Introduction to Web Technology

Course Code: **CSC 3222** Course Title: WEB TECHNOLOGIES

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**1**	**Week No:**	**1**	**Semester:**	**Fall 2024-2025**
Lecturer:	*MD.AL-AMIN (alamin@aiub.edu)*				

Lecture Outline

1. **Course Objectives**
2. **Course Logistics**
    - I. **Grading policy**
    - II. **OBE evaluation**
    - III. **Classroom and Course Policies**
    - IV. **Lab Work and Assignment**
    - V. **Required software and tools**
3. **Learning Objectives**
4. **Client/Server model**
    - I. **Three-tier Client Server Architecture**
    - II. **Peer to Peer (P2P)**
    - III. **Internet vs WWW**
    - IV. **Uniform Resource Identifier (URI)**
5. **Browsers**

Lecture Outline(contd.)

6. **HTTP protocol**
    I. **Process of http**
    II. **http Steps**
    III. **HTTP Request Methods**
7. **HTML**
8. **XML**
9. **XHTML**
10. **What is the DOM?**
11. **DHTML**
12. **Book**
13. **References**

Course Objectives

- In this course, we will learn about Web Technology and those branches: HTTP, HTML, CSS, JavaScript and PHP.
- To understand, Client-Server based applications architecture.
- Design and develop real life and society targeted Client-Server based Web applications.

Course Logistics

Grading policy

Marking Distribution (Midterm and Final term)	
Quiz	20%
Attendance	10%
Assignment & performance	10%
Lab Exam	20%
Term Project	40%
Total	100%

Final Grade/ Grand Total	
Midterm:	40%
Final Term:	60%
Grand Total	100%

Course Logistics

OBE evaluation

- Mid Term Project [Report]
  - CO1- Describe the increasing importance of web technologies on modern society and environment. **[Project Proposal, Background Study, Requirement Analysis – 15 marks]**
  - CO3- Design real life and society targeted Client-Server based Web applications**.[Entity Relationship (ER) Diagram, System Images against the Specification – 10 marks]**
- Final Term Project[Implementation]
  - CO2- Apply the fundamental web technologies to obtain business sustainability.**[Completeness, Validation, Feature Implementation against the Specification – 15 marks]**
  - CO4- Develop real life and society targeted Client-Server based Web applications.**[Concept Understanding, Promptness– 10 marks]**

Course Logistics

Classroom and Course Policies

- No make-up Quiz/Assignment will be taken.
- At least 80% presence is required by the student to sit for examination. Absent classes must be defended through application and proper documentation to the department.
- Student come after 10-15 minutes of due time is considered late.
- 3 late attendances are considered as one absent.
- Late during quiz/presentation are not given additional time.

Course Logistics

Lab Work and Assignment

- Every theory class will be followed by a lab class.
- We will run and test lab work in every lab class.
- There will be an assessment in every lab.
- Every one should have **github account(version control).**
- All assignments and project(lab tasks) push in the github repositories.
- Please submit your github account in the first lab. [Google form Link]
- Deploy your application on www.heroku.com if possible.

Course Logistics

Required software and tools

- Text editor to write HTML, CSS, JavaScript, php code. E.g. Sublime Text, Note Pad ++,VS code, atoms, etc.
- Modern browser and server need to test them out. E.g. – google chrome(browser), XAMP(apache server).

