XML AND AJAX

BY Lecturer Suraj Pandey CTC College

XML AND AJAX

1.1 Introduction to XML

1.2 Working with Basics of XML

1.3 Converting XML Documents in other formats

1.4 Working with XSLT

1.5 Working with Xpath, Xlink and Xpointer

1.6 XML Application

1.7 Overview of AJAX

1.8 AJAX components

1.9 Asynchronous Data Transfer with XML Http request

1.10 Implementing AJAX Frameworks

1.11 Consuming web services using AJAX

INTRODUCTION TO XML

Introduction

XML, or Extensible Markup Language, is a markup language that you can use to create your own tags. It was created by the World Wide Web Consortium (W3C) to overcome the limitations of HTML, the Hypertext Markup Language that is the basis for all Web pages. Like HTML, XML is based on SGML -- Standard Generalized Markup Language.

WHY DO WE NEED XML?

HTML is the most successful markup language of all time. You can view the simplest HTML tags on virtually any device, from palmtops to mainframes, and you can even convert HTML markup into voice and other formats with the right tools.

XML and HTML were designed with different goals:

1. XML is designed to carry data emphasizing on what type of data it is.
2. HTML is designed to display data emphasizing on how data looks
3. XML tags are not predefined like HTML tags.
4. HTML is a markup language whereas XML provides a framework for defining markup languages.
5. HTML is about displaying data,hence it is static whereas XML is about carrying information,which makes it dynamic.

There are three important characteristics of XML that make it useful in a variety of systems and solutions −

XML is extensible − XML allows you to create your own self-descriptive tags, or language, that suits your application.

XML carries the data, does not present it − XML allows you to store the data irrespective of how it will be presented.

XML is a public standard − XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML FEATURES

- Here are some important features of XML:
- It is extensible and human-readable.
- It is platform and language independent.
- It preserves white space.
- Overall simplicity.
- Self-descriptive nature.
- It separates data from HTML.
- XML tags are not predefined. You need to define your customized tags.
- XML was designed to carry data, not to display that data.
- Mark-up code of XML is easy to understand for a human.
- Well-structured format is easy to read and write from programs.
- XML is an extensible markup language like HTML.

XML USAGE

A short list of XML usage says it all −

1. XML can work behind the scene to simplify the creation of HTML documents for large web sites.
2. XML can be used to exchange the information between organizations and systems.
3. XML can be used for offloading and reloading of databases.
4. XML can be used to store and arrange the data, which can customize your data handling needs.
5. XML can easily be merged with style sheets to create almost any desired output.

XML DOES NOT USE PREDEFINED TAGS

The XML language has no predefined tags.

The tags in the example above (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document.

HTML works with predefined tags like <p>, <h1>, <table>, etc.

With XML, the author must define both the tags and the document structure.

XML IS EXTENSIBLE

Most XML applications will work as expected even if new data is added (or removed).

Imagine an application designed to display the original version of note.xml (<to> <from> <heading> <body>).

Then imagine a newer version of note.xml with added <date> and <hour> elements, and a removed <heading>.

The way XML is constructed, older version of the application can still work:

```
<note>
  <date>2015-09-01</date>
  <hour>08:30</hour>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```

- XML Separates Data from HTML

When displaying data in HTML, you should not have to edit the HTML file when the data changes.

With XML, the data can be stored in separate XML files.

With a few lines of JavaScript code, you can read an XML file and update the data content of any HTML page.

Books.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>

  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>

  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>

</bookstore>
```

- **History of XML**
- XML started way back in 1996 and was first published in 1998. **W**orld **W**ide **W**eb **C**onsortium (W3C) is the developer of **XML**, and it became a **W3C recommendation** in 1998.
- There are two versions of XML.
- XML 1.0
- XML 1.1
- **XML 1.1** is the latest version. Yet, **XML 1.0** is the most used version.

- **XML Encoding**
- Encoding is the conversion of Unicode characters to their binary representation. UTF is use for XML encoding. **UTF** stands for **U**CS (**UCS** stands for **U**niversal **C**haracter **S**et) **T**ransformation **F**ormat.
- Mainly, there are two types of UTF encoding.
- UTF-8 : UTF-8 uses 8-bits to represent the characters.
- Example:
- <?xml version="1.0" encoding="UTF-8"?>

- UTF-16
- It uses 16-bits to represent the characters.
- Example:
- **<?xml version="1.0" encoding="UTF-16"?>** You can use encoding inside the XML declaration. UTF-8 is the default encoding in XML.

- **XML Syntax**
- The below code segment shows the basic XML syntax.
- **<?xml version = "1.0" encoding = "UTF-8" ?>**

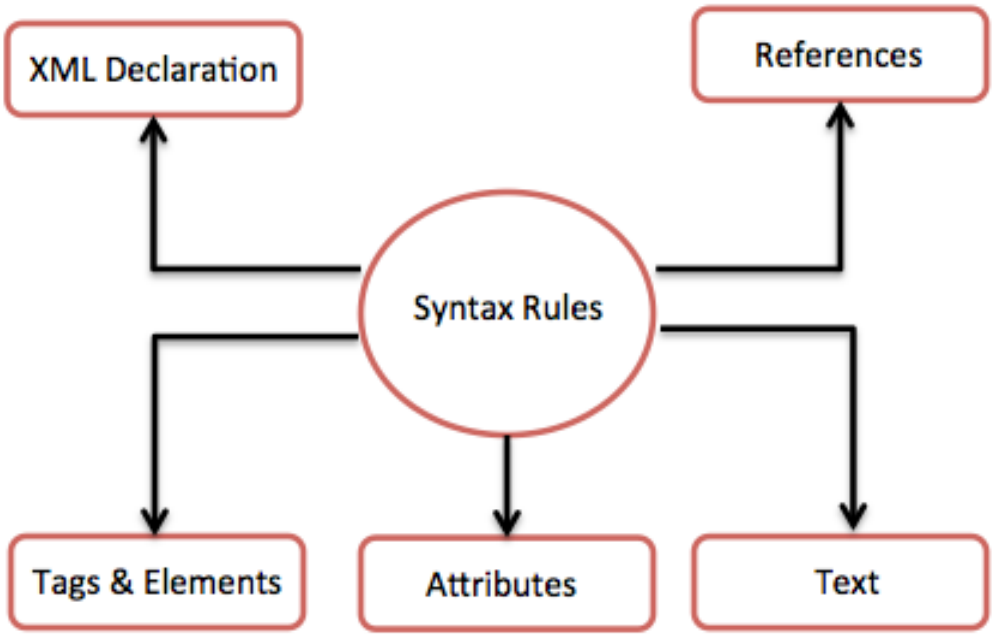
```
<root>
    <child>
        <subchild>.....</subchild>
    </child>
</root>
```

IS XML A PROGRAMMING LANGUAGE?

