Web Programming C333-CS614-CS644

Dr Walid M. Aly

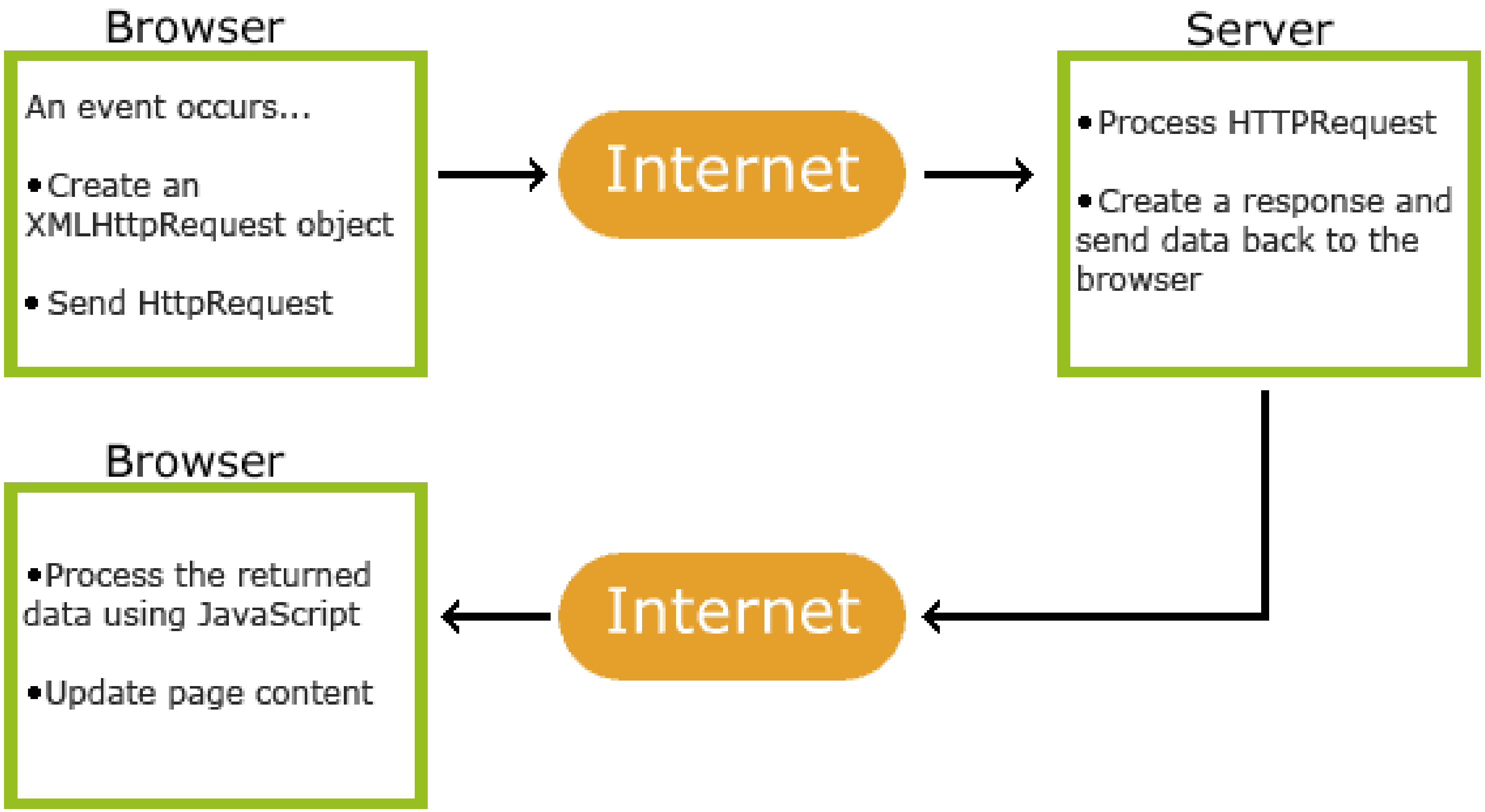
Lecture#11

Introduction to AJAX

- AJAX is an acronym for **A**synchronous JavaScript **A**nd **X**ML.
- AJAX is not a programming language. but simply a development technique for creating interactive web applications.
- The technology uses JavaScript to send and receive data between a web browser and a web server.
- The AJAX technique makes web pages more responsive by exchanging data with a server behind the scenes, instead of reloading an entire web page each time a user makes a change.
- With AJAX, web applications can be faster, more interactive, and more user friendly.
- AJAX was made popular in 2005 by Google, with Google Suggest.
- Google Suggest is using AJAX to create a very dynamic web interface: When you start typing in Google's search box, a JavaScript sends the letters off to a server and the server returns a list of suggestions.