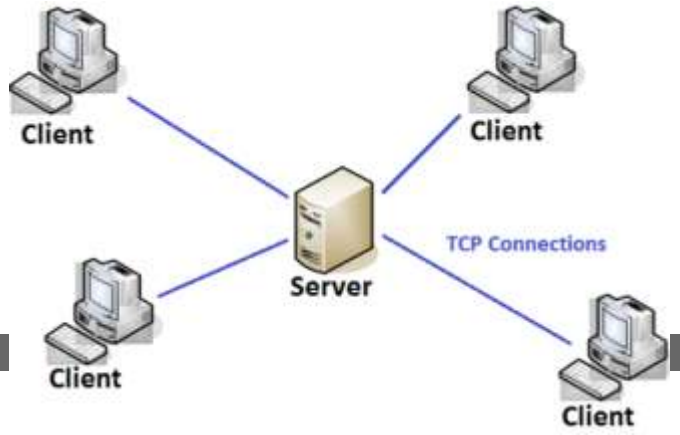
Learning Objectives

- In this lecture , we will learn about Web Technology and few terminology : HTTP, HTML, XML and XHTML.
- We will learn about how web technology can help in the Business World as well.

Client/Server model

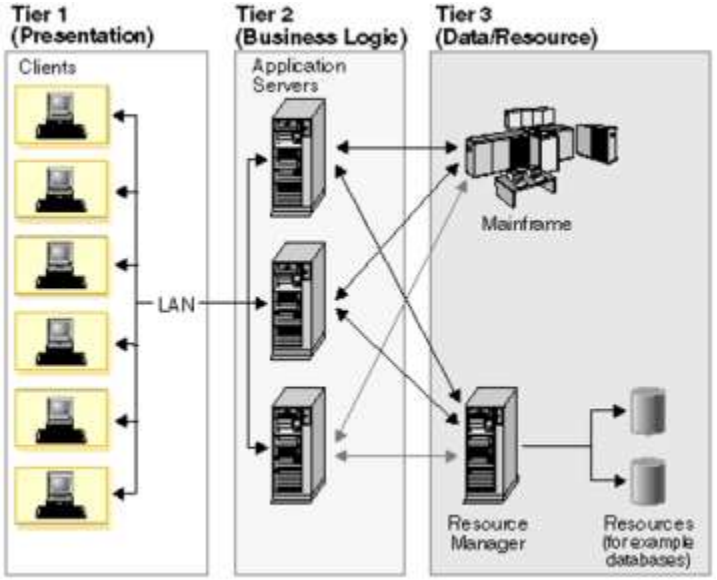
Client Server Architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or internet connection. This system shares computing resources.

Client/server architecture is also known as a networking computing model or client/server network because all the requests and services are delivered over a network.



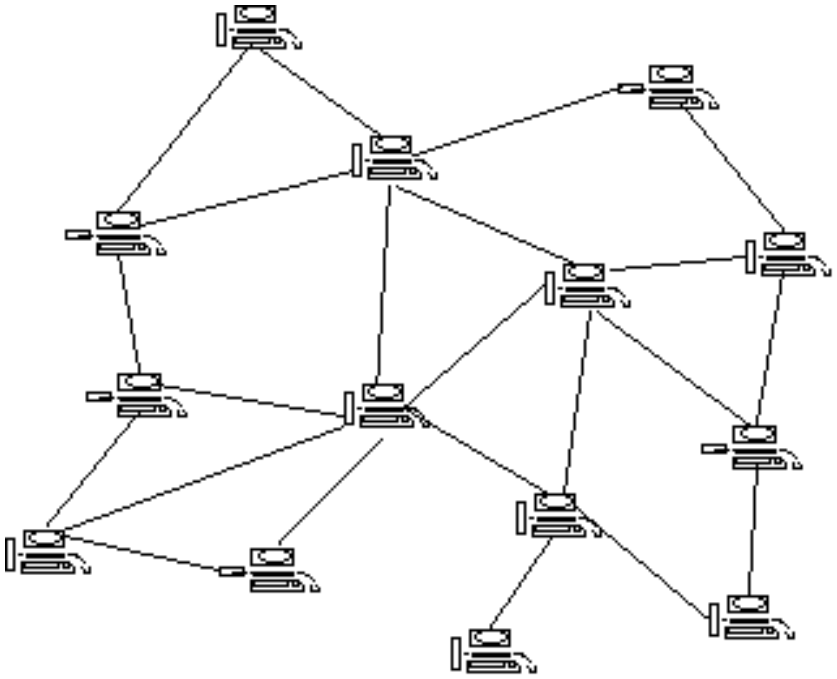
Three-tier Client Server Architecture

- The traditional client/server architecture involves two levels, a client level and a server level. Another common design of client/server systems uses three tiers:
- A client that interacts with the user
- An application server that contains the business logic of the application
- A resource manager that stores data.



