UNIVERSITY OF THE WEST of SCOTLAND
UWS

INTRODUCTION TO WEB DEVELOPMENT

Code: COMP07009

Week 9

Introduction to JavaScript

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JAVASCRIPT

http://w3schools.com/js/js_intro.asp

UNIVERSITY OF THE WEST of SCOTLAND
UWS

What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a client-side scripting language
- A scripting language is a lightweight programming language
- JavaScript is either embedded directly into HTML pages or in a separate file with extension .js with a link to the HTML file
- JavaScript is an interpreted language
  (scripts execute without preliminary compilation)
- JavaScript is NOT Java
- JavaScript is a lightweight, interpreted programming language
- Java is a compiled programming language

UNIVERSITY OF THE WEST of SCOTLAND
UWS

What can JavaScript do?

- JavaScript gives HTML designers a programming tool
- JavaScript can put dynamic text into an HTML page
- JavaScript can react to events
- JavaScript can read and write HTML elements
- JavaScript can be used to validate data
- JavaScript can be used to detect the visitor's browser
- JavaScript can be used to create cookies

- **The power of HTML5 is only realised with the use of JavaScript**

UNIVERSITY OF THE WEST of SCOTLAND
UWS

How does it work?

- Browsers include a JavaScript interpreter
- JS is written into an HTML page
- As the browser reads down a page, the JS interpreter interprets JS scripts
- Browser may stop if scripts contains any errors
- Some browsers have debugging tools
  - Error console in Firefox
  - Javascript console in Chrome
- Can enter the code manually into an editor
- Some applications (eg Dreamweaver) can generate JS code quickly based on user input

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Where do JS scripts go?

- In the **<head> element** inside <script> and </script> tags
  - Scripts will be automatically executed as the page loads
- In the **<body> element** inside <script> and </script> tags
  - Scripts will be executed at that point within the HTML page
- In an **external file** which only contains javascript code
  - Scripts will be automatically executed as the page loads
  - The <script> tag is inside the <head> tag within the HTML file to link to the external js file
  - Eg <script src="external.js"></script>

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Inside <head>

<head>
 <title>.....</title>
<script>
……
</script>
</head>
<body
Normal HTML
</body>
Old examples used “type=text/Javascript” within <script>
Not required as JS is default scripting language in HTML5

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Inside <body>

<head>
 <title>.....</title> ……..</head>
<body>
Normal HTML
<script>
……
</script>
Normal HTML
</body>

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JS Syntax

- Scripts consist of series of instructions called **statements**
- Statements should be on separate lines
- Each statement should end with a semi-colon

```
statement 1;
statement 2;
```

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Comments

- Comments will be ignored by the JavaScript interpreter
- Use them to make sense of your scripts
- Single line
- Start with //
- // This is a single line comment
- Multiple lines
- Start with /* and end with */
- /* JS .....
  - over several lines.....
  - .....
- */

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JS Data Types

- String (any text inside single or double quotes)
- Number (with or without decimals, doesn’t need quotes)
- Boolean (true or false)
- Undefined (no value)
- Null (no content, different from value of zero)
- Array
- Object

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JS Variables

- JS variables are “containers” for storing information
- Variable names
  - must begin with a letter
  - Case sensitive so x and X are different
- var keyword declares a variable var x;
- “=“ assigns a value to a variable var x=5;
- Can declare a variable and assign a value at the same time
- var x=20;
- JS has dynamic types / JS is weakly typed
  - The same variable can be used as different data types
  - var x; x is declared but undefined
  - var x=5; x is re-declared and is now a Number
  - var x=“Fred” x is re-declared and is now a String

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JS Variables

- var x=100;
- var pi=3.14159;
- var message =“Hello”;
- var y=false;
- var dob=“Nov 13, 1989”
- var cars=new Array(); // more on Arrays later

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JS Operators

=	Assignment
+	Addition
*	Multiplication
==	Equality
===	Exactly equal to (value and type)
!=	Not equal to
!==	Not equal to (neither value nor type)
>	Greater than
<	Less than
++	Increment
--	Decrement
-	Subtraction
/	Division

