Web Applications (1)

Han-fen Hu

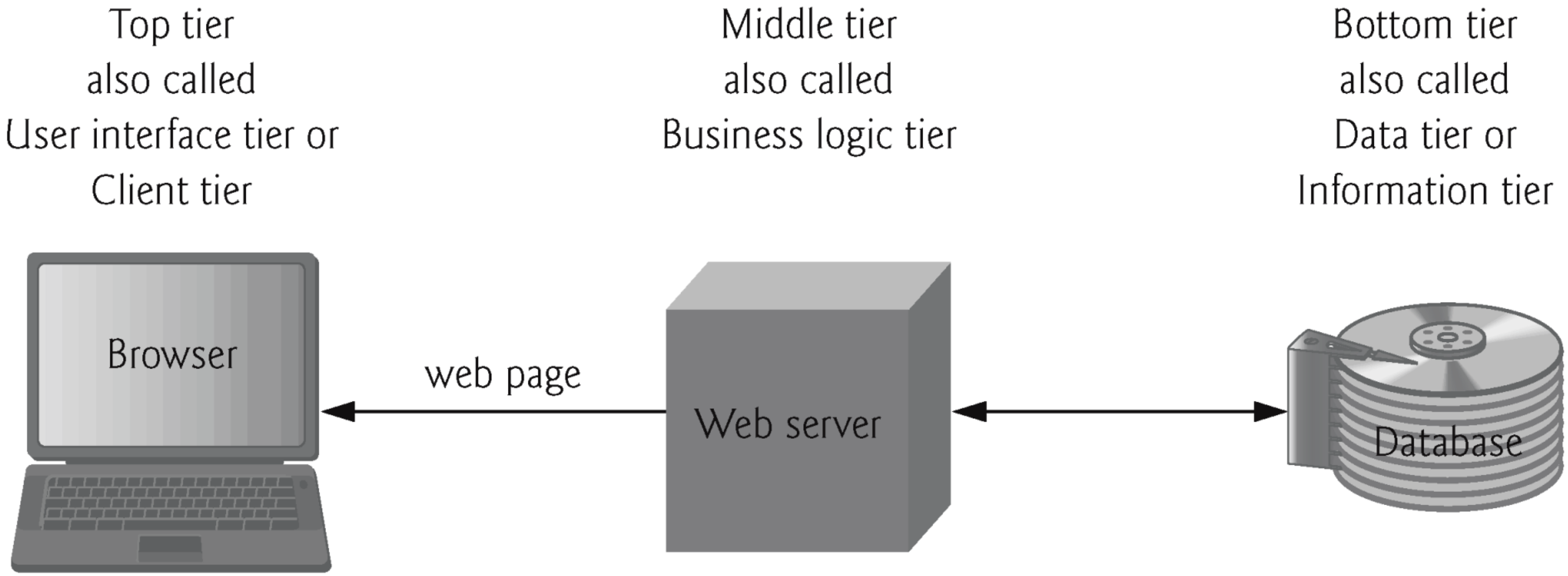
Outline

- Web Application Architecture
- Basics of HTML
- Basics of CSS
- Difference of JavaScript and Java
- Web Application Using Java
- First Java Servlet

Characteristics of Web Applications

- Interacts with users through a Web browser over a network
- Basic client-server architecture
  - Client: Interact with user, send the request via HTTP
  - Server: Handle the main business logic
- Light client / load sharing
- Cross-platform
  - Standardized user interface
  - User friendly

Web Application Architecture



Server-Side Programming

- Interpreted or executed by the server
- Main purpose: Data processing
  - Handles logging in, personal information and preferences
  - Provides the specific data which the user wants (and allows new data to be stored)
- Have a wider range of programmatic capabilities than their client-side equivalents
- Code is never seen by the user
- Respond to HTTP requests from clients

Client-Side Programming

- Interpreted or executed by the browser
  - Rely on the user's computer and browser
- Main purpose: Appearance and interaction
  - Can be used to validate user input, to interact with the browser, to enhance web pages, and to add communication between a browser and a web server.
- Receives user input and send request to server
  - Cannot read data from a server directly, must communicate via HTTP requests.