Peer to Peer (P2P)

- Distributed application architecture that partitions tasks or work loads between peers.
- Peers are equally privileged, equipotent participants in the application.
- They are said to form a peer-to-peer network of nodes.



Internet and World Wide Web

- **Global system of interconnected computer networks .**
  - Use the **standard Internet Protocol Suite (TCP/IP)**.
  - Serve billions of users worldwide.
  - Network of networks that consists of millions of private, public, academic, business, and government networks.
  - Carries a vast range of **information resources** and services, such as the inter-linked **hypertext** documents of the **World Wide Web** and the infrastructure to support **electronic mail**.
- **World Wide Web** commonly known as the **Web(Abbreviated as WWW).**
- A system of interlinked hypertext documents accessed via the Internet. With a web browser, any one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

Internet vs WWW

- Internet is global system of interconnected computer networks. In short, the Web is an application running on the Internet.
- WWW is one of the services that runs on the Internet. It is a collection of interconnected documents and other resources, linked by hyperlinks and URLs. An application running on the Internet.

WWW function

- Viewing a web page on the **World Wide Web** normally begins
  - typing the URL of the page into a web browser,
  - following a hyperlink to that page or resource.
- The web browser then initiates a series of communication messages, behind the scenes, in order to fetch and display it.
- First, the server-name portion of the URL is resolved into an **IP address** using the global, distributed Internet database known as the **Domain Name System (DNS).**
- The browser then requests the resource by sending an HTTP request to the Web server at that particular address.
- While receiving these files from the web server, browsers may progressively render the page onto the screen as specified by its HTML, Cascading Style Sheets (CSS), or other page composition languages.

WWW standards

- Many of the documents are the work of the World Wide Web Consortium (W3C).
- Some of the documents are produced by the Internet Engineering Task Force (IETF) and other organizations.
- markup languages, especially HTML and XHTML, from the W3C
  - define the structure and interpretation of hypertext documents
- ECMAScript (usually in the form of JavaScript), from Ecma International.
- Document Object Model, from W3C.
- Uniform Resource Identifier (URI)- a universal system for referencing resources on the Internet, such as hypertext documents and images.
- HyperText Transfer Protocol (HTTP)- how the browser and server authenticate each other.

Uniform Resource Identifier (URI)

- **String of characters used to identify a name or a resource on the Internet.**
  - enables interaction with representations of the resource over a network (typically the World Wide Web) using specific protocols.
  - Schemes specifying a concrete syntax and associated protocols define each URI.
- **defined as consisting of four parts, as follows**
- **<scheme name> : <hierarchical part> [? <query>] [# <fragment>]**
- The scheme name consists of a letter followed by any combination of letters
- The hierarchical part of the URI is intended to hold identification information hierarchical in nature.
- Usually this part begins with a double forward slash ("//"), followed by an authority part and an optional path.

Uniform Resource Identifier (Contd.)

- **The path part is a sequence of segments separated by a forward slash ("/").**
  - conceptually similar to directories.
  - Each segment can contain parameters separated from it using a semicolon (";"), though this is rarely used in practice.
- **The query is an optional part separated with a question mark**
  - which contains additional identification information which is not hierarchical in nature.
  - commonly organized as a sequence of <key>=<value> pairs separated by a semicolon or by an ampersand
  - e.g. Semicolon: key1=value1;key2=value2;key3=value3
    Ampersand: key1=value1&key2=value2&key3=value3
- **The fragment is an optional part separated from the front parts by a hash ("#").**
  - holds additional identifying information that provides direction to a secondary resource
  - e.g. a section heading in an article identified by the remainder of the URI.

