ">This is heading " + i);
  document.write("</h" + i + ">");
}
```

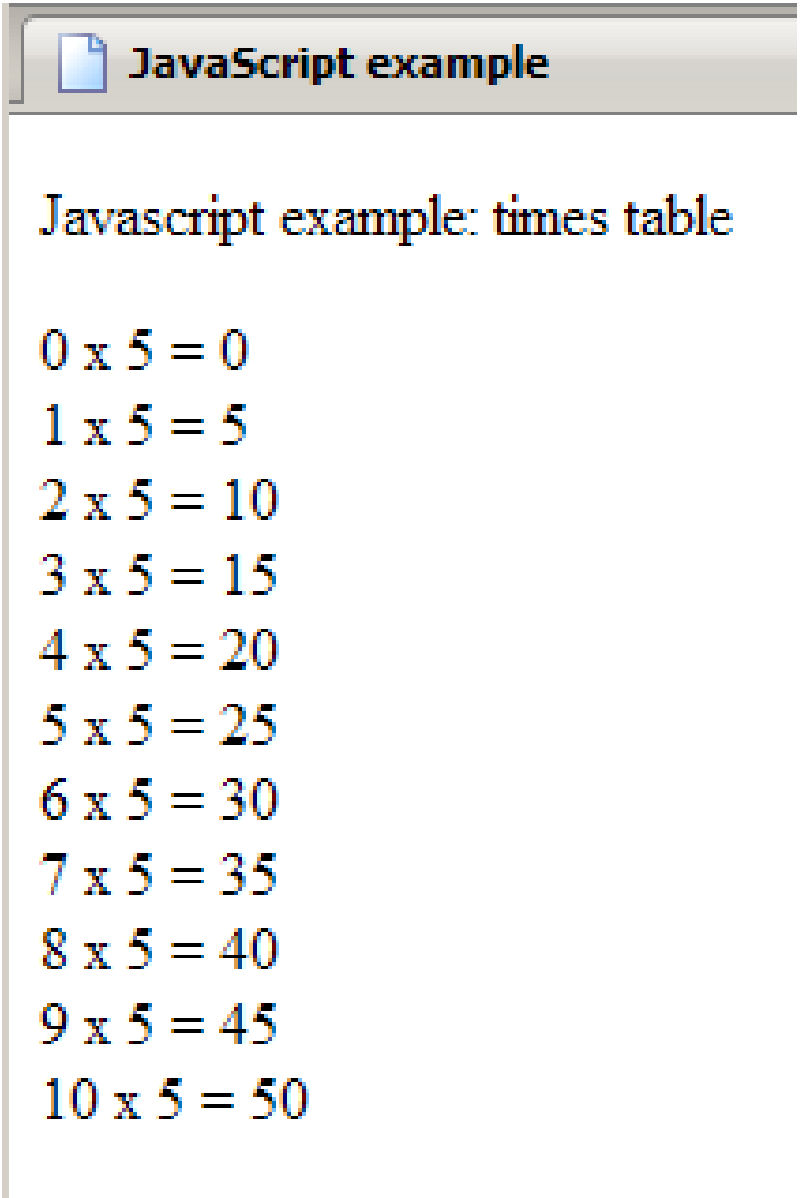
6.3 while loop

- Example while loop: repeat get a number until !isNaN

```
var x = prompt('x?');
while (isNaN(x)) {
  document.write('<p>not a number, try again</p>');
  x = prompt('x?');
}
document.write('<p>Number is '+x+'</p>');
```

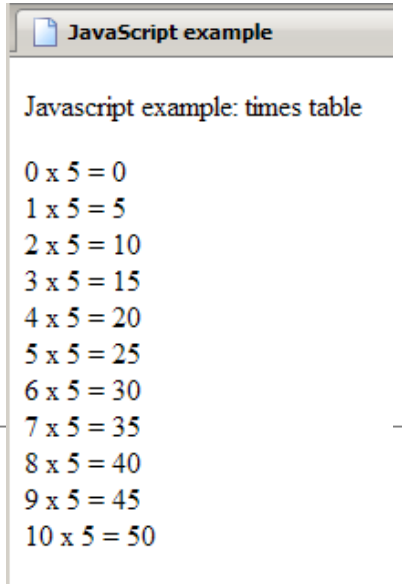
6.4 Exercise

- Write a program to display a times table. The particular times table to display is given by the user.



6.4 Exercise

- Write a program to display a times table. The particular times table to display is given by the user.