A programming language consists of grammar rules and its own vocabulary which is used to create computer programs. These programs instruct the computer to perform specific tasks. XML does not qualify to be a programming language as it does not perform any computation or algorithms. It is usually stored in a simple text file and is processed by special software that is capable of interpreting XML.

XML - SYNTAX

The following diagram depicts the syntax rules to write different types of markup and text in an XML document.



XML DECLARATION

The XML document can optionally have an XML declaration. It is written as follows −

<?xml version = "1.0" encoding = "UTF-8"?>Where *version* is the XML version and *encoding* specifies the character encoding used in the document.

Syntax Rules for XML Declaration

The XML declaration is case sensitive and must begin with "**<?xml>**" where "**xml**" is written in lower-case.

TAGS AND ELEMENTS

An XML file is structured by several XML-elements, also called XML-nodes or XML-tags. The names of XML-elements are enclosed in triangular brackets < > as shown below −

<element>Syntax Rules for Tags and Elements

Element Syntax − Each XML-element needs to be closed either with start or with end elements as shown below −

<element>....</element>or in simple-cases, just this way −

<element/>

XML ATTRIBUTES

An **attribute** specifies a single property for the element, using a name/value pair. An XML-element can have one or more attributes. For example −

CRIMSON! Here **href** is the attribute name and **http://www.CCT.COM/** is attribute value.

Syntax Rules for XML Attributes

Attribute names in XML (unlike HTML) are case sensitive. That is, *HREF* and *href* are considered two different XML attributes.

XML REFERENCES

References usually allow you to add or include additional text or markup in an XML document. References always begin with the symbol **"&"** which is a reserved character and end with the symbol **";".** XML has two types of references −

Entity References − An entity reference contains a name between the start and the end delimiters. For example **&** where *amp* is *name*. The *name* refers to a predefined string of text and/or markup.

Character References − These contain references, such as **A**, contains a hash mark (“#”) followed by a number. The number always refers to the Unicode code of a character. In this case, 65 refers to alphabet "A".

XML TEXT

The names of XML-elements and XML-attributes are case-sensitive, which means the name of start and end elements need to be written in the same case. To avoid character encoding problems, all XML files should be saved as Unicode UTF-8 or UTF-16 files.

Whitespace characters like blanks, tabs and line-breaks between XML-elements and between the XML-attributes will be ignored.

- **XML Comments**
- Comments are optional. Adding comments help to understand the document content.
- **Syntax for XML Comments**
- A comment begins with **<!–** and ends with **–>**.
- Following code segment shows the syntax for XML comments.
- **<!-- Add your comment here -->**

HTML	XML
HTML stands for Hyper Text Markup Language.	XML stands for extensible Markup Language.
HTML is static.	XML is dynamic.
HTML is a markup language.	XML provides framework to define markup languages.
HTML can ignore small errors.	XML does not allow errors.
HTML is not Case sensitive.	XML is Case sensitive.
HTML tags are predefined tags.	XML tags are user defined tags.
There are limited number of tags in HTML.	XML tags are extensible.
HTML does not preserve white	White space can be preserved in XML.

HTML tags are used for displaying the data.	XML tags are used for describing the data not for displaying.
In HTML, closing tags are not necessary.	In XML, closing tags are necessary.
HTML is used to display the data.	XML is used to store data.
HTML does not carry data it just display it.	XML carries the data to and from database.

HOW TO CONVERT AN XML FILE

The best solution to converting an XML file to another format is to use one of the editors . The program that's creating the XML file is more than likely able to save the same file to a different format.

For example, a simple text editor, which can open a text document like XML, can usually save the file to another text-based format like TXT.

Here are some other free online XML converters that might be more useful for you:

XML to HTML

XML to CSV

XML to XSD

XML to PDF

Here are some free converters that convert *to* XML instead of *from* XML:

XLS/XLSX to XML

SQL to XML

CSV to XML

JSON to XML

WORKING WITH XSLT

XSL

XSL which stands for E**X**tensible **S**tylesheet **L**anguage. It is similar to XML as CSS is to HTML.

Need for XSL

In case of HTML document, tags are predefined such as table, div, and span; and the browser knows how to add style to them and display those using CSS styles. But in case of XML documents, tags are not predefined. In order to understand and style an XML document, World Wide Web Consortium (W3C) developed XSL which can act as XML based Stylesheet Language. An XSL document specifies how a browser should render an XML document.

Following are the main parts of XSL −

XSLT − used to transform XML document into various other types of document.

XPath − used to navigate XML document.