Client-Side Technology

- Hypertext Markup Language (HTML)
  - Standard markup language for creating Web pages
- Cascading Style Sheets (CSS)
  - Stylesheet language used to describe the presentation of a document written in HTML or XML
- JavaScript
  - A lightweight programming language that supports the writing of scripts to enables interactive web pages
  - inserted into HTML pages to be executed by web browser
  - Can be viewed by the client by using the browser’s source-viewing capability

HTML

- An HTML document is composed of elements that represent distinct items in the Web page
  - such as a paragraph, the page heading, or even the entire body of the page itself
- A two-sided tag is a tag that contains some document content. General syntax for a two-sided tag:
  <element>content</element>
  - e.g., <p>Welcome to the J-Prop Shop</p>
  - A two-sided tag’s opening tag (<p>) and closing tag (</p>) should completely enclose its content
- The comment tag adds notes to your HTML code
  <!-- comment -->
  - Comments can be spread over several lines

Structure of an HTML File

Information about the document,
for example the document title or the keywords

Not displayed within the Web page

```
<html>
    <head>
        head content
    </head>
    <body>
        body content
    </body>
</html>
```

Contain code that tells the browser
how to render the content

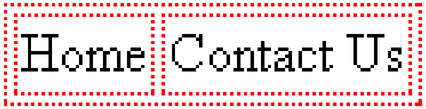
All of the content to appear on the Web page

Marking Elements with Tags

- Elements can contain other elements

```
e.g., <body>
          <p>Welcome to the J-Prop Shop</p>
      </body>
```

```
e.g., <table><tr>
              <td>Home</td>
              <td>Contact Us</td>
         </tr></table>
```



Home	Contact Us

- Tags cannot overlap

```
e.g., <body><p>Welcome to the J-Prop Shop
      </body></p>
```

Incorrect!!

Adding an Attribute to an Element

- To add an element attribute, use the format

  <element attribute1="value1"
  attribute2="value2" ...>
  content</element>

  - e.g., <img alt="an image of desert"
    src="img/desert.jpg">
  - where attribute1, attribute2, etc. are the names of attributes associated with the element, and value1, value2, etc. are the values of those attributes

Head Element: Meta Tags (1)

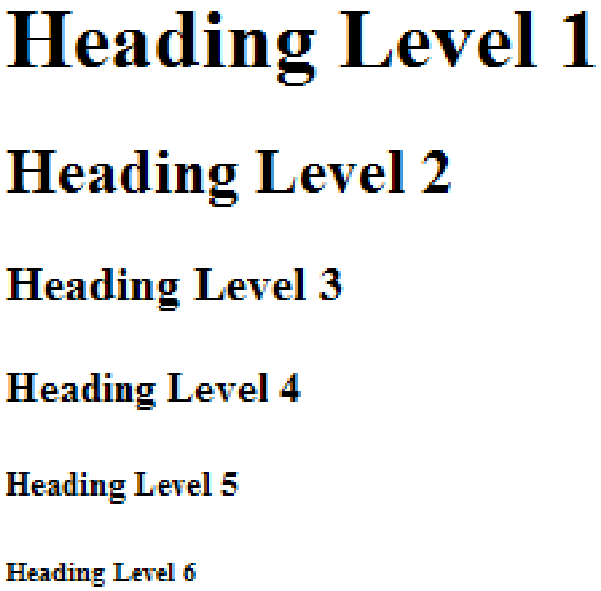
- Metadata will not be displayed on the page but will be machine parsable.
  - Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- Metadata can be used by machines
  - Browsers: how to display content or reload page
  - Search engines: what your page or site is about; help with click through rates to your sites and site rankings
  - Other web services

Head Element: Meta Tags(2)

- Meta tags always goes inside the head element
- Examples
  - Define keywords for search engines
    <meta name="keywords" content="cupcakes, las vegas, nevada, bakery, sweet, dessert, cake, pie">
  - Define a description of your web page:
    <meta name="description" content="Desert and cakes perfect for all occasions. We make your events extra special.">
  - Define the author of a page
    <meta name="author" content="Han-fen Hu">

Body Element: Heading Elements

```
<h1>Heading Level 1</h1>
<h2>Heading Level 2</h2>
<h3>Heading Level 3</h3>
<h4>Heading Level 4</h4>
<h5>Heading Level 5</h5>
<h6>Heading Level 6</h6>
```