```
<body>
<p>Javascript example: times table</p>
<script>
  var number = prompt('Which times table (e.g. 2)? ');
  number = Number(number);
  for (var i=0; i<12; i++) {
    document.write(i + " x " + number + " = "
                   + i*number);
    document.write('<br />');
  }
</script>
</body>
```

7. More on scripts

- The script can be placed in a number of locations in the html file:
  - In the body
  - In the head
  - In a separate file

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
<h1>Example</h1>
  <script>
  </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
 <script>
 </script>
</head>
<body>
<h1>Example</h1>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="utf-8" />
   <title>JavaScript examples</title>
  <script src="a_script.js"></script>
</head>
<body>
  <script src="b_script.js"></script>
</body>
</html>
```

```
/* script in
separate file */
```

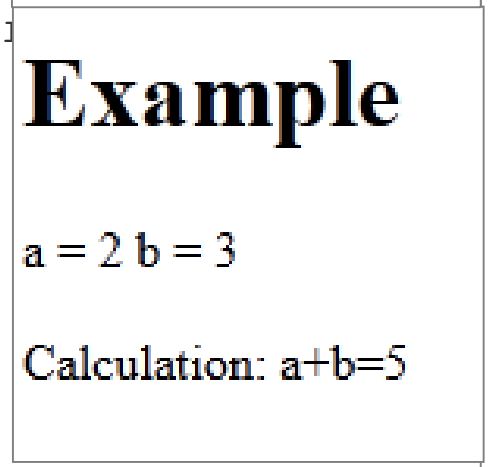
```
/* script in
separate file */
```

7.1 In the body

- The results of executing the script are placed at the location of the script in the html file

```
<!DOCTYPE html><html lang="en">
<head><meta charset="utf-8" /><title>JavaSc
examples</title></head>
<body>
<h1>Example</h1>
<script>
  var a = 2, b = 3;
  document.write("a = " + a);
  document.write(" b = " + b);
  document.write("<br \/>");
  document.write("<p>Calculation: a+b=" + (a+b) + "<\/p>");
</script>
</body>
</html>
```

Resulting web page display:



Examples of string concatenation

7.2 In the head

- The script is executed and the results are displayed before the body is loaded – this can be controlled (see a later lecture)

```
<!DOCTYPE html><html lang="en">
<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title>
  <script>
    var a = 2, b = 3;
    document.write("a = " + a);
    document.write(" b = " + b);
    document.write("<br \/>");
    document.write("<p>Calculation: a+b=" + (a+b) + "<\/p>");
  </script>
</head>
<body>
  <h1>Example</h1>
</body>
</html>
```

Resulting web page display:

a = 2 b = 3

Calculation: a+b=5

Example

7.3 In a separate file

- Script is usually in a separate file
- Multiple scripts can be included

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>JavaScript examples</title>
  <script src="write1_script.js"></script>
</head>

<body>
  <h1>Example</h1>
  <script src="write1_script.js"></script>
</body>

</html>
```

Resulting web page display:

a = 2 b = 3

Calculation: a+b=5

Example

a = 2 b = 3

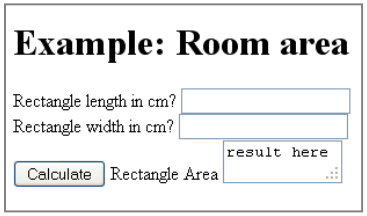
Calculation: a+b=5

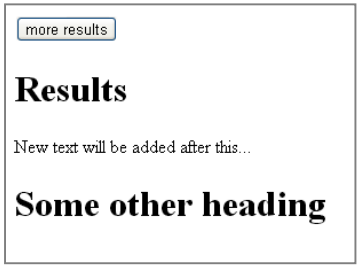
write1_script.js