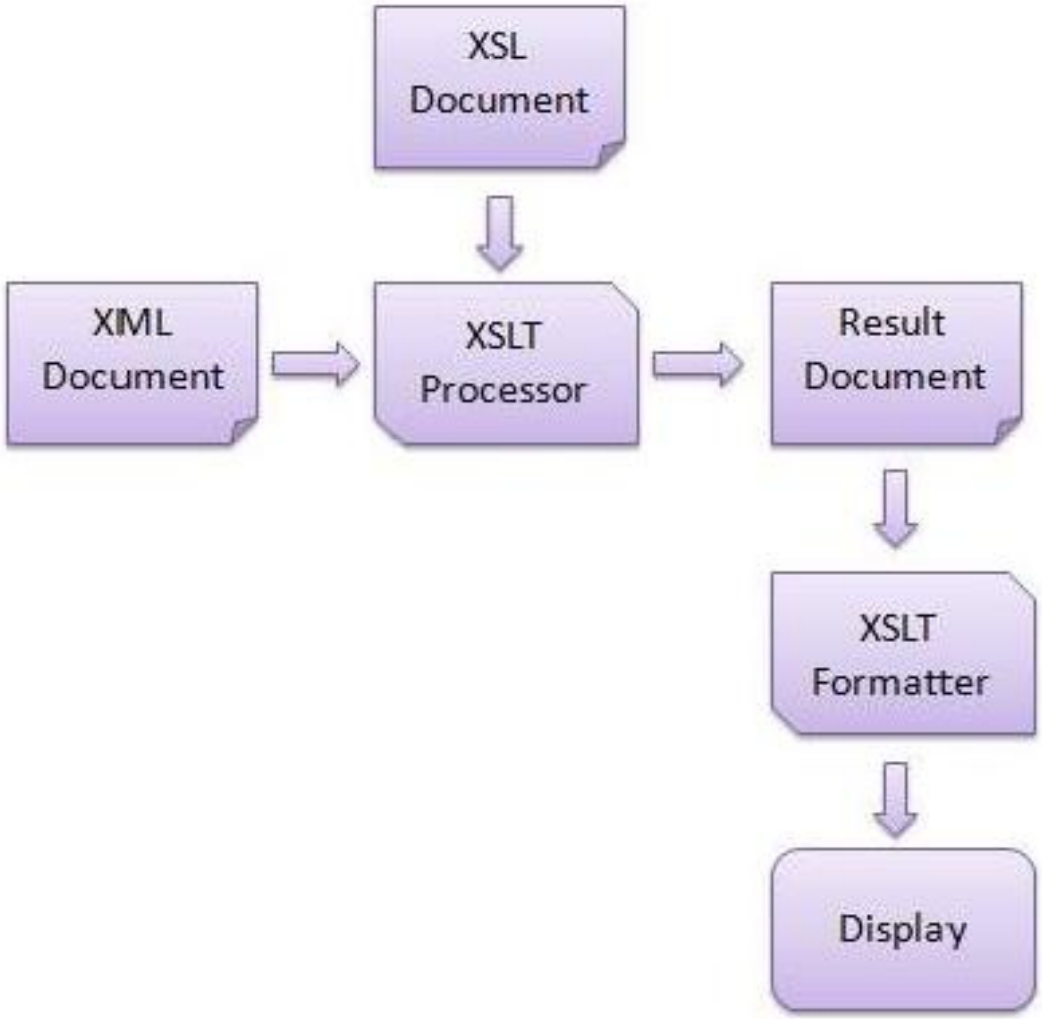
XSL-FO − used to format XML document.

WHAT IS XSLT

XSLT, Extensible Stylesheet Language Transformations, provides the ability to transform XML data from one format to another automatically.

How XSLT Works

An XSLT stylesheet is used to define the transformation rules to be applied on the target XML document. XSLT stylesheet is written in XML format. XSLT Processor takes the XSLT stylesheet and applies the transformation rules on the target XML document and then it generates a formatted document in the form of XML, HTML, or text format. This formatted document is then utilized by XSLT formatter to generate the actual output which is to be displayed to the end-user.


XSL
Document
XML
Document
XSLT
Processor
Result
Document
XSLT
Formatter
Display

Advantages

Here are the advantages of using XSLT −

-Independent of programming. Transformations are written in a separate xsl file which is again an XML document.

-Output can be altered by simply modifying the transformations in xsl file. No need to change any code. So Web designers can edit the stylesheet and can see the change in the output quickly.

XSLT SYNTAX

Let’s suppose we have the following sample XML file, students.xml, which is required to be transformed into a well-formatted HTML document.

students.xml

```
<?xml version = "1.0"?>
 <class>
 <student rollno = "393"> <firstname>Dinkar</firstname>
<lastname>Kad</lastname> <nickname>Dinkar</nickname>
<marks>85</marks> </student>
<student rollno = "493"> <firstname>Vaneet</firstname>
<lastname>Gupta</lastname> <nickname>Vinni</nickname>
<marks>95</marks> </student>
<student rollno = "593"> <firstname>Jasvir</firstname>
<lastname>Singh</lastname> <nickname>Jazz</nickname>
<marks>90</marks>
 </student>
 </class>
```

We need to define an XSLT style sheet document for the above XML document to meet the following criteria −

Page should have a title **Students**.

Page should have a table of student details.

Columns should have following headers: Roll No, First Name, Last Name, Nick Name, Marks

Table must contain details of the students accordingly.

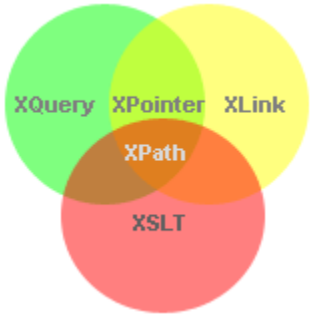
Step 1: Create XSLT document

Create an XSLT document to meet the above requirements, name it as students.xsl and save it in the same location where students.xml lies.

WHAT IS XPATH?

XPath is a major element in the XSLT standard.

XPath can be used to navigate through elements and attributes in an XML document.



- XPath stands for XML Path Language
- XPath uses "path like" syntax to identify and navigate nodes in an XML document
- XPath contains over 200 built-in functions
- XPath is a major element in the XSLT standard
- XPath is a W3C recommendation

There are various types of nodes, including element nodes, attribute nodes, and text nodes. XPath defines a way to compute a string-value for each type of node.

XPath defines a library of standard functions for working with strings, numbers and boolean expressions.

Examples:

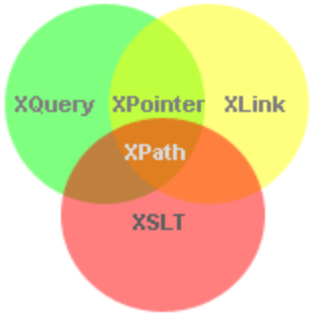
child::* - Selects all children of the root node.

.//name - Selects all elements having the name "name", descendants of the current node.