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JS Logical Operators

Based on **x=6 and y=3**

• &&	• And	(x<10 && y>1) is true
• ||	• Or	(x==5 || y==5) is false
• !	• Not	!(x==y) is true

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JavaScript Objects

- Almost everything in JS can be an Object:
  - Strings, Functions, Arrays, Dates …
- Objects have properties and methods
- Properties are values associated with objects
- Methods are actions that objects can perform
- Car object may have properties such as:
  name, model, colour
- Each car will have these properties but the values will differ between them
- Car object may have methods such as:
  start(), drive(), brake(), changeGear()
- All cars have these methods but they are performed at different times

UNIVERSITY OF THE WEST of SCOTLAND
UWS

JS Objects

- Predefined objects such as:
  window
  Note: JS assumes the properties and methods referenced in code belongs to the window object if an object is not specified
- document
- Date (makes date and time manipulations easy)
- Array (stores multiple values in a single variable)
- Math (provides access to a mathematical functions)
- Objects can also be custom built

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Syntax for Accessing Object

- Object Properties
  - objectName.propertyName

```
var message="HelloWorld";
message.length;
Outputs -> 11
```

- Object Methods
  - objectName.methodName()

```
var message="Hello World";
message.toUpperCase();
Outputs - > "HELLO WORLD"
```

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Write to the Document

- Writing HTML into the document window

```
<script>
document.write(“<h1>Hello world!</h1>") ;
document.write(“<h2>Writing to the Document</h2>”);

//can also be done
document.write(“<h1>Hello world!</h1><h2>Writing to the Document</h2>”);

</script>
```

Hello world!

Writing to the Document

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Change HTML elements

- Manipulate the content of a <p> element when the page loads

```
<body>
<h1>Javascript</h1>
<p id="demo">My First Javascript</p>
<script>
document.getElementById("demo").innerHTML="My First Paragraph";
</script>
</body>
```

- A <p> element has been assigned an id=“demo”
- JS is placed inside the <script> element within the <body> element AFTER the <p> element

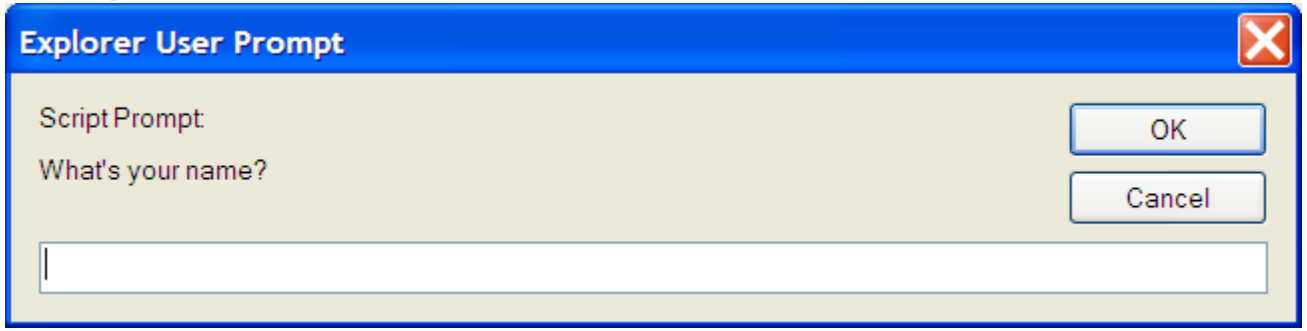
UNIVERSITY OF THE WEST of SCOTLAND
UWS

Window object

- Alert and prompt methods

```
<script>
window.alert("Hello World!");
window.prompt("What's your name?","");
</script>
```