Uniform Resource Identifier (Example)

- **http://en.wikipedia.org/wiki/URI?page=2&frame=1#Examples _of_URI_references**
  - "http" specifies the 'scheme' name
  - "en.wikipedia.org" is the 'authority‘
  - "/wiki/URI" the 'path' pointing to this article
  - “page=2&frame=1” is the query
  - "#Examples _of_URI_references " is a 'fragment' pointing to this section

Browsers

- **A web browser or Internet browser is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video, or other piece of content.**
  - e.g. Internet Explorer, Netscape, Mozilla Firefox, Opera, Google Chrome, Apple Safari

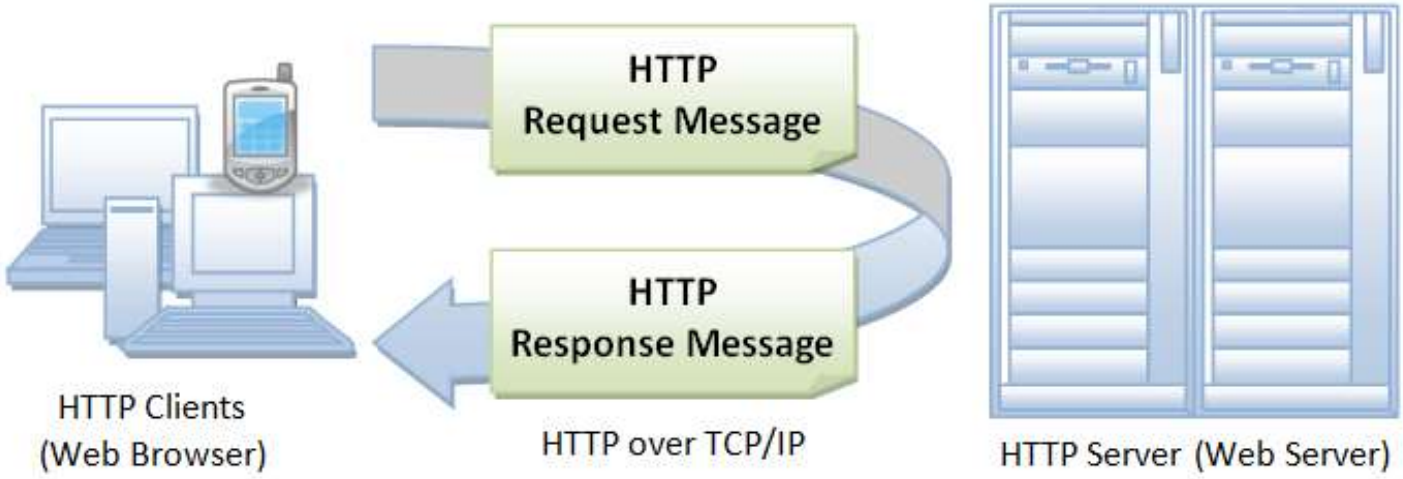
HTTP protocol

- HTTP stands for Hyper Text Transfer Protocol
- Communication between client and web servers is done by sending http Requests and receiving http Responses.
- **http is Connectionless**: The http client initiates an http request and after a request, the client waits for the response. The server processes the request and sends back response to the client. So client and server knows about each other only during current request and response.
- **http is stateless:** The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other.