/catalog/cd[price>10.80] - Selects all the cd elements that have a price element with a value larger than 10.80.

XLINK

XLink is used to create hyperlinks in XML documents.



- XLink is used to create hyperlinks within XML documents
- Any element in an XML document can behave as a link
- With XLink, the links can be defined outside the linked files
- XLink is a W3C Recommendation

```
<?xml version="1.0" encoding="UTF-8"?>

<homepages xmlns:xlink="http://www.w3.org/1999/xlink">
  <homepage xlink:type="simple" xlink:href="https://www.w3schools.com">Visit W3Schools</homepage>
  <homepage xlink:type="simple" xlink:href="http://www.w3.org">Visit W3C</homepage>
</homepages>
```

To get access to the XLink features we must declare the XLink namespace. The XLink namespace is: "http://www.w3.org/1999/xlink".

The xlink:type and the xlink:href attributes in the <homepage> elements come from the XLink namespace.

The xlink:type="simple" creates a simple "HTML-like" link (means "click here to go there").

The xlink:href attribute specifies the URL to link to.

XLink Example

The following XML document contains XLink features:

```
<?xml version="1.0" encoding="UTF-8"?>

<bookstore xmlns:xlink="http://www.w3.org/1999/xlink">

<book title="Harry Potter">
  <description
  xlink:type="simple"
  xlink:href="/images/HPotter.gif"
  xlink:show="new">
  As his fifth year at Hogwarts School of Witchcraft and
  Wizardry approaches, 15-year-old Harry Potter is.......
  </description>
</book>

<book title="XQuery Kick Start">
  <description
  xlink:type="simple"
  xlink:href="/images/XQuery.gif"
  xlink:show="new">
  XQuery Kick Start delivers a concise introduction
  to the XQuery standard.......
  </description>
</book>

</bookstore>
```

Example explained:

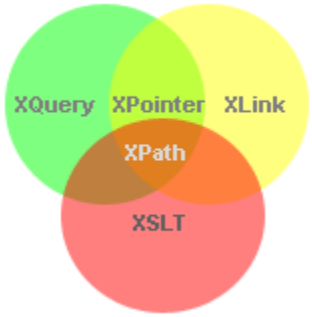
The XLink namespace is declared at the top of the document (xmlns:xlink="http://www.w3.org/1999/xlink")

The xlink:type="simple" creates a simple "HTML-like" link

The xlink:href attribute specifies the URL to link to (in this case - an image)

The xlink:show="new" specifies that the link should open in a new window

XPOINTER



- XPointer allows links to point to specific parts of an XML document
- XPointer uses XPath expressions to navigate in the XML document
- XPointer is a W3C Recommendation

XPOINTER BROWSER SUPPORT

There is no browser support for XPointer. But XPointer is used in other XML languages.

XPointer Example

In this example, we will use XPointer in conjunction with XLink to point to a specific part of another document.

We will start by looking at the target XML document (the document we are linking to):

```xml
<?xml version="1.0" encoding="UTF-8"?>

<dogbreeds>

<dog breed="Rottweiler" id="Rottweiler">
  <picture url="https://dog.com/rottweiler.gif" />
  <history>The Rottweiler's ancestors were probably Roman
  drover dogs.....</history>
  <temperament>Confident, bold, alert and imposing, the Rottweiler
  is a popular choice for its ability to protect....</temperament>
</dog>

<dog breed="FCRetriever" id="FCRetriever">
  <picture url="https://dog.com/fcretriever.gif" />
  <history>One of the earliest uses of retrieving dogs was to
  help fishermen retrieve fish from the water....</history>
  <temperament>The flat-coated retriever is a sweet, exuberant,
  lively dog that loves to play and retrieve....</temperament>
</dog>

</dogbreeds>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>

<mydogs xmlns:xlink="http://www.w3.org/1999/xlink">

<mydog>
  <description>
  Anton is my favorite dog. He has won a lot of.....
  </description>
  <fact xlink:type="simple" xlink:href="https://dog.com/dogbreeds.xml#Rottweiler">
  Fact about Rottweiler
  </fact>
</mydog>

<mydog>
  <description>
  Pluto is the sweetest dog on earth......
  </description>
  <fact xlink:type="simple" xlink:href="https://dog.com/dogbreeds.xml#FCRetriever">
  Fact about flat-coated Retriever
  </fact>
</mydog>

</mydogs>
```

WHAT IS AJAX?

AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.

With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

AJAX is Based on Open Standards

AJAX is based on the following open standards −

Browser-based presentation using HTML and Cascading Style Sheets (CSS).

Data is stored in XML format and fetched from the server.

Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.

JavaScript to make everything happen.

THE ORIGINS OF AJAX

The key technical components of AJAX are:

- XHTML – a stricter, cleaner rendering of HTML into XML.
- CSS for marking up and adding styles.
- The Javascript Document Object Model (DOM) which allows the content, structure and style of a document to be dynamically accessed and updated.
- The XMLHttpRequest object which exchanges data asynchronously with the Web server reducing the need to continually fetch resources from the server.

Since data can be sent and retrieved without requiring the user to reload an entire Web page, small amounts of data can be transferred as and when required. Moreover, page elements can be dynamically refreshed at any level of granularity to reflect this. An AJAX application performs in a similar way to local applications residing on a user's machine, resulting in a user experience that may differ from traditional Web browsing.

Examples of AJAX usage include GMail and Flickr. It is largely due to these and other prominent sites that AJAX has become popular only relatively recently – the technology has been available for some time. One precursor was dynamic HTML (DHTML), which twinned HTML with CSS and JavaScript but suffered from cross-browser compatibility issues.

AJAX is not a technology, rather, the term refers to a proposed set of methods using a number of existing technologies. As yet, there is no firm AJAX standard, although the recent establishment of the Open AJAX Alliance [2], supported by major industry figures such as IBM and Google, suggests that one will become available soon.

ADVANTAGES AND DISADVANTAGES OF AJAX

1. State can be maintained throughout a Web site.
2. A Web application can request only the content that needs to be updated, thus drastically reducing bandwidth usage and load time.
3. Users may perceive an AJAX-enabled application to be faster or more responsive.
4. Use of Ajax can reduce connections to the server, since scripts and style sheets only have to be requested once.

