AJAX

What is AJAX ?

- Asynchronous Javascript and XML.
- Not a stand-alone language or technology.
- It is a technique that combines a set of known technologies in order to create faster and more user friendly web pages.
- It is a client side technology.

Purpose of AJAX

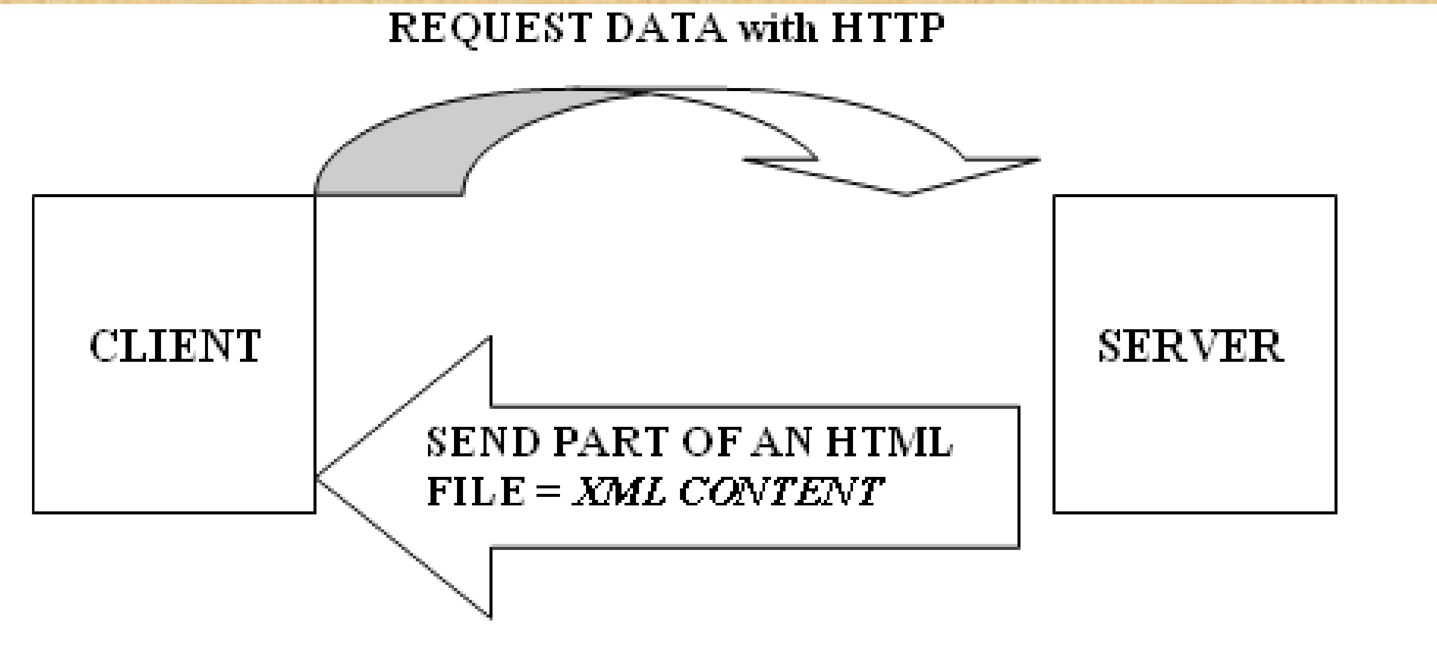
- Prevents unnecessary reloading of a page.
- When we submit a form, although most of the page remains the same, whole page is reloaded from the server.
- This causes very long waiting times and waste of bandwidth.
- AJAX aims at loading only the necessary innformation, and making only the necessary changes on the current page without reloading the whole page.

Technologies Used

- **AJAX uses:**
  - **<u>Javascript</u> (for altering the page)**
  - **<u>XML</u> (for information exchange)**
  - **<u>ASP or JSP</u> (server side)**