HTTP protocol

Process of http

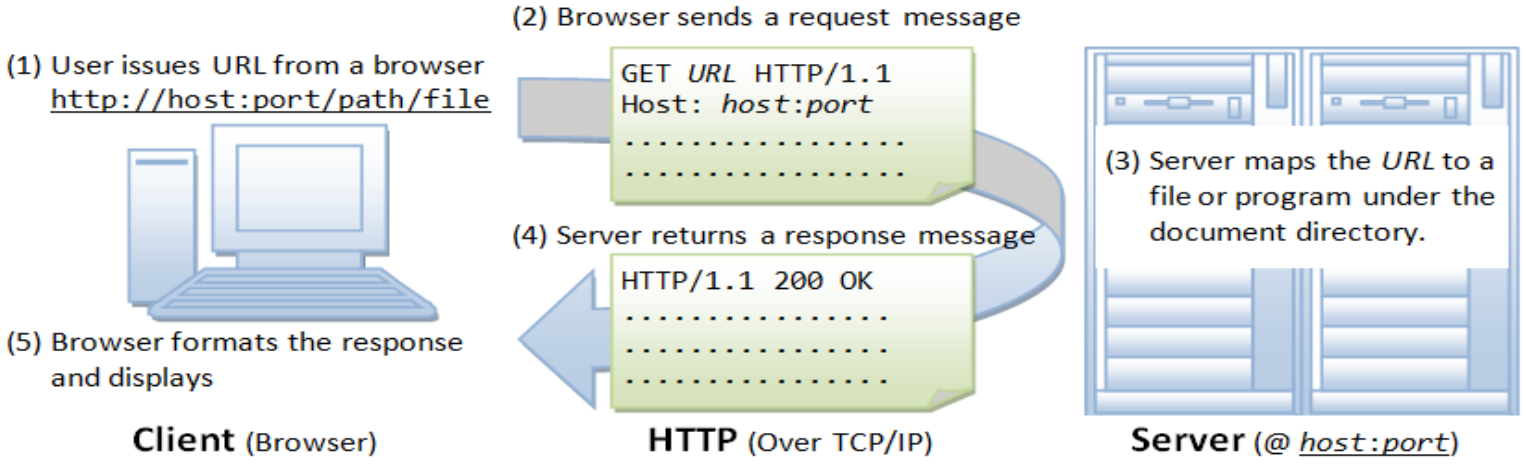
- An HTTP client sends a request message to an HTTP server. The server, in turn, returns a response message



HTTP protocol

http Steps

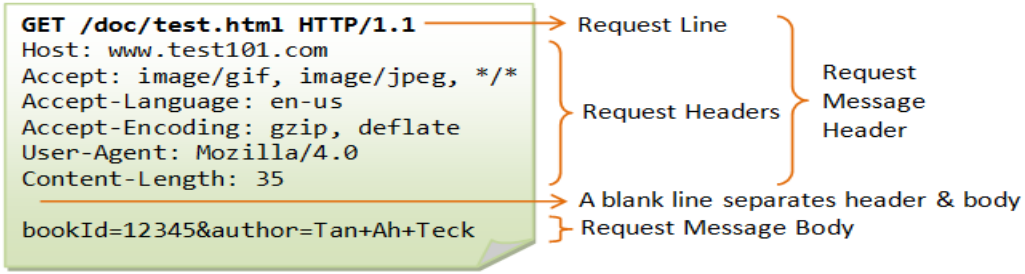
- Whenever you issue a URL from your browser to get a web resource using HTTP, e.g. http://www.abc.com/index.html, the browser turns the URL into a request message and sends it to the HTTP server.



HTTP protocol

http Steps(continue)