- If empty quotes not used, then undefined would appear in prompt window

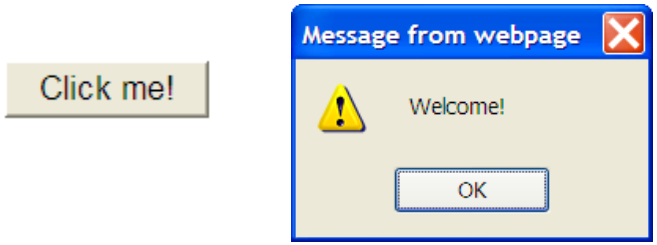


UNIVERSITY OF THE WEST of SCOTLAND
UWS

React to Events

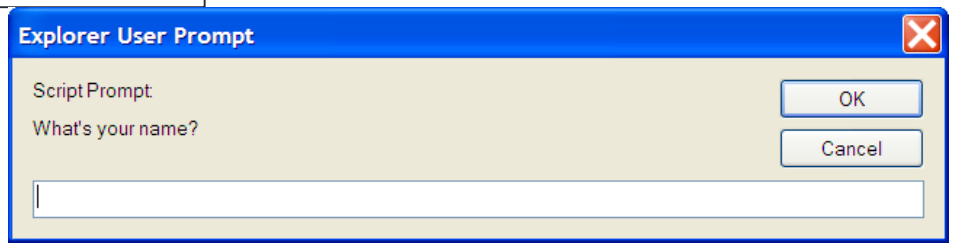
- The click of a button is one of many HTML events

```
<button type="button" onclick="alert('Welcome!')">Click me!</button>
```



- alert statement within the double quotes
- Welcome text within single
- Different quotes also useful to show quotes within text

```
window.prompt("What's your name?","");
```



UNIVERSITY OF THE WEST of SCOTLAND
UWS

Backslash(Escape character)

- Used to represent common characters
- \n new line
- \’ single quote
- \” double quote
- \t tab
- \\ backslash
- \r carriage return
- Backslash pairs are known as inline or escaped characters


```
window.alert("This\nis\nover\nseveral\nlines");
```

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Functions

- A block of code which performs a specific task
- Needs to be defined
- Will be executed when it is called
- Can be called directly when an event occurs (user clicking on a button) and it can be called from anywhere by JS code
- JS has some built-in functions
- Can also create your own functions
- Can have multiple functions in the one file
- function keyword must be lowercase
- function name must be called with the same case as used when it was defined

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Functions

- Syntax

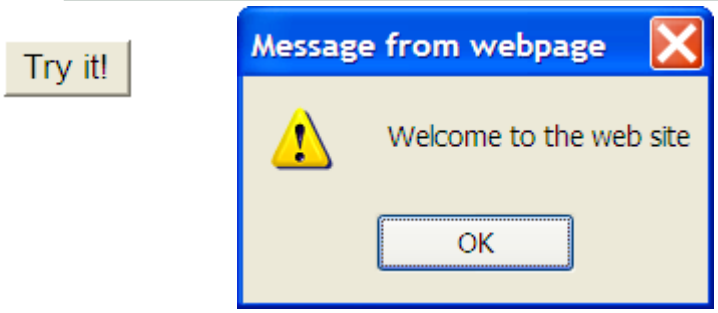
```
function functionname(){
code to be executed
}
```

Example

```
function welcomeMsg(){
alert("Welcome to the web site");
}
```

- Code placed inside the <head> tag to define it
- HTML code to create a button which calls the function when clicked

```
<button onclick="welcomeMsg()">Try it!</button>
```



UNIVERSITY OF THE WEST of SCOTLAND
UWS

Function with arguments

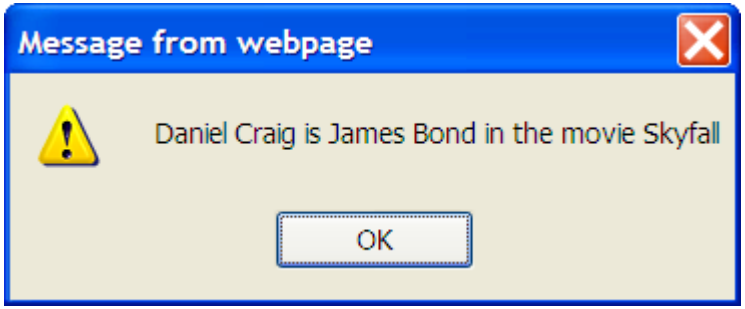
- A function can be created with arguments so that values can be passed to it
- Syntax

```
myFunction(argument1, argument2)
```