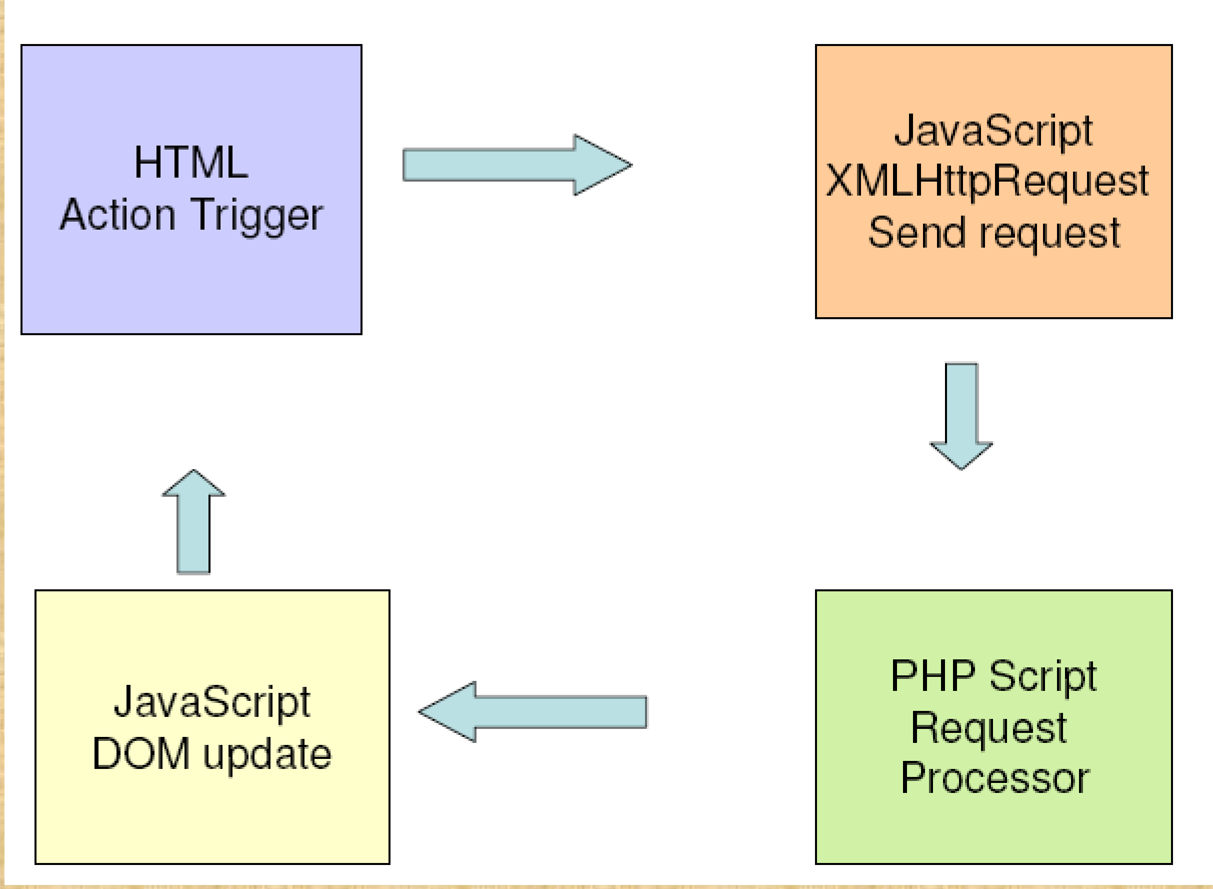
Data Exchange in AJAX

- **In AJAX:**



Data Exchange in AJAX

PHP and AJAX



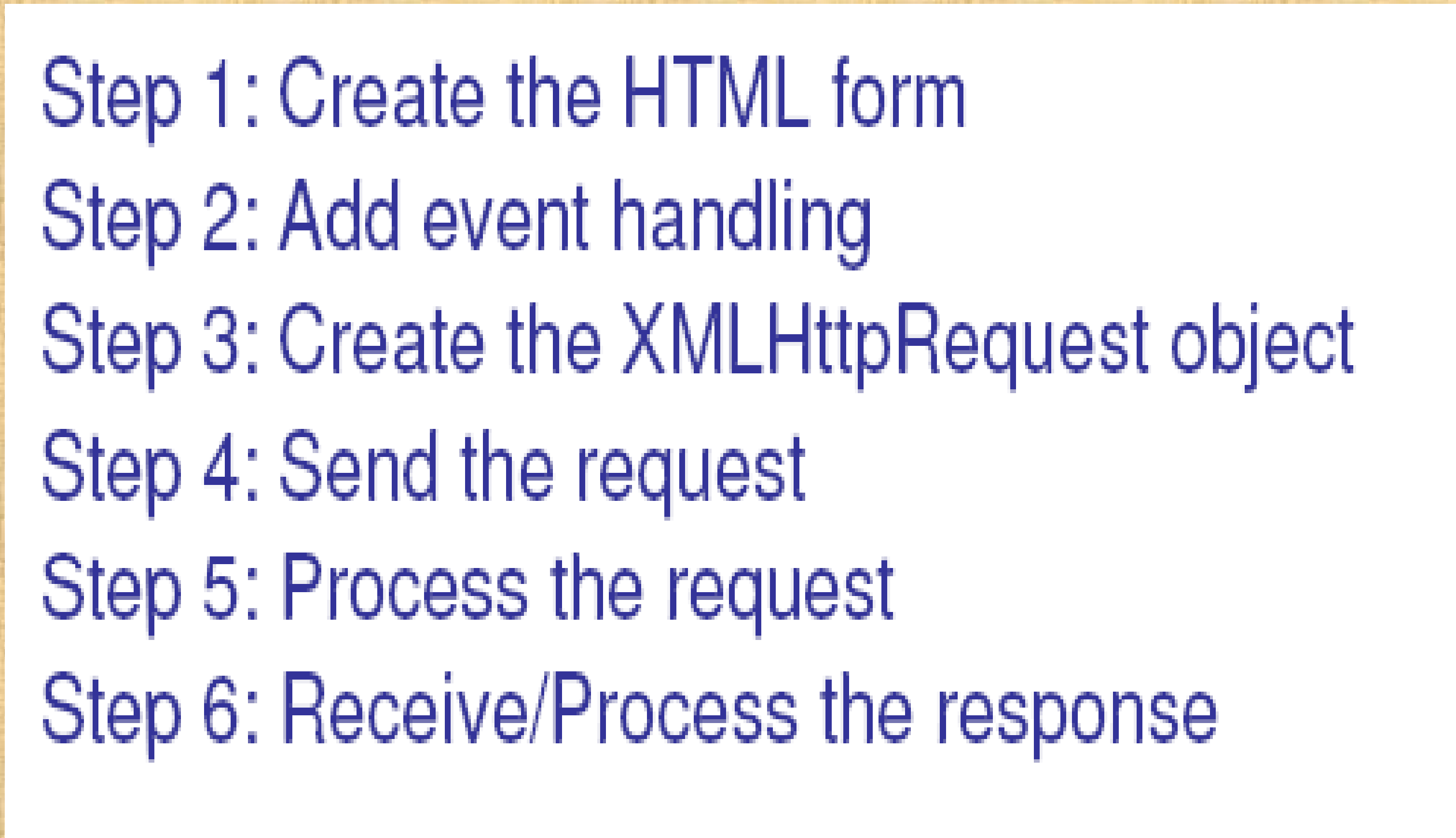
HTML
Action Trigger

JavaScript
XMLHttpRequest
Send request

PHP Script
Request
Processor

JavaScript
DOM update

Web application using AJAX and PHP



Step 1: Create the HTML form

Step 2: Add event handling

Step 3: Create the XMLHttpRequest object

Step 4: Send the request

Step 5: Process the request

Step 6: Receive/Process the response

XMLHTTP Object

- **The object that is used to connect to the remote server is called <u>XMLHTTP</u>.**
- **It resembles the Java’s URL object that is used to access a specific URL and get the contents.**
- **<u>For IE 5.5:</u>**

  objXmlHttp=new ActiveXObject(“Microsoft.XMLHTTP”)
- **<u>For Mozilla:</u>**

  objXmlHttp=new XMLHttpRequest()

Sending information

- **After creating the object, we can send information to the web server and get the answer using this object's functions:**
- **GET METHOD:** ***xmlhttp.open("GET", url, true)***

  xmlhttp.send()
- **POST METHOD:** ***xmlhttp.open("POST", url, true)***

  xmlhttp.send("date=11-11-2006&name=Ali")

- **Third argument tells that data will be processes asynchronously. When server responds, the "OnReadyStateChange" event handler will be called.**

OnReadyStateChange

- myRequest.open(“GET”, url, true);// true ➔ asynchronous request (default)
- myRequest.onreadystatechange = responseAjax();

/* onreadystatechange determines which handler will be called when ready state changes */

- myRequest.send(null);// sends the request
- The readystate property of our XMLHTTPRequest object can have the following values:
- 0 ➔ uninitialized
- 1 ➔ loading
- 2 ➔ loaded
- 3 ➔ interactive
- 4 ➔ completed

XMLHttpRequest Object

- Methods:

  abort() - stop the current request

  getAllResponseHeaders - Returns complete set of headers (labels and values) as a string

  getResponseHeader(:headerLabel”) – returns the string value of the requested header field

  open(“method”,”URL”) sets a pending request