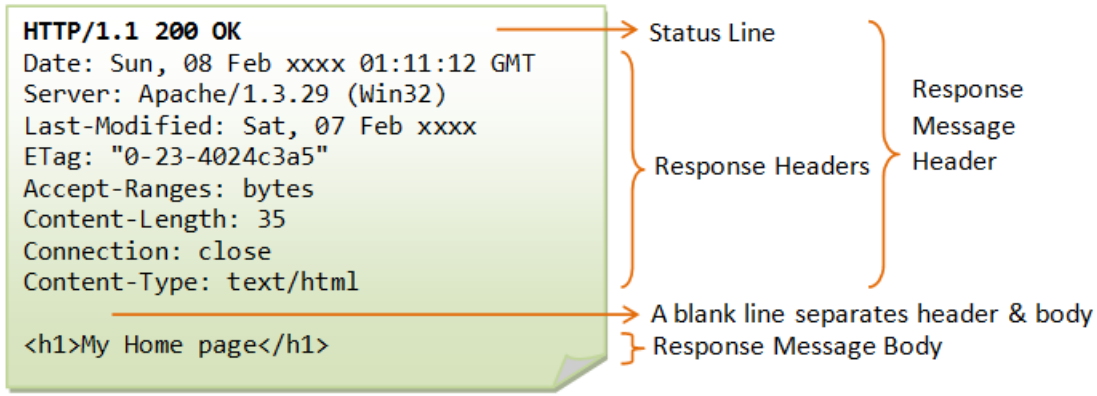
- The HTTP server interprets the request message, and returns you an appropriate response message, which is either the resource you requested or an error message.
- A URL (Uniform Resource Locator) is used to uniquely identify a resource over the web. URL has the following syntax: **protocol://hostname:port/path-and-file-name**
- As mentioned, whenever you enter a URL in the address box of the browser, the browser translates the URL into a request message according to the specified protocol; and sends the request message to the server.



HTTP protocol

http Steps(continue)

- The server interprets the request received, maps the request into a file under the server's document directory, and returns the file requested to the client. Or
- The server interprets the request received, maps the request into a program kept in the server, executes the program, and returns the output of the program to the client. If the request not be satisfied, server returns an error message.
- Common status code "200 OK", "404 Not Found", "403



HTTP protocol

HTTP Request Methods

GET: A client can use the GET request to get a web resource from the server.

POST: Used to post data up to the web server. HTML form uses POST request.

PUT: Ask the server to store the data.

DELETE: Ask the server to delete the data.

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements are represented by tags
- HTML tags label pieces of content such as ”heading”, ”paragraph”, ”table”, and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

HTML Tags and Example

- HTML tags are element names surrounded by angle brackets:

<tagname>content goes here...</tagname>

```
1   <!DOCTYPE HTML>
2   <html lang="en-US">
3   <head>
4       <meta charset="UTF-8">
5       <title></title>
6   </head>
7   <body>
8
9   </body>
10  </html>
```

HTML Documents = Web Pages

- HTML documents **describe web pages**
- HTML documents **contain HTML tags** and plain text
- HTML documents are also **called web pages**
- e.g.
  - <html>
    <body>
    <h1>My First Heading</h1>
    My first paragraph.
    </body>
    </html>

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.

```
<? xml    version = "1.0">
    <Student>
        <Name> Abd El-Aziz </Name>
        <Semester>
            <Number> I </Number>
            <Course>
              <Name> XML security </Name>
              <Grade> A </Grade>
            </Course>
        </Semester>
    </Student>
```

What is XML?

- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive
- XML is a W3C Recommendation
- Maybe it is a little hard to understand, but XML does not DO anything. XML was created to structure, store, and transport information.
- XML is the most common tool for data transmissions between all sorts of applications.

