CSE417: WEB ENGINEERING

Daffodil International University

LEARNING OUTCOMES

- Basic concepts of-
  - XML
  - JSON
  - Ajax

XML

- XML stands for eXtensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be both human- and machine-readable.

Why Study XML?

- XML plays an important role in many different IT systems.
- XML is often used for distributing data over the Internet.
- It is important (for all types of software developers!) to have a good understanding of XML.

XML USAGE

- XML Separates Data from Presentation
    - XML does not carry any information about how to be displayed.
    - The same XML data can be used in many different presentation scenarios.
    - Because of this, with XML, there is a full separation between data and presentation.
- XML is Often a Complement to HTML
    - In many HTML applications, XML is used to store or transport data, while HTML is used to format and display the same data.
- 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.

XML EXAMPLE

Example 1

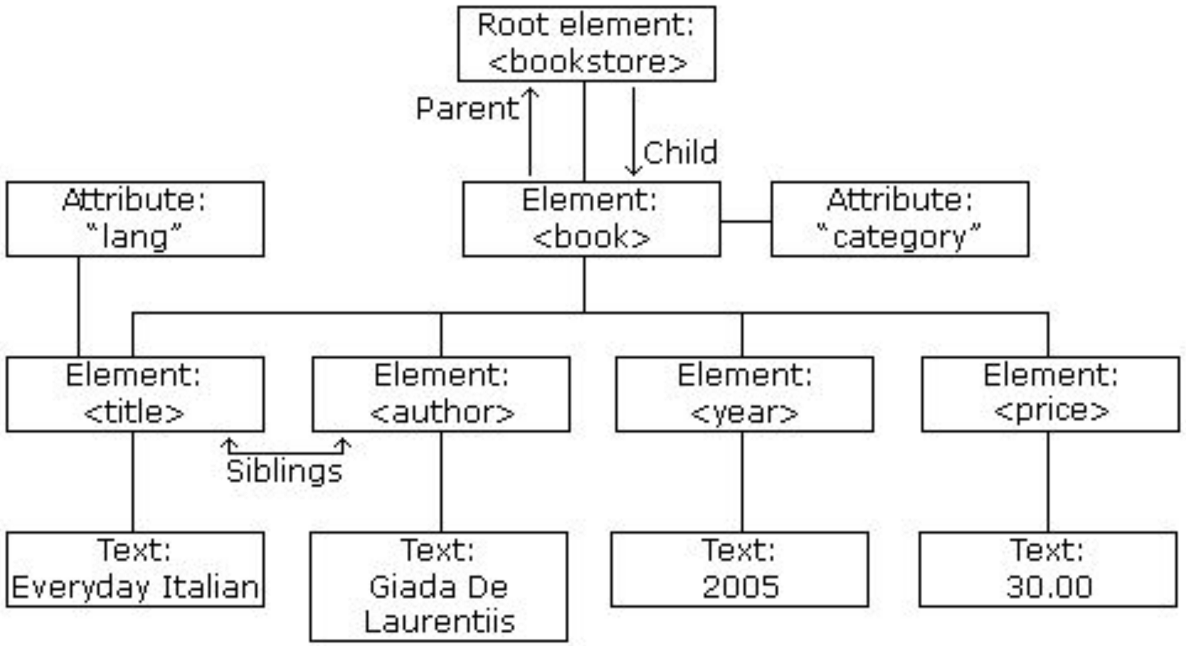
```xml
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Example-2

```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>

</bookstore>
```

XML…

XML Tree Structure



More on XML…

The image in left represents books in this XML:

```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">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

JSON

- JSON: **J**ava**S**cript **O**bject **N**otation.
- JSON is a syntax for storing and exchanging data.
- JSON is text, written with JavaScript object notation
- Exchanging Data
  - Data between a browser and a server, the data can only be text.
  - JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server.
  - We can also convert any JSON received from the server into JavaScript objects.

JSON SYNTAX

- JSON syntax is derived from JavaScript object notation syntax:
  - Data is in name/value pairs
  - Data is separated by commas
  - Curly braces hold objects
  - Square brackets hold arrays

JSON data is written as **name/value pairs**.

```
"name":"John"
```

In **JSON**, ***keys* must be strings**, written **with double quotes**:

```
{ "name":"John"
 }
```

In **JavaScript**, keys can be strings, numbers, or identifier names:

```
{ name:"John" }
```

MORE JSON SYNTAX…

- With JavaScript you can create an object and assign data to it, like this:

```
var person = { name: "John", age: 31, city: "New York" };
```

- You can access a JavaScript object like this:

```
person.name; // returns John

person["name"];// this also works
```

- Data can be modified like this:

```
person.name = "Gilbert";//Another way to do it?
```

- JavaScript arrays can also be used as JSON. [How?]

Parse the data with JSON.parse(), and the data becomes a JavaScript object.