The disadvantages include:

1. Clicking the browser's "back" button may not function as expected.
2. Dynamic Web page updates make it difficult for a user to use bookmarks.
3. Browser does not support JavaScript or have JavaScript disabled, will not be able to use its functionality.
4. AJAX may provide a mechanism for attacks by malicious code

AJAX - EXAMPLES

Here is a list of some famous web applications that make use of AJAX.

Google Maps

A user can drag an entire map by using the mouse, rather than clicking on a button.

https://maps.google.com/

Google Suggest

As you type, Google offers suggestions. Use the arrow keys to navigate the results.

https://www.google.com/webhp?complete=1&hl=en

Gmail

Gmail is a webmail built on the idea that emails can be more intuitive, efficient, and useful.

https://gmail.com/

Yahoo Maps (new)

Now it's even easier and more fun to get where you're going!

https://maps.yahoo.com/

IMPLEMENTING AJAX FRAMEWORK

The AJAX Framework is a cross browser framework that allows developers to quickly develop web pages that can call web services, web pages and other types of content through JavaScript without having to submit the current page.The AJAX Framework is:

Easy to use.

Works well with other existing frameworks.

Supports both "GET" and "POST".

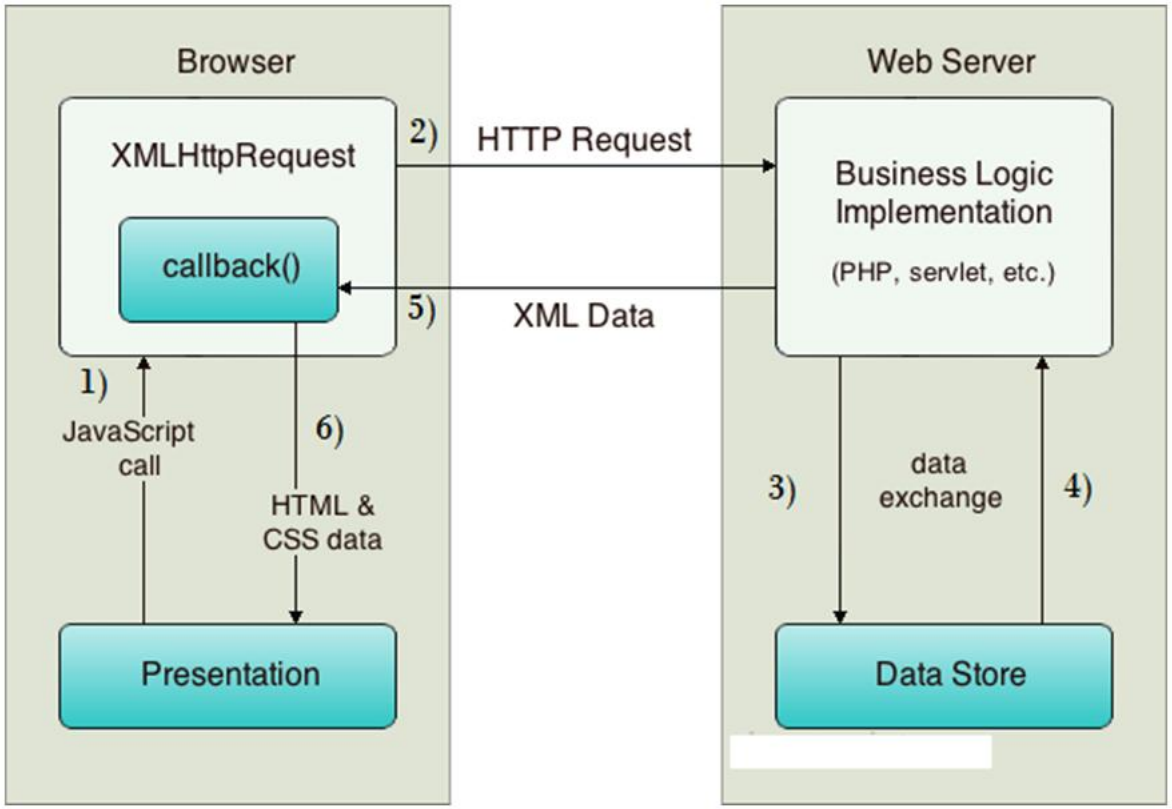
Works with Internet Explorer 6+, Opera 8+, Safari 3+, Firefox 2+, Google's Chrome and other Mozilla based browsers.

