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.

XML EXAMPLE

Example 1

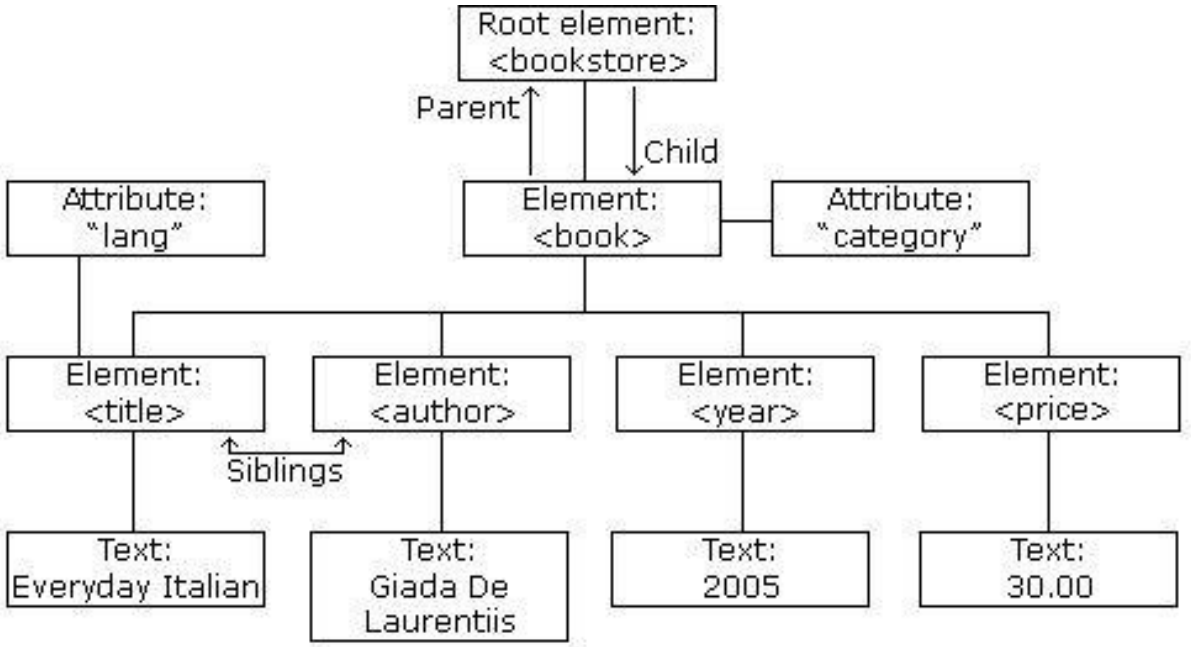
```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Example-2

```
<?xml 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



The image in left represents books in this 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 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

XML Vs. JSON

XML

```
<empinfo>
    <employees>
        <employee>
            <name>James Kirk</name>
            <age>40></age>
        </employee>
        <employee>
            <name>Jean-Luc Picard</name>
            <age>45</age>
        </employee>
        <employee>
            <name>Wesley Crusher</name>
            <age>27</age>
        </employee>
    </employees>
</empinfo>
```

JSON

```
{   "empinfo" :
        {
            "employees" :   [
            {
                "name" : "James Kirk",
                "age" : 40,
            },
            {
                "name" : "Jean-Luc Picard",
                "age" : 45,
            },
            {
                "name" : "Wesley Crusher",
                "age" : 27,
            }
                                ]
        }
}
```

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
```

MORE…

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

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

$myJSON = json_encode($myArr);

echo $myJSON;
?>
```