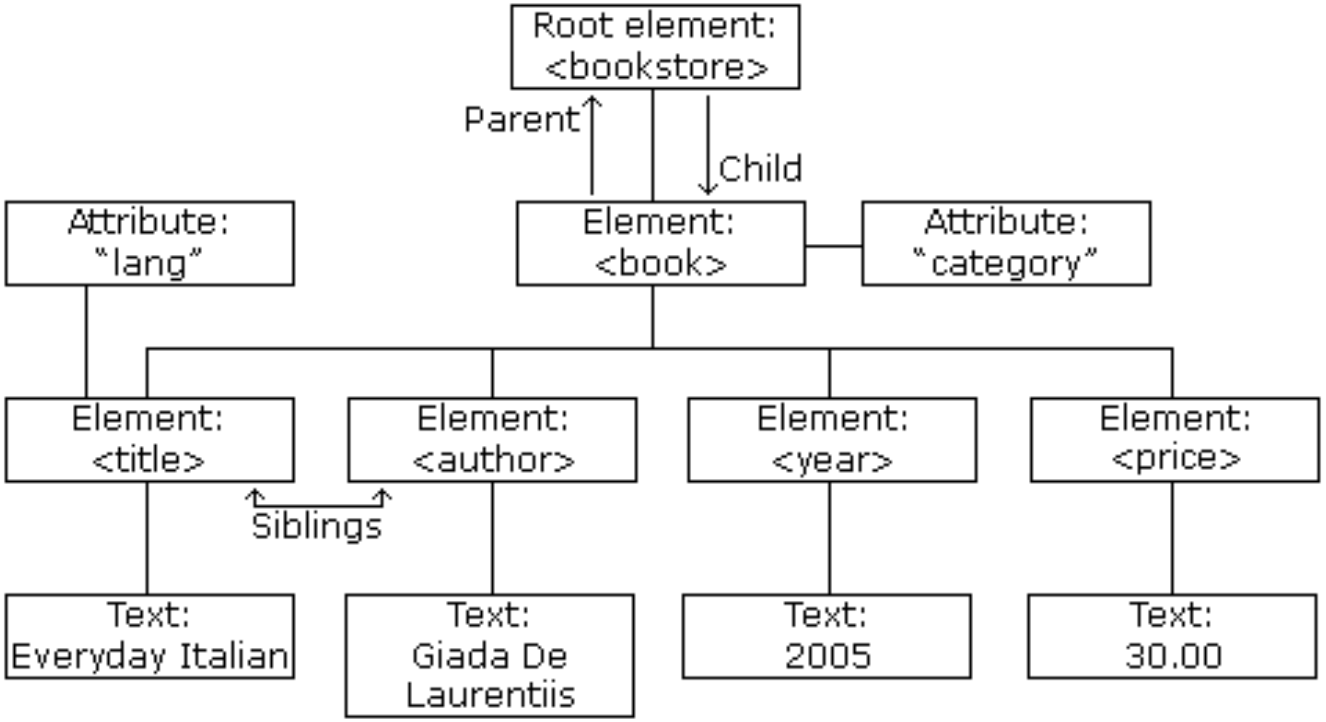
XML Documents Form a Tree Structure

- XML Documents Form a Tree Structure
- XML documents must contain a root element. This element is "the parent" of all other elements.
- The elements in an XML document form a document tree.
- The tree starts at the root and branches to the lowest level of the tree.
- All elements can have sub elements (child elements):

XML Tags and Example

An XML element is everything from (including) the element's start tag to (including) the element's end tag.

An element can contain: other elements, text, attributes or a mix of all of the above...



XML Naming Rules

XML elements must follow these naming rules:

- Names can contain letters, numbers, and other characters
- Names cannot start with a number or punctuation character
- Names cannot start with the letters xml (or XML, or Xml, etc)
- Names cannot contain spaces

Any name can be used, no words are reserved.

XML Syntax

All XML Elements Must Have a Closing Tag
XML Tags are Case Sensitive
XML Elements Must be Properly Nested
XML Documents Must Have a Root Element
XML Attribute Values Must be Quoted

XHTML

- XHTML stands for EXtensible HyperText Markup Language
- XHTML is almost identical to HTML
- XHTML is stricter than HTML
- XHTML elements must be properly nested
- bold and italic is wrong
- XHTML documents must be well-formed

<html>
<head> ... </head>
<body> ... </body>
</html>

- Tag names must be in lowercase.
- All XHTML elements must be closed.

XHTML - Why?

XHTML is a combination of HTML and XML (EXtensible Markup Language).