Heading Level 1

Heading Level 2

Heading Level 3

Heading Level 4

Heading Level 5

Heading Level 6

- Since users may skim your pages by its headings, it is important to use headings to show the document structure.
- <h1> to <h6> tags are supported in all major browsers

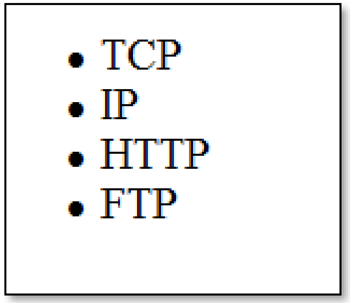
Body Element:
Paragraph and Line Break

- Paragraph element
  - <p> …paragraph goes here… </p>
  - Groups sentences and sections of text together.
  - Configures a blank line above and below the paragraph
- Line Break element
  - …text goes here <br>
    This starts on a new line….
  - Causes the next element or text to display on a new line

Body Elements: List

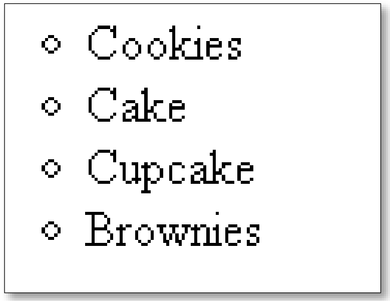
```
<ul>
      <li>TCP</li>
      <li>IP</li>
      <li>HTTP</li>
      <li>FTP</li>

</ul>
```



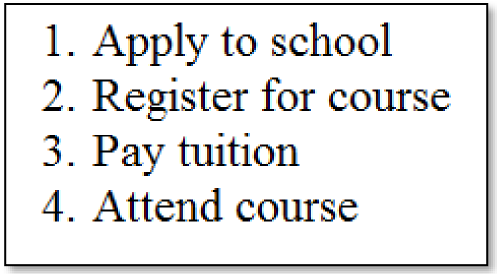
- TCP
- IP
- HTTP
- FTP

```
<ul style="list-style-type: circle;">
     <li>Cookies</li>
     <li>Cake</li>
     <li>Cupcake</li>
     <li>Brownies</li>
</ul>
```



○ Cookies
○ Cake
○ Cupcake
○ Brownies

```
<ol>
       <li>Apply to school</li>
       <li>Register for course</li>
       <li>Pay tuition</li>
       <li>Attend course</li>
 </ol>
```



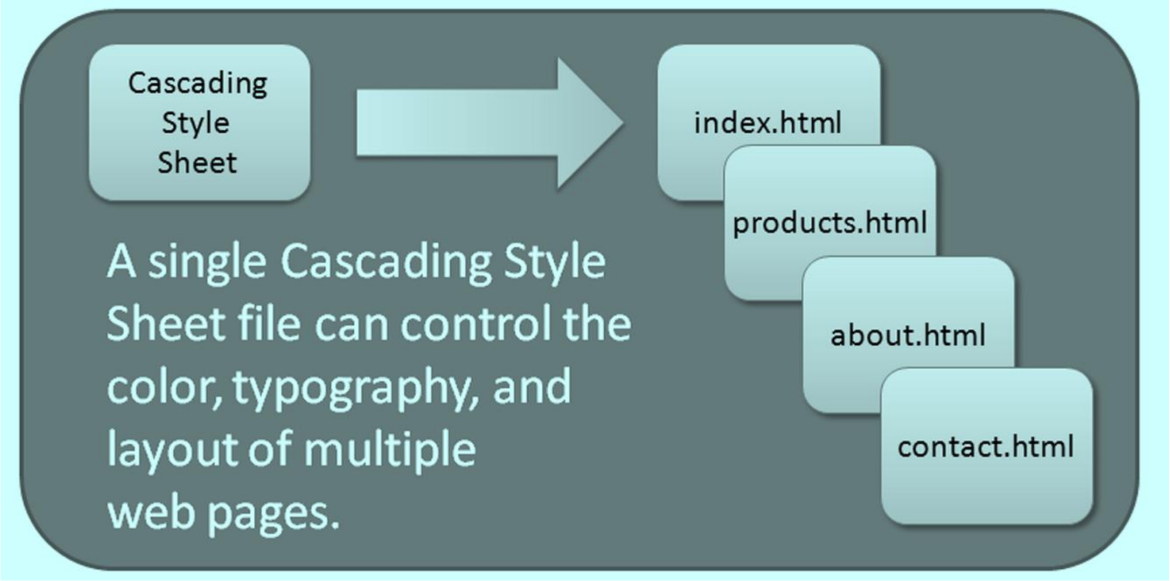
1. Apply to school
2. Register for course
3. Pay tuition
4. Attend course

Body Elements: Anchor Element

- Specifies a hyperlink reference (href) to a file
  - Text between the <a> and </a> is displayed on the web page.