MORE…

Arrays in PHP will also be converted into JSON when using the PHP function `json_encode()`

```
<?php
$myArr = array("John", "Mary", "Peter", "Sally");

$myJSON = json_encode($myArr);

echo $myJSON;
?>
```

JSON can be used along with Ajax, html, php, JS too…

There are other functions too: json_decode(), stringify() etc.
What are the purpose of these functions?

JSON VS XML

JSON is Like XML Because

- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages
- Both JSON and XML can be fetched with an XMLHttpRequest

JSON is Unlike XML Because

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

Question: Why JSON is Better Than XML?

AJAX

AJAX = **A**synchronous **J**avaScript **A**nd **X**ML.
AJAX is not a programming language.

- Read data from a web server - after the page has loaded
- Update a web page without reloading the page
- Send data to a web server - in the background

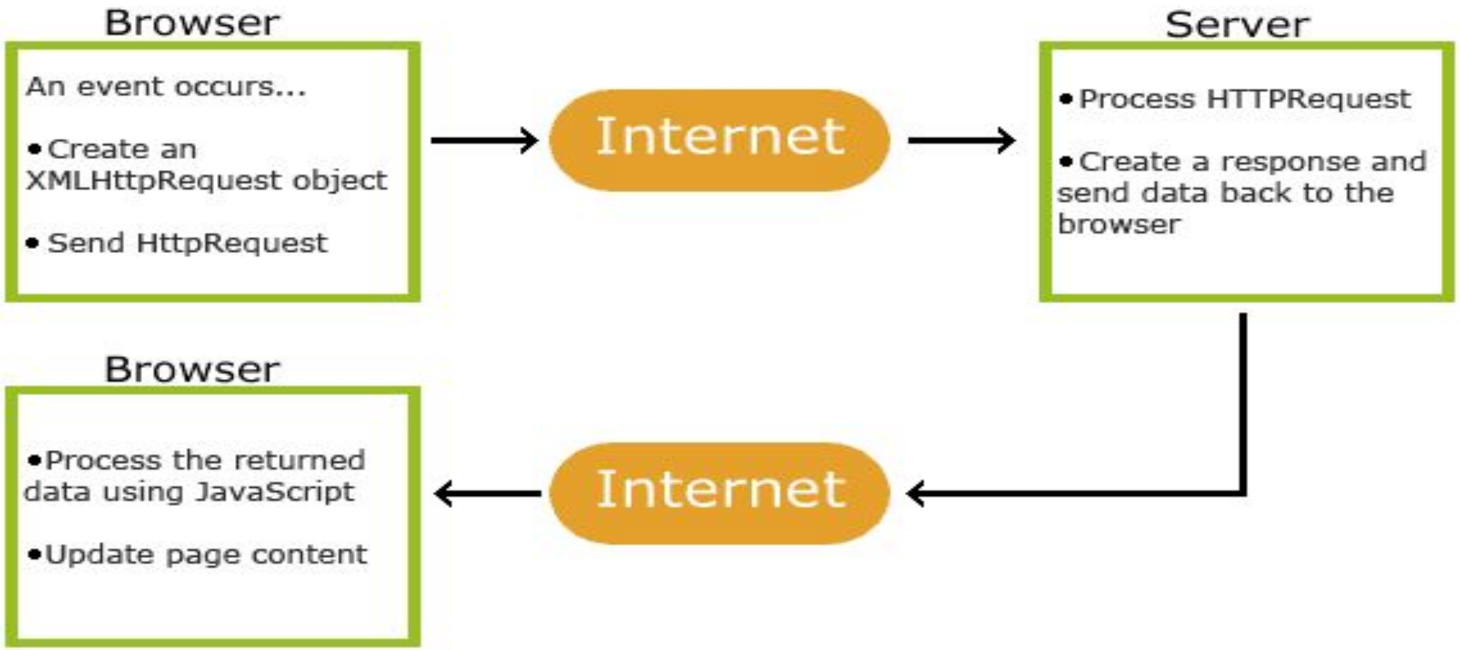
AJAX just uses a combination of:

- A browser built-in XMLHttpRequest object (to request data from a web server)
- JavaScript and HTML DOM (to display or use the data)

AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.

HOW AJAX WORKS

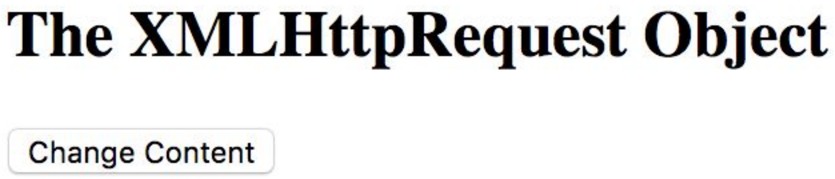
1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript


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

Example

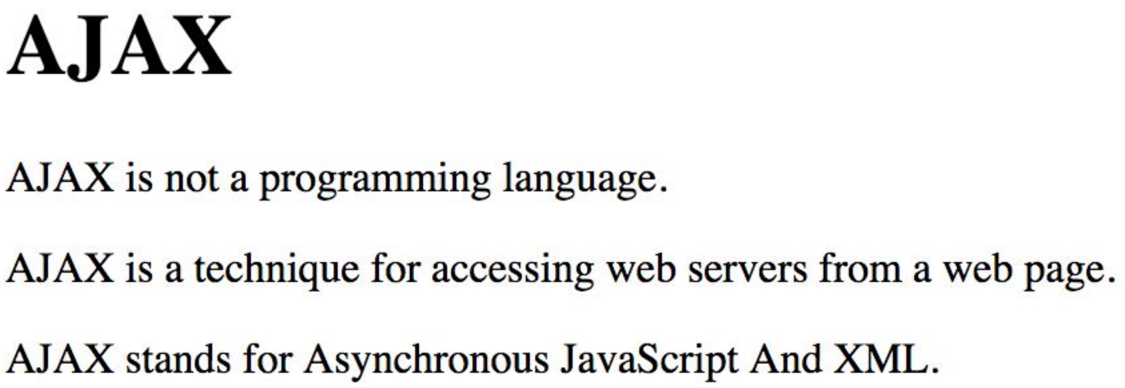
```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
<h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="loadDoc()">Change Content</button>
</div>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

Before



The XMLHttpRequest Object

After



AJAX

AJAX is not a programming language.

AJAX is a technique for accessing web servers from a web page.

AJAX stands for Asynchronous JavaScript And XML.

More Ajax…Request

- The keystone of AJAX is the XMLHttpRequest object.

```
var xhttp = new XMLHttpRequest(); //Creates a new XMLHttpRequest object
```

- The XMLHttpRequest object is used to exchange data with a server.

```
xhttp.open("GET", "demo_get.asp", true);xhttp.send();
```

Can we use POST method?

- With the XMLHttpRequest object you can define a function to be executed when the request receives an answer.
- The function is defined in the onreadystatechange property of the XMLHttpRequest object:

```
xhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    document.getElementById("demo").innerHTML = this.responseText;
  }
};
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

More…

THE ONREADYSTATECHANGE PROPERTY

- The readyState property holds the status of the XMLHttpRequest.
- The onreadystatechange property defines a function to be executed when the readyState changes.
- The status property and the statusText property holds the status of the XMLHttpRequest object.

Property	Description
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
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

More Ajax…Response

- The onreadystatechange function is called every time the readyState changes.
- When readyState is 4 and status is 200, the response is ready:

```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
```

WHAT'S MORE in this course?

- Now we will be choosing an Independent topic!!
- For each group, there will be one topic, it can be anything…
- Implement basic idea of that
- You will present/demonstrate
- You have one week to do that!

Few topics:

- Web Services
- Django
- Laravel
- REST
- CGI Programming
- Cloud

- look-why-these-top-web-development-trends-technologies-will-rule

- **EXERCISE**
  - Use JSON, XML and Ajax at least once in your project!
- **READINGS**
  - https://www.w3schools.com/xml/
  - https://www.w3schools.com/js/js_json_intro.asp
  - https://www.php.net/manual/en/book.json.php

ACKNOWLEDGEMENT

- This module is designed and created with the help from following sources-
  - https://cgi.csc.liv.ac.uk/~ullrich/COMP519/
  - http://www.csc.liv.ac.uk/~martin/teaching/comp519/
  - Md. Al-Amin Hossain, Daffodil International University

WEB SERVICES

- Web services are open standard (XML, SOAP, HTTP, etc.)
  - Based on web applications
  - Interaction between web applications
  - Purpose of exchanging data.
- Web services are web application components.
- Web services can be published, found, and used on the Web.

WEB SERVICES…

- WSDL
  - Web Services Description Language
  - XML-based language for describing Web services.
- SOAP
  - Simple Object Access Protocol
  - XML based protocol for accessing Web Services.
- RDF
  - Resource Description Framework
  - Framework for describing resources on the web
  - RDF is written in XML

WEB SERVICES…

- RSS
  - Really Simple Syndication
  - Allows you to syndicate your site content
  - Easy way to share and view headlines and content
  - RSS files can be automatically updated
  - RSS allows personalized views for different sites
  - RSS is written in XML

XML RSS

- RSS is useful for web sites that are updated frequently, like:
  - News sites - Lists news with title, date and descriptions
  - Companies - Lists news and new products
  - Calendars - Lists upcoming events and important days
  - Site changes - Lists changed pages or new pages

WSDL, SOAP, RDF are W3C recommendation

//By the way, what is w3c?

There is no official standard for RSS.

RSS EXAMPLE

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">

<channel>
  <title>W3Schools Home Page</title>
  <link>https://www.w3schools.com</link>
  <description>Free web building tutorials</description>
  <item>
    <title>RSS Tutorial</title>
    <link>https://www.w3schools.com/xml/xml_rss.asp</link>
    <description>New RSS tutorial on W3Schools</description>
  </item>
  <item>
    <title>XML Tutorial</title>
    <link>https://www.w3schools.com/xml</link>
    <description>New XML tutorial on W3Schools</description>
  </item>
</channel>

</rss>
```

More…