How AJAX Works

- AJAX uses an **XMLHttpRequest** object to send data to a web server, and plain text /XML is commonly used as the format for receiving server data
- XMLHttpRequest is a misleading name. You don't have to understand XML to use AJAX.

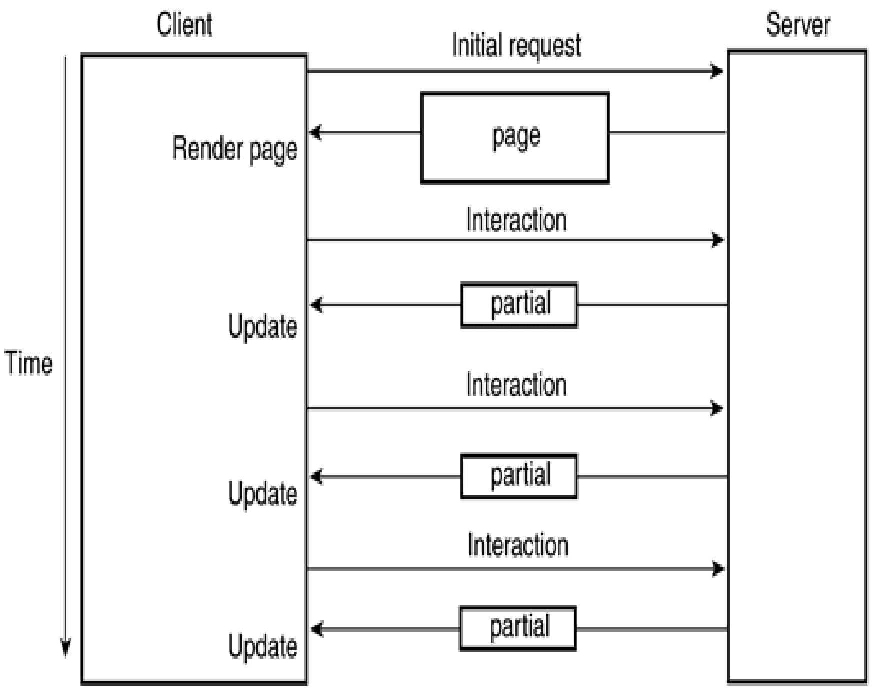

Browser
An event occurs...
• Create an XMLHttpRequest object
• Send HttpRequest
Internet
Server
• Process HTTPRequest
• Create a response and send data back to the browser
Internet
Browser
• Process the returned data using JavaScript
• Update page content

How AJAX Works

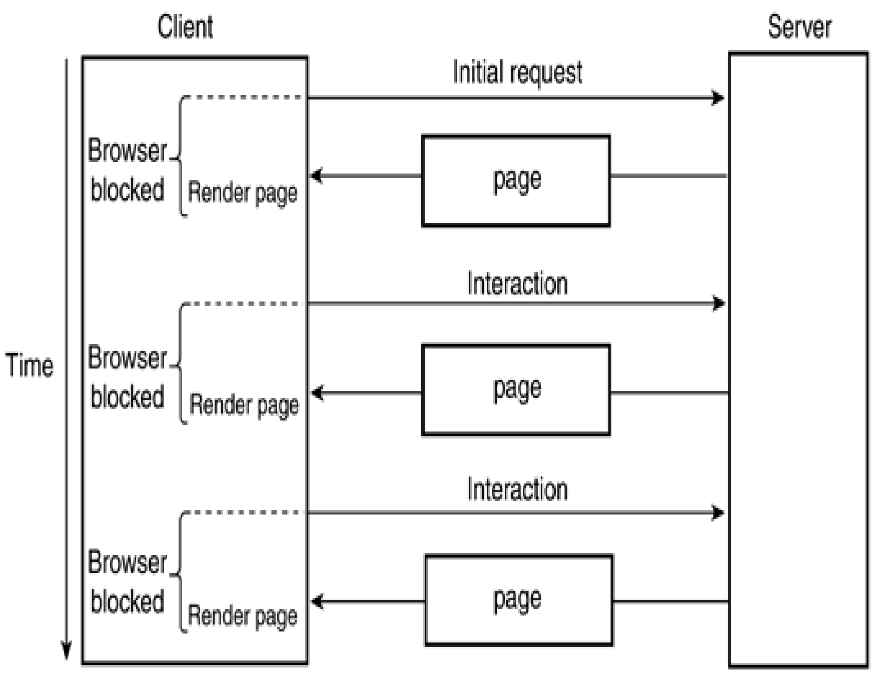
- Specific kind of Web applications that benefit from Ajax are those that have frequent interactions between the client and the server