- Absolute link
  - Link to other websites
    <a href="http://yahoo.com">Yahoo</a>
- Relative link
  - Link to pages on your own site
    <a href="index.htm">Home</a>
    <a href=“contact.htm">Contact Us</a>

CSS Advantages


Cascading Style Sheet
index.html
products.html
about.html
contact.html
A single Cascading Style Sheet file can control the color, typography, and layout of multiple web pages.

- Greater typography and page layout control
- Style is separate from structure
- Styles can be stored in a separate document and associated with the web page
- Potentially smaller documents
- Easier site maintenance

CSS Style Rules (1)

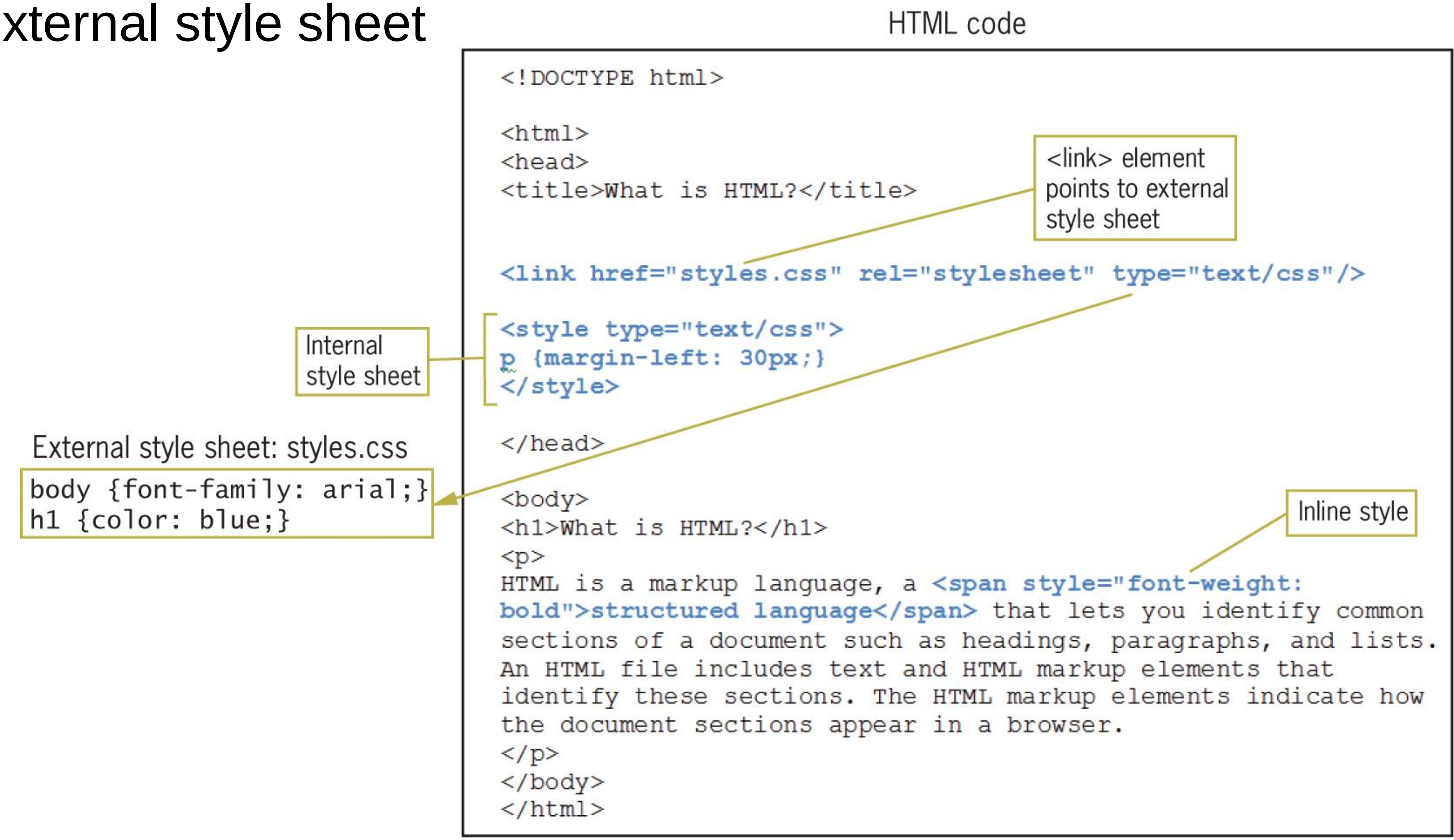
- The general syntax of a CSS style rule is

```
selector {
  property1: value1;
  property2: value2;
  property3: value3;
...
}
```