- Eg

```
function myFilm(movie,actor){
alert(actor +" is James Bond in the movie "+movie);
}
```

```
<button onclick="myFilm('Skyfall','Daniel Craig')">Try it!</button>
```



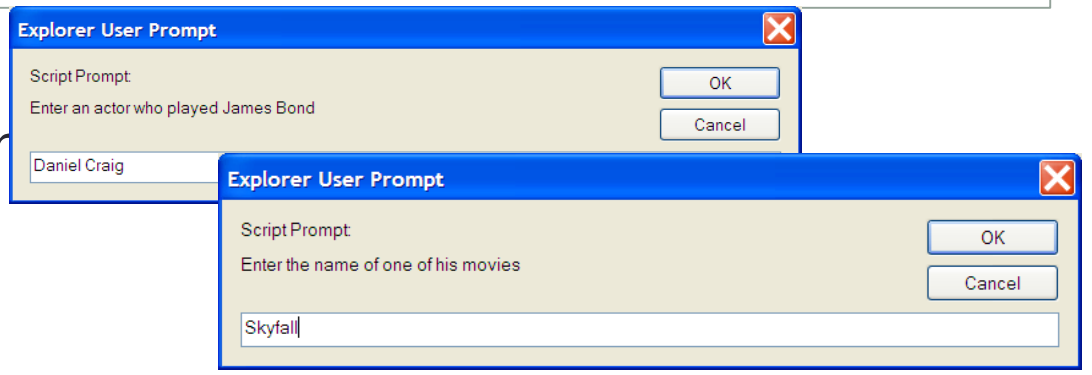
UNIVERSITY OF THE WEST of SCOTLAND
UWS

Prompt for arguments

- The user an be prompted for arguments for the function
- Eg

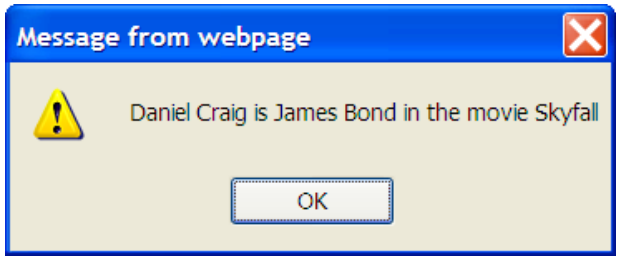
```
actor=window.prompt("Enter an actor who played James Bond","");
movie=window.prompt("Enter the name of one of his movies","");
myFilm(actor,movie);    //calling the function
```

- Function definition



- Eg

```
function myFilm(movie,actor){
alert(actor +" is James Bond in the movie
"+movie);
}
```

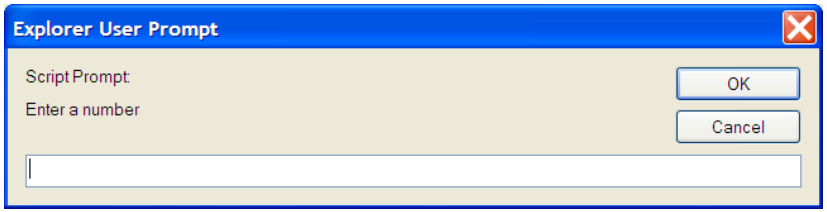


UNIVERSITY OF THE WEST of SCOTLAND
UWS

Built in Functions

- Examples isNaN() parseInt() Number()

```
x=window.prompt("Enter a number","");
if (isNaN(x)){
document.write(x +" is not a number");
}
else{
document.write(x +" is a number");
}
```



testing 1, 2, 3 is not a number

643284732.99 is a number

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Built in Properties

- Examples innerHTML

```
function myName(){
name=window.prompt("What's your name?","");
document.getElementById("demo").innerText=name
;
}
```

```
<p id="demo"></p>
```

- Called by

```
<button onclick="myName()">Try it!</button>
```

- or

```
<body onload="myName()">
```

- Name entered by user is displayed inside the <p> tag

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Built in Properties

- Examples innerText (same as innerHTML)

```
function displayDate(){
document.getElementById("demo").innerText=Date();
}
```

```
<p id="demo"></p>
```