- Goals are achieved with two different combined approaches:

1. Client requests are handled asynchronously
2. Only small parts of the current document are updated



Ajax session



Non-Ajax session

First Example on AJAX

```
<!DOCTYPE html> <html> <head>
<script>
function loadXMLDoc() {
var xmlhttp;
 xmlhttp=new XMLHttpRequest();  // Create an XMLHttpRequest Object
xmlhttp.onreadystatechange=function()
 {
 if (xmlhttp.readyState==4 && xmlhttp.status==200)
  {
  document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
  }
 }
xmlhttp.open("GET",“CCIT.txt",true);
xmlhttp.send();
}
</script>
</head>
<body>
<div id="myDiv"><h2>Let AJAX change this text</h2></div>
<button type="button" onclick="loadXMLDoc()">Change Content</button>
</body>
</html>
```

CCIT.txt

```
College of Computing and Information Technology
</br>
Arab Academy for Science, Technology and Maritme Transpor
```

http://127.0.0.1/ajax1/AjaxDemo1.html

XMLHttpRequest Object

AJAX - Create an XMLHttpRequest Object

The keystone of AJAX is the XMLHttpRequest object
Syntax for creating an XMLHttpRequest object:

```
variable=new XMLHttpRequest();
```

AJAX - Send a Request To a Server

- To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object
- If you want to send information with the GET method, add the information to the URL

```
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
```

Methods in XMLHttpRequest

Method	Description
open(*method,url,async*)	*method*: the type of request: GET or POST *url*: the location of the file on the server *async*: true (asynchronous) or false (synchronous)
send(*string*)	Sends the request off to the server. *string*: Only used for POST requests

```
xmlhttp.open("GET","ajax_test.asp",true);
Xmlhttp.send();
```

```
xhttp.open("POST", "ajax_test.asp", true);
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname=Henry&lname=Ford");
```

- ❑ The url - A File On a Server
    - ❑ The file can be any kind of file, like .txt and .xml, or server scripting files like .asp and .php (which can perform actions on the server before sending the response back).

AJAX - Server Response

Server Response

To get the response from a server, use the responseText or responseXML property of the XMLHttpRequest object.

Property	Description
responseText	get the response data as a string
responseXML	get the response data as XML data

```
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

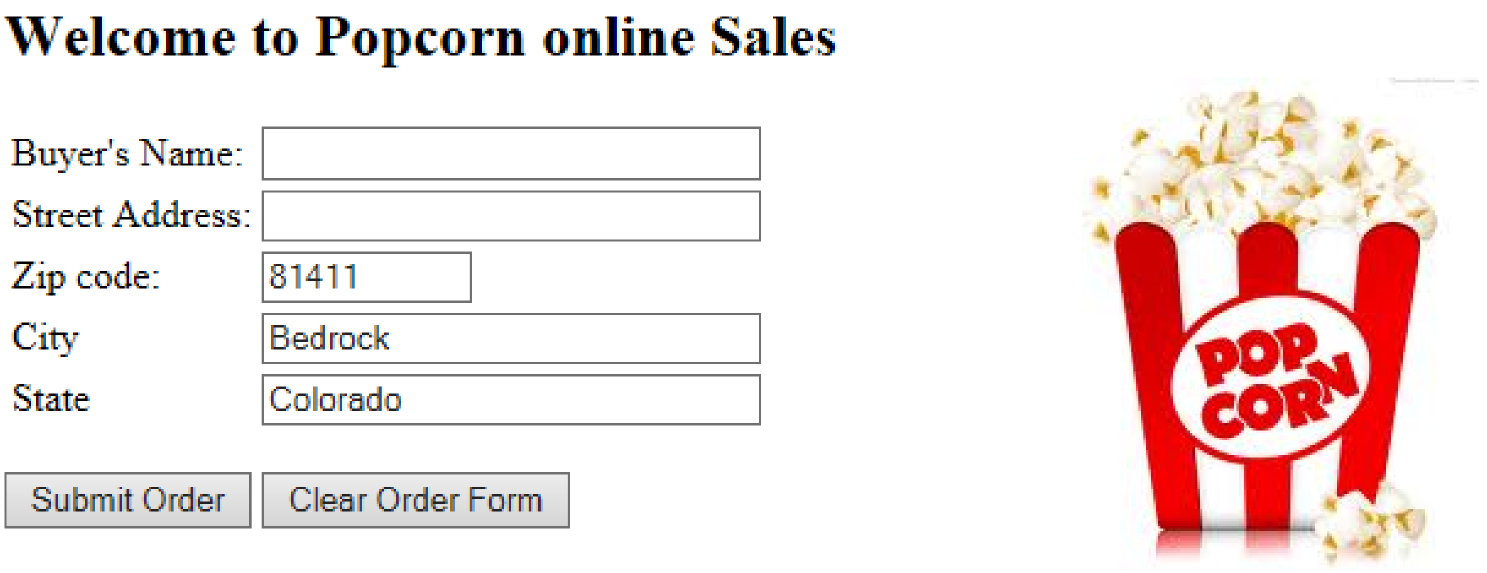
AJAX - The onreadystatechange Event

- When a request to a server is sent, we want to perform some actions based on the response.
- The onreadystatechange event is triggered every time the readyState changes.
- The readyState property holds the status of the XMLHttpRequest.

