# JavaScript and AJAX

Cases, Objects and AJAX

daviesah7@cardiff.ac.uk

# JavaScript and AJAX

- The **switch** statement
  - Like a series of if … else if … else if … … else if …
  - BUT: the same variable must be tested in each condition
  - So you can't do:

```
if (a < b) { … stuff … }
else if (x > y) { … stuff … }
else if (m == n) { … stuff … }
else { … more stuff … }
```

# JavaScript and AJAX

- **switch** statement syntax:

```
switch (testvar) {
    case 'Apple': alert('Apple'); break;
    case 'Orange': alert('Orange'); break;
    case 'Pear': alert('Pear'); break;
    default: alert('Any old fruit'); break;
}
```

# JavaScript and AJAX

- **switch** statement syntax (2)
  - **break** jumps out of the case statement.  If it's not there, the next statement is executed and so on until a **break** is found or the last condition is executed (called 'falling through')
  - **default** is optional
  - Each **case** can contain any number of statements

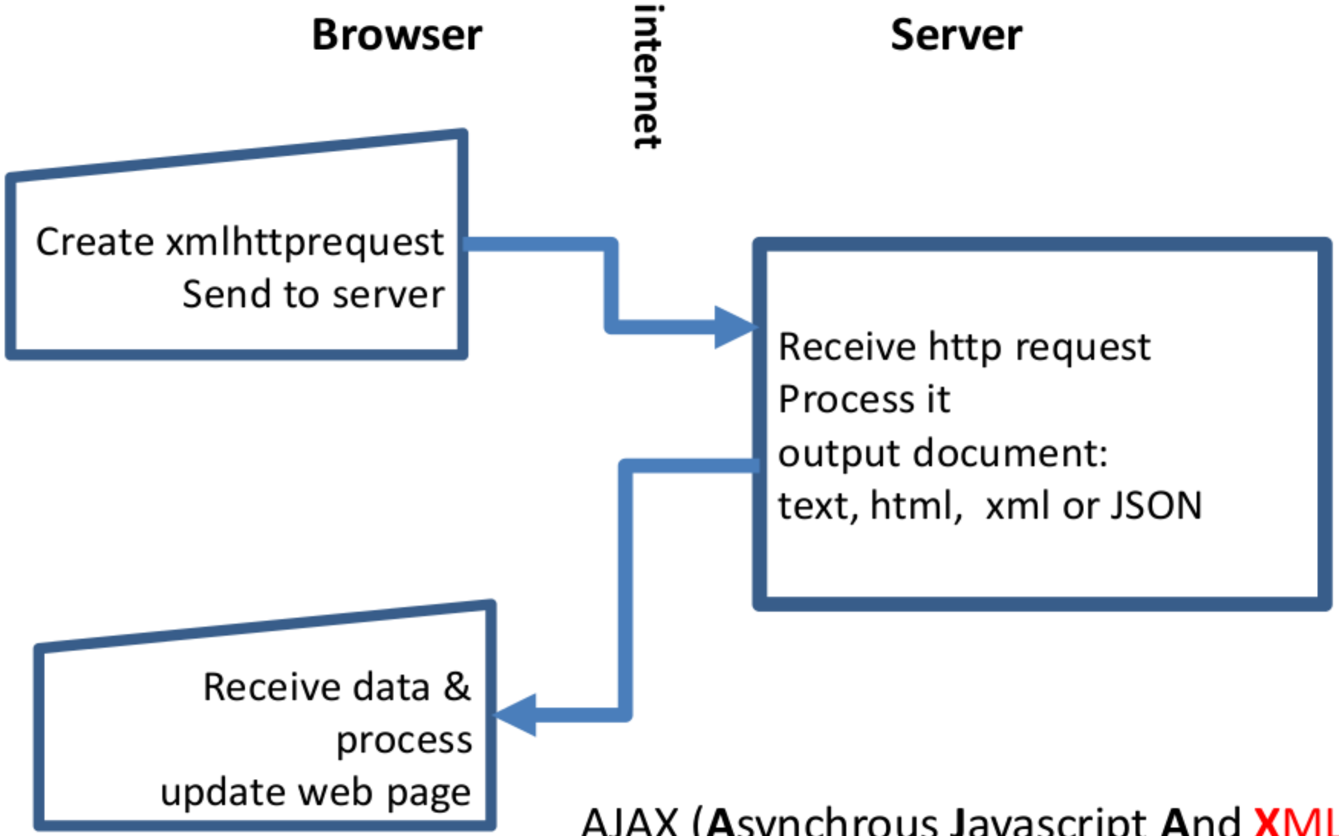
# JavaScript and AJAX

- The JavaScript Math object
  - Math.random(): generates a pseudo-random number x where 0 >= x < 1
  - Math.ceil(number): round up to the next integer
  - Math.floor(number): round down to the next integer
  - Math.round(number): rounds to the nearest integer