  - Semicolons (;) are used to separate multiple properties

Combining CSS Style Rules with HTML

- Inline style
- Internal style sheet
- External style sheet



HTML code
<link> element points to external style sheet
Internal style sheet
External style sheet: styles.css
Inline style

```
<!DOCTYPE html>

<html>
<head>
<title>What is HTML?</title>


<link href="styles.css" rel="stylesheet" type="text/css"/>

<style type="text/css">
p {margin-left: 30px;}
</style>

</head>

<body>
<h1>What is HTML?</h1>
<p>
HTML is a markup language, a <span style="font-weight:
bold">structured language</span> that lets you identify common
sections of a document such as headings, paragraphs, and lists.
An HTML file includes text and HTML markup elements that
identify these sections. The HTML markup elements indicate how
the document sections appear in a browser.
</p>
</body>
</html>
```

```
body {font-family: arial;}
h1 {color: blue;}
```

JavaScript

- Scripting Language
- JavaScript was designed to add interactivity to HTML pages
  - It can be inserted into HTML pages to be executed by web browser
- A subset of Java

Java vs. JavaScript

Java	JavaScript
A Compiled language	An interpreted language
Compiled code runs on the client-side computer or on the web server	Code run on the client-side computer directly within the Web browser
Requires a Java Development Kit (JDK) to create an Java application	Only a text editor is needed
Requires a Java virtual machine to run a Java application	Requires a browser that can interpret JavaScript code
Source code is hidden from users	Source code is accessible to users
Powerful, requiring programming knowledge and experience	Simpler, requiring less programming knowledge and experience

Common Uses of JavaScript

- Display a message box
- Edit and validate information entered by the user
- Create a new window with a specified size and screen position
- Display current date
- Simple calculations

Examples of JavaScript Common Uses

- Write directly into the HTML output

```
document.write("<p>This is a paragraph</p>");
```

- Change the content of HTML elements

```
//Find an element
x=document.getElementById("demo")
//Change the content
x.innerHTML="Hello";
```

- React to events

```
<button type="button" onclick="myFunction()">Click
Me!</button>
```