XHTML consists of all the elements in HTML 4.01, combined with the strict syntax of XML.

The following HTML code will work just fine if you view it in a browser (even if it does NOT follow the HTML rules):

```
<html>
        <head>
        <title>This is bad HTML</title>
        <body>
                <h1>Bad HTML
                <p>This is a paragraph
        </body>
```

XHTML - Why? (contd.)

XML is a markup language where everything must be marked up correctly, which results in "well-formed" documents.

Today's market consists of different browser technologies, some browsers run on computers, and some browsers run on mobile phones or other small devices. The last-mentioned do not have the resources or power to interpret a "bad" markup language.

Combining the strengths of HTML and XML, W3C recommended a markup language i.e XHTML.

HTML vs XHTML

XHTML elements must be **properly nested**
XHTML elements must always be **closed**
XHTML elements must be in **lowercase**
XHTML documents must have **one root element**

What is the DOM?

- a standard for accessing documents like XML and HTML
- "The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."
- The DOM is separated into 3 different parts / levels:
- Core DOM - standard model for any structured document
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents
- The DOM defines the objects and properties of all document elements, and the methods (interface) to access them.

DOM application and Example

- When an HTML page is rendered in a browser, the browser parses the markup (e.g. HTML), downloaded from the web-server into an in-memory DOM.
  - The DOM is used to construct additional internal structures used to display the page in the browser window.
- The nodes of every document are organized in a tree structure, called the DOM tree.
  - The topmost node in the DOM tree is the Document object. Each node has zero or more children.

```
|-> Document
  |-> Element (<html>)
    |-> Element (<body>)
      |-> Element (<div>)
        |-> text node
        |-> Anchor
          |-> text node
      |-> Form
           |-> Text-box
           |-> Text Area
           |-> Radio Button
           |-> Check Box
           |-> Select
           |-> Button
```

DHTML

- DHTML stands for Dynamic HTML, it is totally different from HTML.
- The DHTML is based on the properties of the HTML, JavaScript, CSS, and DOM (Document Object Model which is used to access individual elements of a document) which helps in making dynamic content.
- The DHTML make use of Dynamic object model to make changes in settings and also in properties and methods.

DHTML Example(Changing HTML Content)

- The HTML document above contains an <h1> element with id="id01"
- We use the HTML DOM to get the element with id="id01"
- A JavaScript changes the content (innerHTML) of that element to "New Heading"

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id01">Old Heading</h1>
<script>
var element = document.getElementById("id01");
element.innerHTML = "New Heading";
</script>
</body>
</html>
```

DHTML Example(Changing HTML Style)

- The following example changes the style of a <p> element:

```
<html>
<body>
<p id="p2">Hello World!</p>
<script>
document.getElementById("p2").style.color = "blue";
</script>
<p>The paragraph above was changed by a script.</p>
</body>
</html>
```

Books

- W3Schools Online Web Tutorials; URL: http://www.w3schools.com
- PHP Documentation; URL: http://www.php.net/docs.php
- Sams Teach Yourself Ajax JavaScript and PHP All in One; Phil Ballard and Michael Moncur; Sams Publishing; 2010
- JavaScript Phrasebook; Christian Wenz; Sams Publishing; 2007
- PHP and MySQL Web Development, 4/E; Luke Welling and Laura Thomson; Addison-Wesley Professional; 2009
- JavaScript for Programmers Paul J. Deitel and Harvey M. Deitel; Prentice Hall; 2009
- Beginning PHP5, Apache, and MySQL Web Development; Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz and Michael K. Glass; Wiley Publishing; 2005
- XML in a Nutshell, 3/E; Elliotte Rusty Harold and W. Scott Means; O'Reilly Media; 2004

References

1. https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
2. https://developer.mozilla.org/enUS/docs/Web/HTTP/Status
3. https://www.w3schools.com/html/html_intro.asp
4. https://www.w3schools.com/html/html_xhtml.asp