# JavaScript and AJAX

- Exercise: Pick a Card
  - Use the Math object, and case statements to simulate drawing random cards from a standard pack
  - To get to the starter file, follow the link in the website

# JavaScript and AJAX



**Browser**

internet

**Server**

Create xmlhttprequest
Send to server

Receive http request
Process it
output document:
text, html, xml or JSON

Receive data &
process
update web page

AJAX (**A**synchrous **J**avascript **A**nd **XML**
AHAH (**A**sychronous **HTML** **A**nd **H**TTP)

# JavaScript and AJAX

- Synchronous and Asynchronous calls
  - Synchronous: browser waits for a response
  - Asynchronous: browser carries on as normal
- During synchronous calls, the browser is unresponsive (freezes)
- Asynchronous calls execute a callback on return

# JavaScript and AJAX

- Simple synchronous call (IE7+, and all sensible browsers):

  ```
  xhr = new XMLHttpRequest();
  xhr.open("GET", "response.txt", false);
  xhr.send();
  document.getElementById("anid").innerHTML =
  xhr.responseText;
  ```

- Setting the third parameter of open() to **false** makes it a synchronous call

# JavaScript and AJAX

- Exercise: create a simple script to retrieve a text file via a synchronous AJAX call
- See course page for copy of starter HTML and text file.

# JavaScript and AJAX

**Delays and synchronous calls**

Demonstration of synchronicity:

See course notes for example.

**Note** that you cannot enter anything into the text box until the data is returned from the server

# JavaScript and AJAX

- **Microsoft browsers** before IE7 did not have an XMLHttpRequest object
- For IE6 you need to code for an ActiveX object

  ```
  xhr = new ActiveXObject('Msxml2.DOMDocument.6.0');
  ```

  or possibly:

  ```
  xhr = new ActiveXObject("Microsoft.XMLHTTP");
  ```

# JavaScript and AJAX

**For full cross-browser compatibility, you need:**

```
if (typeof XMLHttpRequest === "undefined") {
// IE6 or earlier...
  xhr = function () {
    try { return new
ActiveXObject("Msxml2.XMLHTTP.6.0"); }
    catch (e) {}
    try { return new
ActiveXObject("Msxml2.XMLHTTP.3.0"); }
    catch (e) {}
```

# JavaScript and AJAX

## Sending an asynchronous request via POST

```
xhr = new XMLHttpRequest();
// Stop any caching when using POST...
xhr.open('POST', 'ajax.php?r=' +
Math.random(), true);
// Set the content-type for POST
xhr.setRequestHeader("Content-
type","application/x-www-form-urlencoded");
xhr.send("first=fred&last=bloggs");

// Now need to define a function to call on
completion
```

# JavaScript and AJAX

**Callback function...**

```
xhr.onreadystatechange = function() {
   if (xhr.readyState == 4 && xhr.status == 200) {
      document.getElementById("txt").innerHTML =
xmlhttp.responseText; // Assumes text/plain or text/html
   }
};
```

**NOTE:** you should also text for other statuses, such as 30x, 40x and 50x that indicate failures.

# JavaScript and AJAX

- Exercise: make an asynchronous AJAX call
  - http://learn.cf.ac.uk/staff/semahd/eg3async-starter.html
  - Use the GET method
  - Request a file called http://learn.cf.ac.uk/staff/semahd/ajax.php
  - Use Math.random() to avoid browser caching
  - Send the request, with no data
  - Deal with the response in a callback function (replace the text in the div with the id 'replacedtext')

# JavaScript and AJAX

XMLHttpRequest methods

- .open(method,url,async)
  - Specifies the type of request, the URL, and if the request should be handled asynchronously or not.
- .setRequestHeader(string);
  - set http request header - for POST method
- .send(string)
  - Sends the request off to the server
  - String parameter is for POST only. For GET, you build the query string: xhr.open("GET","names.php?fname=Fred&lname=

# JavaScript and AJAX

XMLHttpRequest properties:

- responseXML
  - response from server for XML
- responseText
  - response from server for text (text/plain, text/html)