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

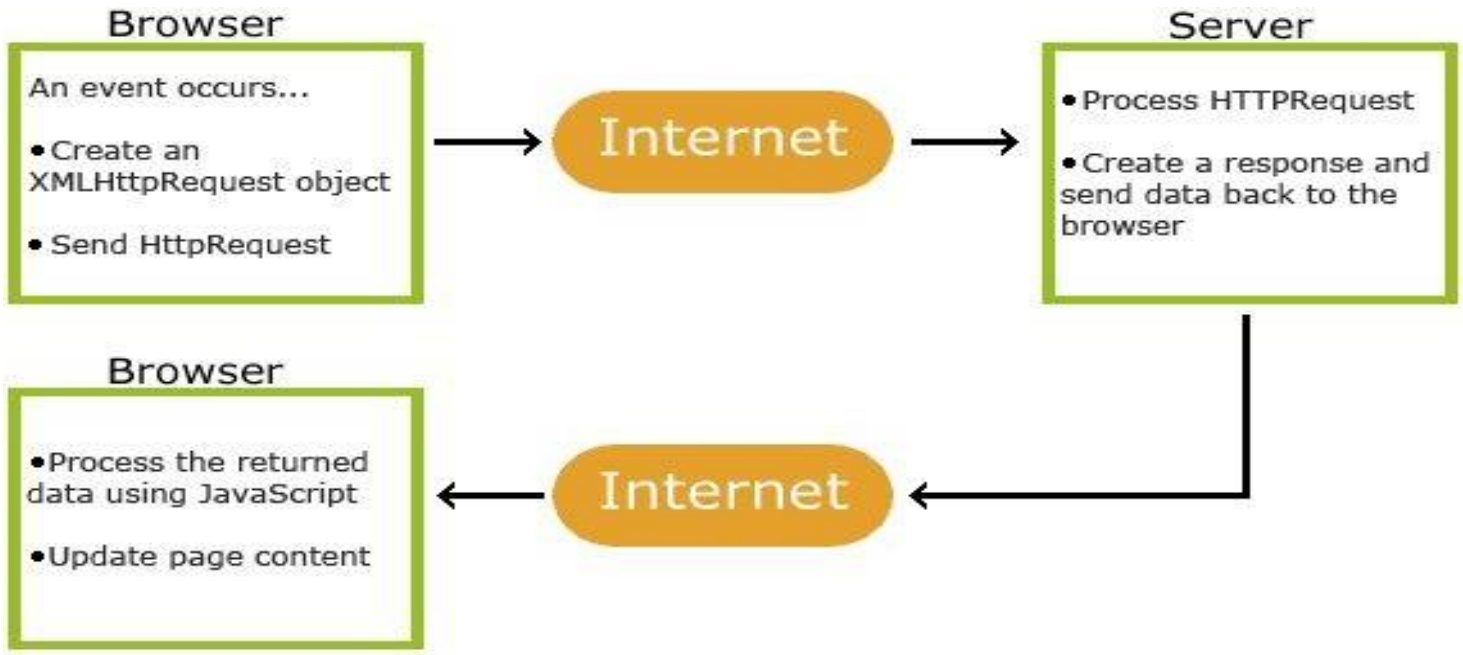
AJAX just uses a combination of:

- 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

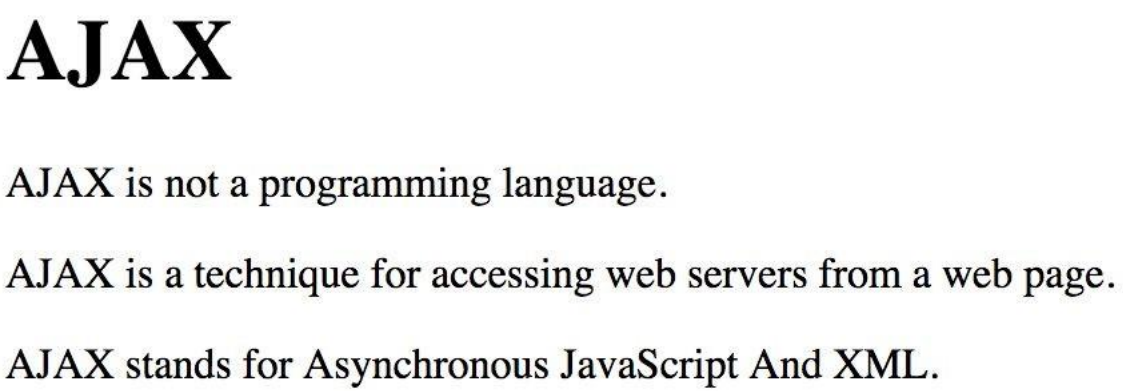


Example

```
<!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

After



- **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