- Test input, and alert about errors

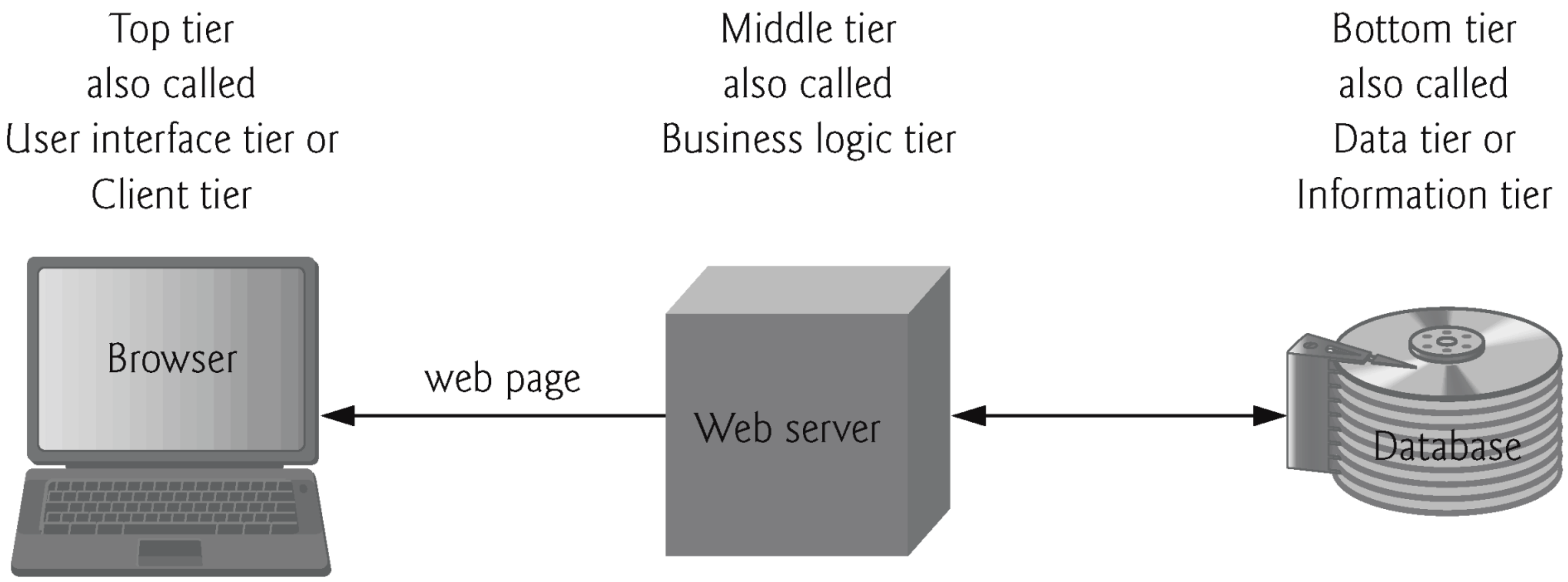
```
if isNaN(x)
{alert("Not Numeric")};
```

General Guidelines to Implement a Feature in Client or Server

For	Use
Pages without conditional text or formatting	Straight HTML
Pages with conditional text or formatting	Client script
Security features	Server-side Programs
Database updates/inserts/deletes	Server-side Programs
Database information formatting and display	Client script or Server-side Programs