- Called by

```
<button onclick="displayDate()">Display Date</button>
```

- Current date will be displayed inside the element with the id=“demo”

- Note: innerText works in IE but not in Firefox

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Functions with a Return

- A function can also return a value back to where the call was made

```
function addNumbers(x,y){
x=parseInt(window.prompt("Enter a number"));
y=parseInt(window.prompt("Enter a second number"));
z=x+y;
return z;
}
```

- The built-in function parseInt() is required to return integers otherwise adding 1 and 7 would give a result of 17
- Calling the function

```
addNumbers();
document.write(z);
```

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Local/Global Variables

- Using (var) to declare a variable becomes LOCAL and can only be accessed from within that function
- This allows you to have local variables with the same name in different functions
- Variable has local scope
- Local variables are deleted as soon as the function is completed
- Variables declared outside a function is GLOBAL
- This allows all scripts and functions in the page can access it
- Global variables are deleted when the page is closed
- If a variable has not been declared and is assigned a value, it automatically becomes a global variable

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Conditional Statements

- Conditional statements perform different outcomes based on different conditions being satisfied
- Execute some code only if a condition is true
- if statement
- Execute some code if the condition is true and another code if the condition is false
- if ... else statement
- Select one of many blocks of code to be executed
- if ... else if ... else Statement
- Select one of many blocks of code to be executed
- switch statement

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Syntax

- If (condition)
  {
  code to be executed if condition is true;
  }
- If (condition)
  {
  code to be executed if cond is true;
  }
  else{
  code to be executed if cond is false;
  }

```
var x="Sunday";
if (x="Sunday"){
document.write("Enjoy your day off!");
}
```

```
var x="Sunday";
if (x="Sunday"){
document.write("Enjoy your day off!");
}
else{
document.write("Enjoy work!");
}
```

UNIVERSITY OF THE WEST of SCOTLAND
UWS

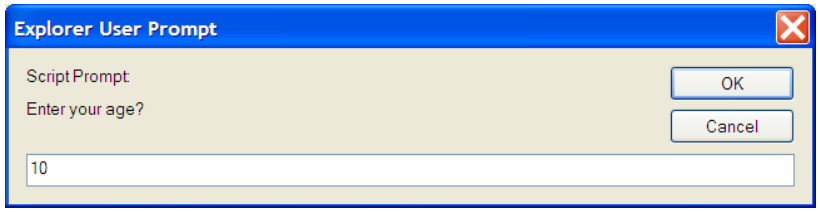
Syntax

```
If (condition1){
  code to be executed if condition1 is true
  }
else if(condition2){
  code to be executed if condition2 is true;
}
else{
  code to be executed if neither
  condition1 or condition 2 is true;
}
```

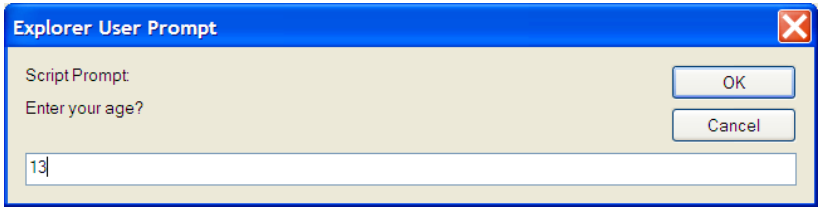
UNIVERSITY OF THE WEST of SCOTLAND
UWS

Example

```
var age=window.prompt("Enter your age?","");
if (age<13){
        document.write("You are a child");
        }
        else if (age<18){
        document.write("You are a teenager");
        }
        else{
        document.write("You are an adult");
        }
```



You are a child



You are a teenager

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Switch statement

- Perform different actions based on different conditions

```
var age=window.prompt("Enter your age?");
switch(age){
case "10":{
            document.write("You are a child");
}
break;
case "13":{
            document.write("You are a teenager");
}
break;
case "18":{
            document.write("You are an adult");
}
break;
}
```

UNIVERSITY OF THE WEST of SCOTLAND
UWS

Lab

- Work through the first Javascript lab on Moodle
- Add appropriate JS comments within the files
- Try the JavaScript Tutorial on the w3schools website