AJAX

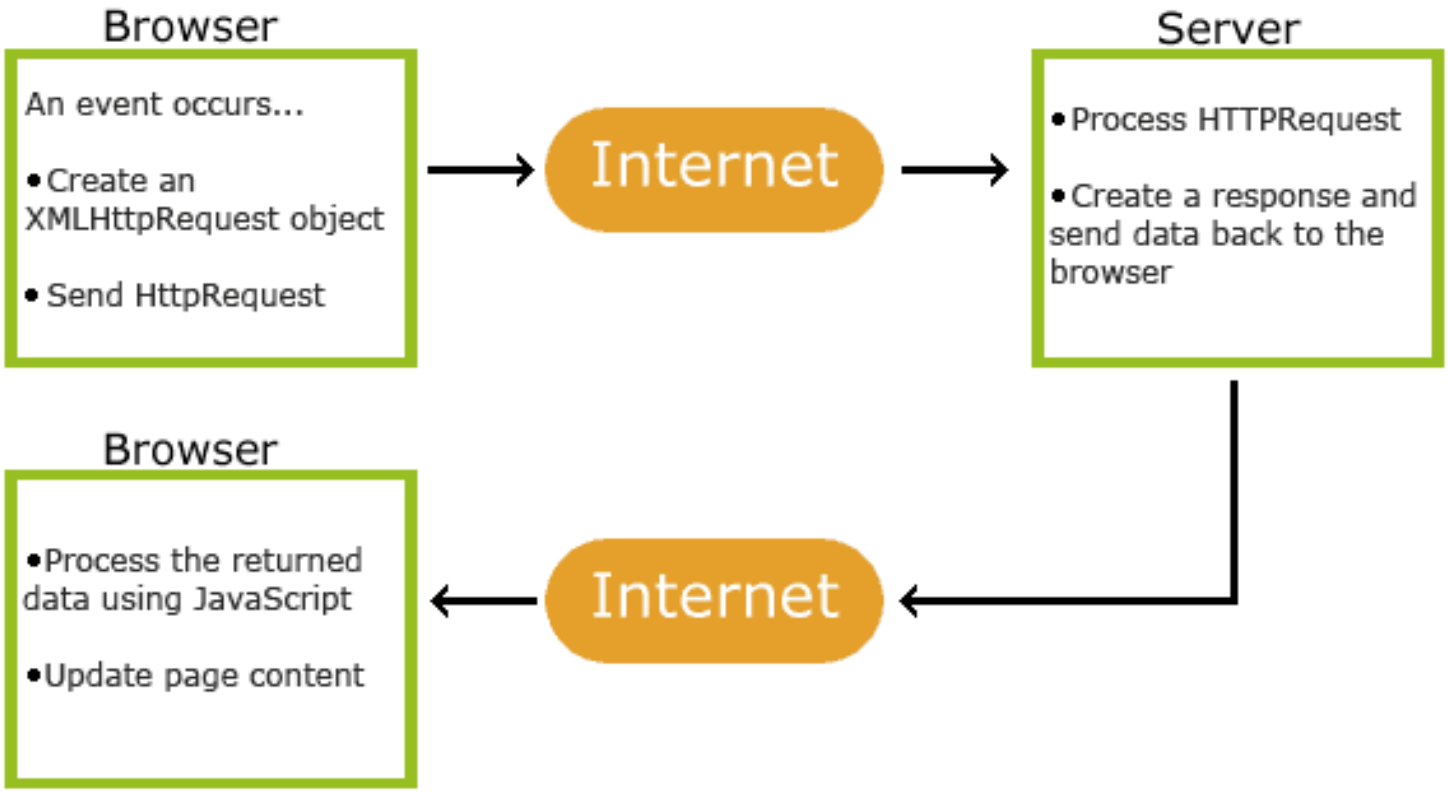
BY LECTURER SURAJ PANDEY CCT COLLEGE

WHAT IS AJAX?

AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.


Browser
XMLHttpRequest
callback()
Presentation
Web Server
Business Logic Implementation
(PHP, servlet, etc.)
Data Store
2)
HTTP Request
5)
XML Data
1)
JavaScript call
6)
HTML & CSS data
3)
data exchange
4)

How AJAX Works



As you can see in the above example, XMLHttpRequest object plays a important role.

1. User sends a request from the UI and a javascript call goes to XMLHttpRequest object.
2. HTTP Request is sent to the server by XMLHttpRequest object.
3. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
4. Data is retrieved.
5. Server sends XML data or JSON data to the XMLHttpRequest callback function.
6. HTML and CSS data is displayed on the browser.

Ajax behind Process (How to Work Ajax)

HTML/CSS

HTML/CSS is website markup language for defining web page layout, such as fonts style and colors.

JavaScript

JavaScript is a web scripting language. JavaScript special object XMLHttpRequest that was designed by Microsoft. XMLHttpRequest provides an easy way to retrieve data from web server without having to do full page refresh. Web page can update just part of the page without interrupting what the users are doing.

Document Object Model

Document Object Model (DOM) method provides a tree structure as a logical view of web page.

XML

XML is a format for retrieve any type of data, not just XML data from the web server. However you can use other formats such as Plain text, HTML or JSON (JavaScript Object Notation). and it supports protocols HTTP and FTP.

ADVANTAGES OF AJAX

AJAX Concept
Before you starting AJAX you'll need to have a strong knowledge of JavaScript. AJAX is not a difficult, you can easy implement AJAX in a meaningful manner. Some IDE are help us to implement AJAX.

Speed
Reduce the server traffic in both side request. Also reducing the time consuming on both side response.

Interaction
AJAX is much responsive, whole page(small amount of) data transfer at a time.

XMLHttpRequest
XMLHttpRequest has an important role in the Ajax web development technique. XMLHttpRequest is special JavaScript object that was designed by Microsoft. XMLHttpRequest object call as a asynchronous HTTP request to the Server for transferring data both side. It's used for making requests to the non-Ajax pages.

Asynchronous calls
AJAX make asynchronous calls to a web server. This means client browsers are avoid waiting for all data arrive before start the rendering.

Form Validation
This is the biggest advantage. Form are common element in web page. Validation should be instant and properly, AJAX gives you all of that, and more.

Bandwidth Usage
No require to completely reload page again. AJAX is improve the speed and performance. Fetching data from database and storing data into database perform background without reloading page.

DISADVANTAGES OF AJAX

1. AJAX application would be a mistake because search engines would not be able to index an AJAX application.

2. Open Source: View source is allowed and anyone can view the code source written for AJAX.

3. ActiveX requests are enabled only in Internet Explorer and newer latest browser.

4 The last disadvantage, XMLHttpRequest object itself. For a security reason you can only use to access information from the web host that serves initial pages. If you need to fetching information from another server, it's is not possible with in the AJAX.

AJAX - EXAMPLES

Here is a list of some famous web applications that make use of AJAX.