http://msdn.microsoft.com/en-us/library/aa239615(v=vs.60).aspx

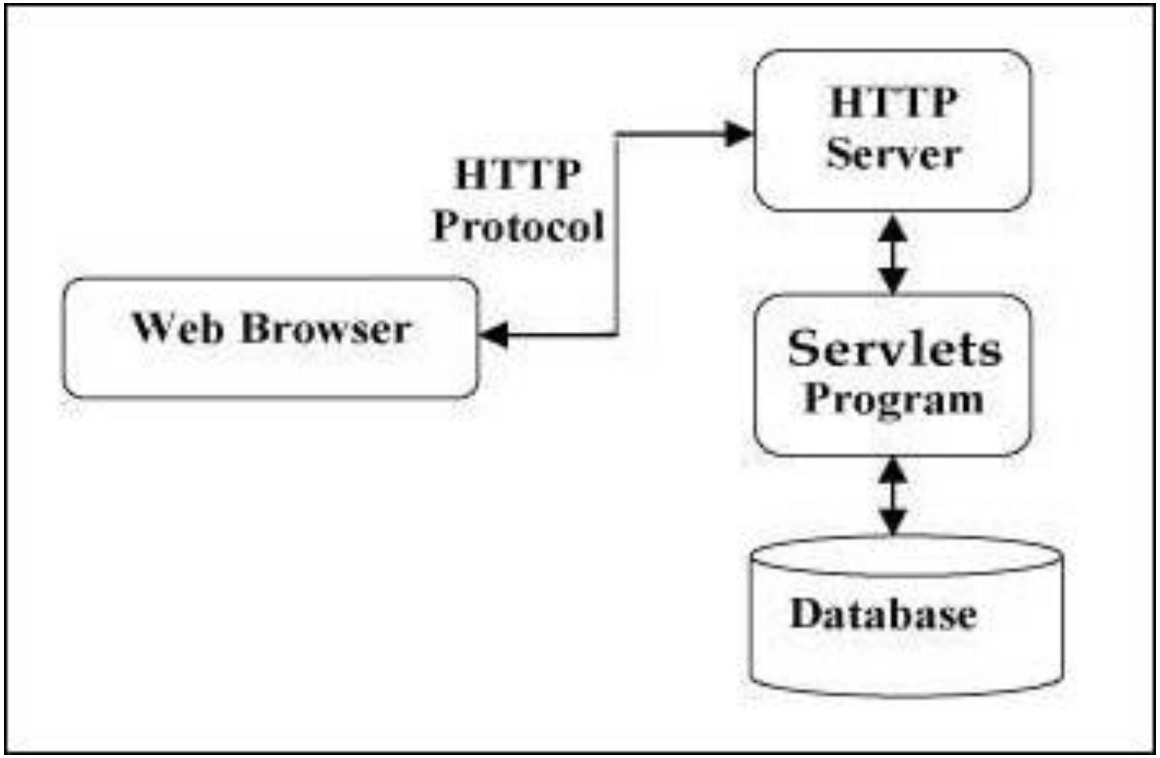
Web Application Architecture



Web Application in Java

- Java Servlet
  - Java programs that run on a server
  - Act as a middle layer between requests from a Web browser and databse
  - Servlets can be created using the **javax.servlet** packages
- Java Server Page
  - Server-side programming technology
  - A type of Java servlet that is designed to fulfill the role of an interface for a Java web application
  - Have access to the entire family of Java APIs, including the JDBC API to access enterprise databases

Web Application in Java



https://www.tutorialspoint.com/servlets/servlets_overview.htm

Development Environment for Web App

- Tomcat: Web server
- Maven in Eclipse
  - A build automation tool used primarily for Java projects
  - Manage all the dependencies (i.e., the libraries that need to be included) of the web project
    - As long as the needed files are specified in pom.xml
  - Create the WAR (Web Application Resource or Web application ARchive) file for installation

Project Object Model (POM)

- The fundamental unit of work in Maven.
- An XML file that contains information about the project and configuration details used by Maven to build the project.
  - Maven will download all the required jar files and make it available to this project
- Default values
  - target: the build directory
  - src/main/java: the source directory

Lab (1)

- Create a Web Application Project

HTTP Get and Post Requests

- The two most common HTTP request types (also known as request methods) are get and post.
  - A get request typically gets (or retrieves) information from a server, such as an HTML document, an image or search results based on a user-submitted search term.
  - A post request typically posts (or sends) data to a server
- For example, when a user performs a search, the web server receives the information specified in the HTML form as part of the request.

Servlet

- Servlet is a Java class used to extend the capabilities of servers that host applications in a request-response model.
  - Request: input
  - Response: output
- Servlet handles the logic to be execute in web server
- Powerful in business logic but not efficient in writing static content

HttpServlet

- A servlet should extend the **`HttpServlet`** class
  - Receives `HttpServletRequest`
  - Replies with `HttpServletResponse`
- A servlet can designate a URL by using the `@WebServlet` annotation
  - e.g., @WebServlet(urlPatterns = "/login.do")
- Handle the client request by responding to get and post requests
  - Override the `doGet()` and `doPost()` method

Lab (2)

- LoginServlet.java

Roles of Servlet and Java Server Page

- Servlet
  - Receives requests (via get or post method)
  - Complete the processing
  - Redirect the results to JSP
    - By setting the attributes of the session in HttpRequest object
    - Dispatch the request to JSP
- JSP
  - HTML-based
  - Display the results
    - Using JavaServer Pages Standard Tag Library (JPST)

Lab (3)

- login.jsp

MVC Design Pattern

- Model
  - Objects the carries data
  - Method to update the data
  - e.g., Object classes
- View
  - Visualization and appearance
  - e.g., Java Server Page
- Controller
  - Controls the data flow and business logic
  - e.g., Servlet

JSP Expression Language

- Makes it possible to easily access application data
- Specify an expression for any of these attribute values
  - ${expr}

Reference

- HTML Tutorial
  http://www.w3schools.com/html/default.asp
- Terry Felke-Morris. *Web Development & Design Foundations with HTML5*, Addison Wesley (2012)
- CSS Tutorial
  https://www.w3schools.com/css/
- JavaScript Tutorial
  https://www.w3schools.com/js/default.asp
- Apache Maven Project
  https://maven.apache.org/guides/introduction/introduction-to-the-pom.html
- Enterprise Edition (Java EE) 7 Documentation
  https://docs.oracle.com/javaee/7/index.html