```
xmlhttp.onreadystatechange=function(){
  if (xmlhttp.readyState==4 && xmlhttp.status==200){
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
  }}
```

Property	Description
onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 404: Page not found

Example: AJAX with a response from php script



Welcome to Popcorn online Sales

Buyer's Name:	
Street Address:	
Zip code:	81411
City	Bedrock
State	Colorado

popcornA.html

http://127.0.0.1/ajax2/popcornA.html

Example: AJAX with a response from php script

- *The application*: Helps the user fill a form
    - The form gathers client information; asks for the zip code before the names of the city and state
    - As soon as the zip code is entered, the application sends a request to the server, which looks up the city and state for the given zip code and returns them to the form
    - Uses JavaScript to put the city and state names in the form
    - Uses PHP on the server to look up the city and state

- *The form*
    - Must reference the JavaScript code file in its head
    - Must register an event handler on the blur event of the zip code text box

```
<input type = "text“  onblur = "getPlace(this.value)" />
```

- Two functions are required by the application:
  1. **Request Phase (function),** in our case the `blur` handler
  2. **Response Phase (function).**

A function to handle the response, in our case recieveplace

The response function is what is called in the callbacks

```
xhr.onreadystatechange = receivePlace;
```

Request Phase (function)

the blur handler

```
function getPlace(zip) {
  var xhr = new XMLHttpRequest();

// Register the embedded handler function
  xhr.onreadystatechange = recievePlace//  will be edited further
  xhr.open("GET", "getCityState.php?zip=" + zip);
  xhr.send( );
}
```

The Response Document

- The response document is the document at server which is called by Ajax
- In our example the response document is the getCityState.php
- The response data is produced with a print statement

```
<?php
$cityState = array("81611" => "Aspen, Colorado",
             "81411" => "Bedrock, Colorado",
             "80908" => "Black Forest, Colorado",
             "80301" => "Boulder, Colorado",
             "81127" => "Chimney Rock, Colorado",
             "80901" => "Colorado Springs, Colorado",
             "81223" => "Cotopaxi, Colorado",
             "80201" => "Denver, Colorado",
             "81657" => "Vail, Colorado",
             "80435" => "Keystone, Colorado",
             "80536" => "Virginia Dale, Colorado"
             );
 $zip = $_GET["zip"];
 if (array_key_exists($zip, $cityState))
  print $cityState[$zip];
 else
  print " , ";
?>
```

getCityState.php

The Receiver Phase

- Actions of the receiver function:

1. **Put all actions in the then clause of a selector that checks to see if readyState is 4**
2. **Get the response value from the responseTextn property of the XHR object**
3. **Split it into its two parts**
4. **Set the values of the city and state text boxes**

```
// popcornA.js
function getPlace(zip) {
  var xhr = new XMLHttpRequest();

// Register the embedded handler function
 //xhr.onreadystatechange = recievePlace
  xhr.onreadystatechange = function () {
    if (xhr.readyState == 4 && xhr.status == 200) {
      var result = xhr.responseText;
      var place = result.split(', ');
      if (document.getElementById("city").value == "")
        document.getElementById("city").value = place[0];
      if (document.getElementById("state").value == "")
        document.getElementById("state").value = place[1];
    }
  }
  xhr.open("GET", "getCityState.php?zip=" + zip,true);
  xhr.send(null);
}
```