Google Maps

A user can drag an entire map by using the mouse, rather than clicking on a button.

https://maps.google.com/

Google Suggest

As you type, Google offers suggestions. Use the arrow keys to navigate the results.

https://www.google.com/webhp?complete=1&hl=en

Gmail

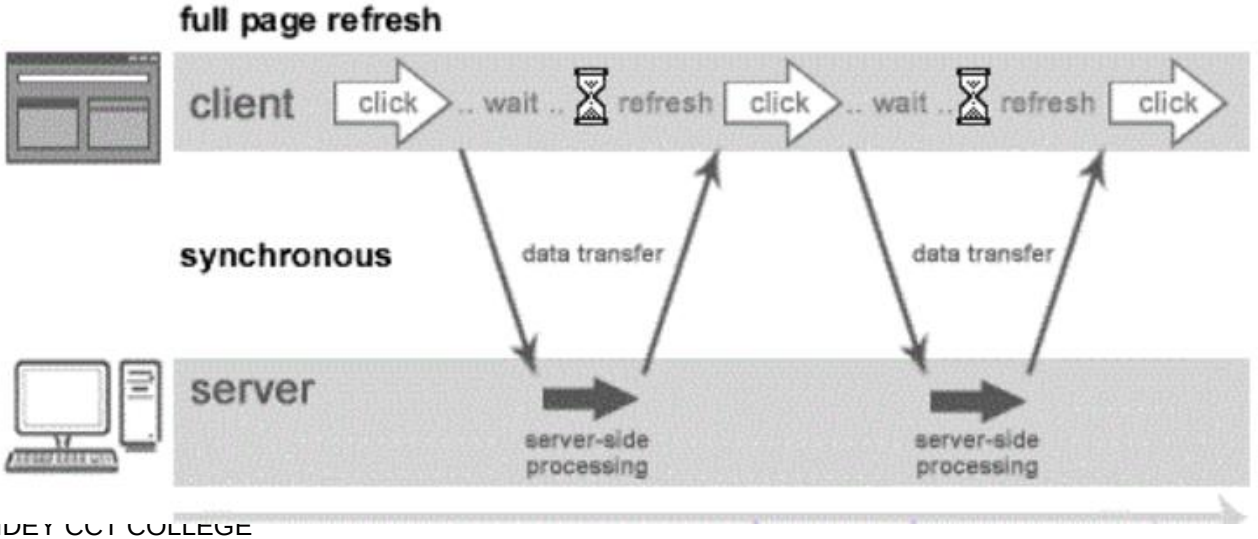
Gmail is a webmail built on the idea that emails can be more intuitive, efficient, and useful.

https://gmail.com/

Understanding Synchronous vs Asynchronous

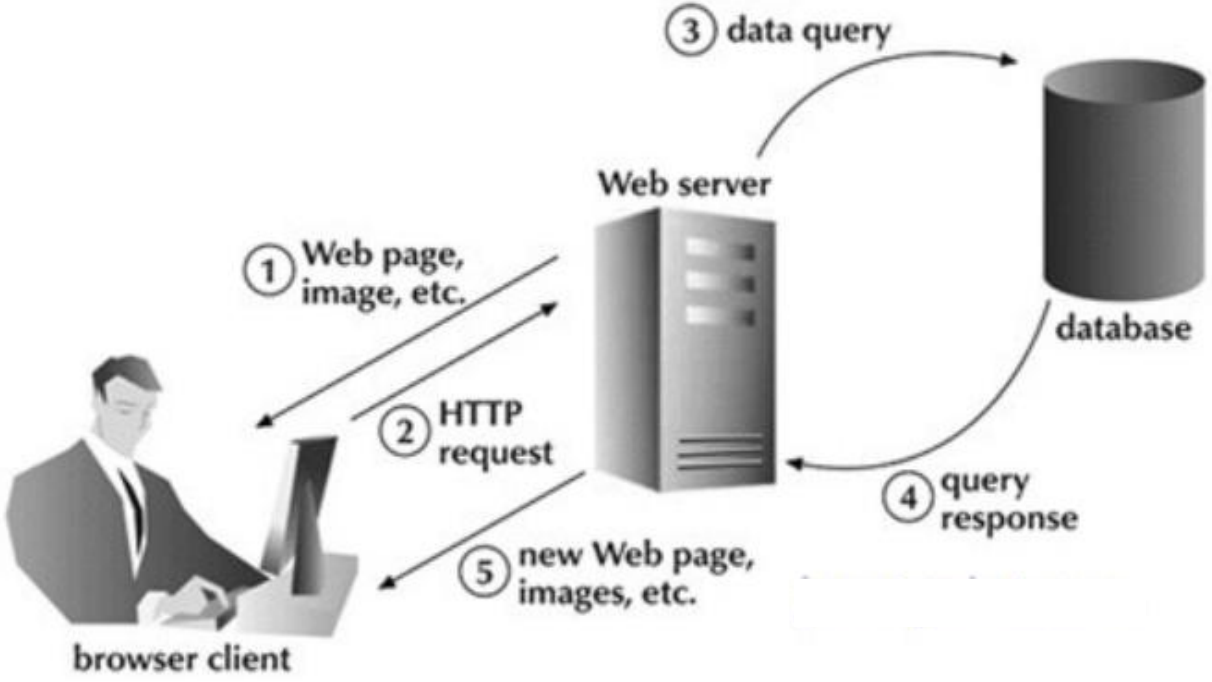
Synchronous (Classic Web-Application Model)

A synchronous request blocks the client until operation completes i.e. browser is unresponsive. In such case, javascript engine of the browser is blocked.