```
var a = 2, b = 3;
document.write("a = " + a);
document.write(" b = " + b);
document.write("<br \/>");
document.write("<p>Calculation: a+b=" +
(a+b) + "<\/p>");
```

8. Summary

- For a Web site: Structure using HTML; Appearance using CSS; Behaviour using JavaScript
- JavaScript is NOT Java
  - E.g. JavaScript uses dynamic typing
- JavaScript is like Java in some ways, e.g.
  - Both have a structured C-like syntax (e.g. if, while, switch)
  - Both are case-sensitive
- Coming soon…
  - Arrays, Functions, Objects, Events, DOM and Canvas
  - Input and output on a form
  - Altering parts of the web page, e.g. adding content





Appendix. An example program. Version 1

- Write a program that calculates the minimum number of coins required to make up a required amount of money given in pence, e.g. 457 pence is two £2 coins, no £1 coins, one 50p coin, etc.
- Hints:
  - 50/8=6.25
  - (There is no integer division as all numeric values are floating point values in JavaScript.)
  - However, Math.floor(50/8) = 6
  - Also, 50%8 = 2
  - % is the modulus operator, i.e. the remainder of a division operation
- Solution


```
var pence=prompt('Amount in pence? ');
document.write('<p>Amount in pence: '+pence+'</p>');
var coins = Math.floor(pence/200);
document.write('<p>&pound;2 coins: '+coins+'</p>');
pence = pence%200;
coins = Math.floor(pence/100);
document.write('<p>&pound;1 coins: '+coins+'</p>');
pence = pence%100;
coins = Math.floor(pence/50);
document.write('<p>50p coins: '+coins+'</p>');
pence = pence%50;
coins = Math.floor(pence/20);
document.write('<p>20p coins: '+coins+'</p>');
pence = pence%20;
coins = Math.floor(pence/10);
document.write('<p>10p coins: '+coins+'</p>');
pence = pence%10;
coins = Math.floor(pence/5);
document.write('<p>5p coins: '+coins+'</p>');
pence = pence%5;
coins = Math.floor(pence/2);
document.write('<p>2p coins: '+coins+'</p>');
pence = pence%2;
coins = Math.floor(pence);
document.write('<p>1p coins: '+coins+'</p>');
```

Appendix. An example program. Version 2

- Write a program that calculates the minimum number of coins required to make up a required amount of money given in pence, e.g. 457 pence is two £2 coins, one 50p coin, one 5p coin and one 2p coin. Only output coins that are part of the change.
- Solution

```
var pence=prompt('Amount in pence? ');
document.write('<p>Amount in pence: '+pence+'</p>');
var coins = Math.floor(pence/200);
if (coins!=0)
  document.write('<p>&pound;2 coins: '+coins+'</p>');
pence = pence%200;
coins = Math.floor(pence/100);
if (coins!=0)
  document.write('<p>&pound;1 coins: '+coins+'</p>');
```

```
pence = pence%100;
coins = Math.floor(pence/50);
if (coins!=0)
  document.write('<p>50p coins: '+coins+'</p>');
pence = pence%50;
coins = Math.floor(pence/20);
if (coins!=0)
  document.write('<p>20p coins: '+coins+'</p>');
pence = pence%20;
coins = Math.floor(pence/10);
if (coins!=0)
  document.write('<p>10p coins: '+coins+'</p>');
pence = pence%10;
coins = Math.floor(pence/5);
if (coins!=0)
  document.write('<p>5p coins: '+coins+'</p>');
pence = pence%5;
coins = Math.floor(pence/2);
if (coins!=0)
  document.write('<p>2p coins: '+coins+'</p>');
pence = pence%2;
coins = Math.floor(pence);
if (coins!=0)
  document.write('<p>1p coins: '+coins+'</p>');
```