  send(content) – transmits the request

  setRequestHeader(“label”,”value”) – sets label/value in the header

Contd…

- Properties

  onreadystatechange - event handler to use

  readyState (0-uninitialized, 1-loading, 2-loaded, 3-interactive, 4- complete)

  responseText – string version of the data returned

  responseXML – DOM compatible document object returned by server

  status – http response header code (200 – good, 400-bad rquest,404-not found,500-Internal server error)

  statusText – string message of status code

Retrieving information

- **We get the returned value with the property “<u>xmlHttp.responseText</u>”.**

Using DOM with javascript

- var results = xmlhttp.responseText.split(",");
- document.getElementById('city').value = results[0];
- document.getElementById('state').value = results[1];

example

```
var url="servertime.php"
xmlHttp.onreadystatechange=stateChanged
xmlHttp.open("GET",url,true)
xmlHttp.send(null)
```

stateChanged

```
function stateChanged()
{
   if (xmlHttp.readyState==4 || xmlHttp.readyState=="complete")
     {
       //update the DOM with the data
       document.getElementById("time").innerHTML=xmlHttp.responseText
     }
}
```

Popular Ajax Frameworks

- Prototype
  - http://www.prototypejs.org/
  - free
- Script.aculo.us
  - http://script.aculo.us/
  - Used with the Prototype Framework, mainly for animations and interface development
  - free
- Backbase
  - Enterprise Ajax Framework
  - not free

Pros/Cons

- **Advantages:**
  - **Independent of server technology.**
  - **Apart from obtaining the XMLHTTP object, all processing is same for all browser types, because Javascript is used.**
  - **Permits the development of faster and more interactive web applications.**
- **Disadvantages:**
  - **The back button problem. People think that when they press back button, they will return to the last change they made, but in AJAX this doesn not hold.**
  - **Possible network latency problems. People should be given feedback about the processing.**
  - **Does not run on all browsers.**