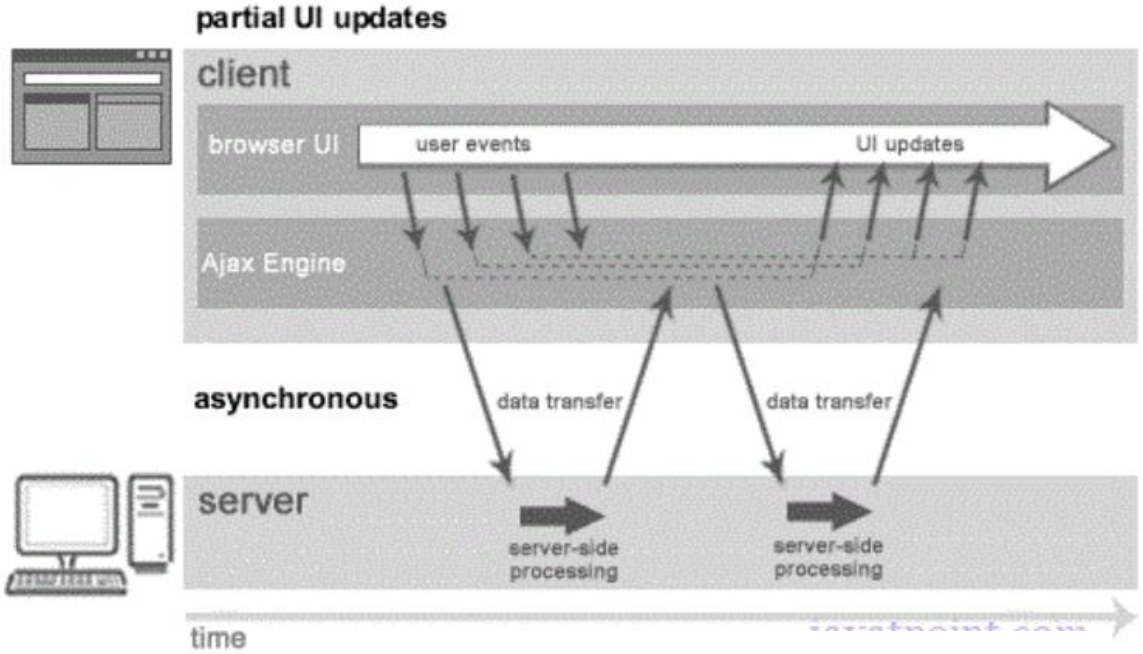
As you can see in the above image, full page is refreshed at request time and user is blocked until request completes.

Let's understand it another way.



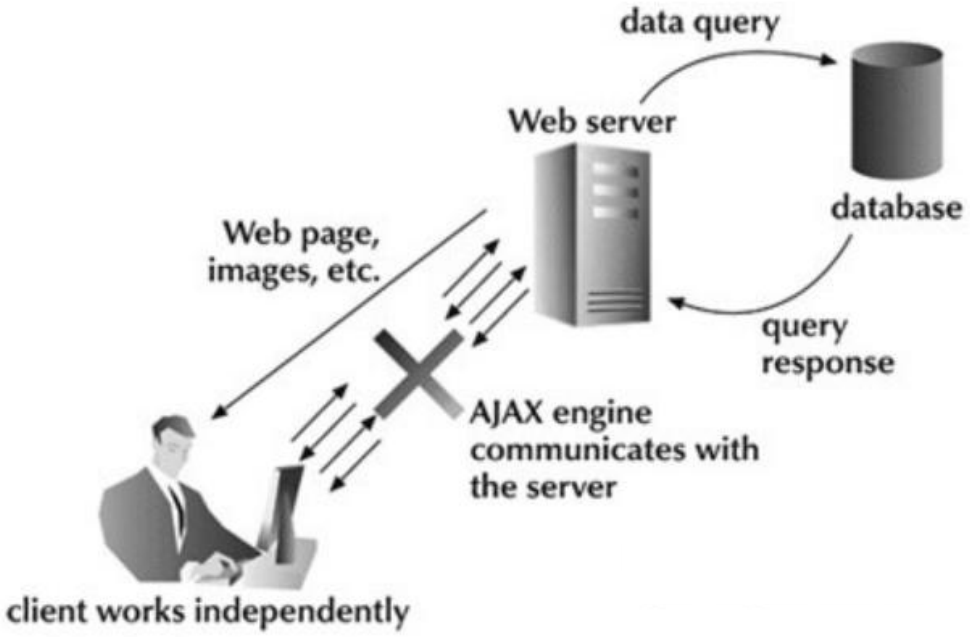
Asynchronous (AJAX Web-Application Model)

An asynchronous request doesn't block the client i.e. browser is responsive. At that time, user can perform another operations also. In such case, javascript engine of the browser is not blocked.



As you can see in the above image, full page is not refreshed at request time and user gets response from the ajax engine.

Let's try to understand asynchronous communication by the image given below.



WHAT IS AJAX XMLHTTPREQUEST OBJECT?

XMLHttpRequest object is an **API for fetching any text base format data,** including **XML** without **user/visual interruptions**. All most **all browser** platform support **XMLHttpRequest object** to make **HTTP requests**.

Following are sequence of **step** for working with **XMLHttpRequest object**:

1. **Define instance** of this XMLHttpRequest.
2. **Create a asynchronous call** to a server page, also defining a **callback function** that will **automatically execute** when the server response is received.
3. Callback function **getting server response**.
4. **DOM manipulate** received data and **added into live page**.

Define instance of this XMLHttpRequest object

Create new instance of the XMLHttpRequest object, using *new* keyword,

var xhr = new XMLHttpRequest();

XMLHttpRequest Object Methods

new XMLHttpRequest()	Creates a new XMLHttpRequest object
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(*method,url,async,user,psw*)	Specifies the request *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password
send()	Sends the request to the server Used for GET requests
send(*string*)	Sends the request to the server. Used for POST requests
setRequestHeader()	Adds a label/value pair to the header to be sent

XMLHttpRequest Object Properties

onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found"
statusText	Returns the status-text (e.g. "OK" or "Not Found")

IMPLEMENTING AJAX FRAMEWORK

The AJAX Framework is a cross browser framework that allows developers to quickly develop web pages that can call web services, web pages and other types of content through JavaScript without having to submit the current page.The AJAX Framework is:

Easy to use.

Works well with other existing frameworks.

Supports both "GET" and "POST".

Works with Internet Explorer 6+, Opera 8+, Safari 3+, Firefox 2+, Google's Chrome and other Mozilla based browsers.

SIMPLE AJAX EXAMPLE

Game-list.php

```
<ul>
<li>ravi</li>
<li>raj</li>
<li>can</li>
<li>home</li>
</ul>
```

```
<script type="text/javascript">
function getcontent()
{
var xmlHttp=new XMLHttpRequest();
xmlHttp.open("GET","game-list.php", false);
xmlHttp.send(null);
var element=document.getElementById("content");
element.innerHTML=xmlHttp.responseText;
}
function hidecontent()
{
        var element=document.getElementById("content");
        element.innerHTML="";
}
</script>
<form>
<input onclick=getcontent(); type="button" value="get content" />
<input onclick=hidecontent(); type="button" value="hide content" />
</form>
<div id="content">


</div>
```