Advanced Java & Web Technology

(INFO3101)

IT 3RD YEAR 1ST SEMESTER

Module1

Static Web Pages: Web Pages - types and issues, tiers; comparisons of Microsoft and java technologies, WWW Basic concepts, web client and web server, http protocol (frame format), universal resource locator(URL), HTML different tags, sections, sections, image & pictures, pictures, listings, listings, tables, tables, frame, frameset, frameset, form.

Dynamic Web Pages: The need of dynamic web pages; an overview of DHTML, cascading style sheet(css), comparative studies of different technologies of dynamic page creation.

Active Web Pages: Need of active web pages; java applet life cycle, Java Swing.

Web Page

- Web page is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.
- A web page can contain huge information including text, graphics, audio, video and hyper links.

- These hyper links are the link to other web pages.

Static Web page

Static web pages are also known as flat or stationary web page. They are loaded on the client’s browser as exactly they are stored on the web server. Such web pages contain only static information information. User can only read the information information but can’t do any modification or interact with the information.

Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.

Dynamic Web page

Dynamic web page shows different information at different point of time. It is possible possible to Change a portion portion of a web page without loading the entire web page. It has been made possible Using Ajax technology.

- **Server-side dynamic web page**

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include include setting setting up of more client-side processing.

- **Client-side dynamic web page**

  It is processed using client side scripting such as JavaScript. And then passed in to **Document Object Model *DOM.***

- **Scripting Languages**

  Scripting languages are like programming languages that allow us to write programs in form of script. These scripts are interpreted not compiled and executed line by line.

Scripting language is used to create dynamic web pages.

- **Client-side Scripting**

Client-side scripting refers to the programs that are executed on

client-side. Client-side scripts contains the instruction for the browser to be executed in response to certain user's action.
Client-side scripting programs can be embedded into HTML files or also can be Kept as separate files.

Server-side Scripting

Sever-side scripting acts as an interface for the client and also limit the user access the resources on web server. It can also collects the user's characteristics characteristics in order to customize customize response.

Comparisons of Microsoft and Java Technologies

Supported Programming Language.	C#, VB.NET, ASP.NET, & more.	Java, Groovy, PHP, Ruby, Python, JavaScript & more
Works On	Windows Operating System	Any Operating System
Runtime	CLR	JVM
Server Components	NET, COM	EJBs
GUI Components	.NET Class	JavaBeans
Web Services Support	Built-in	Add-On
Unit Testing	Microsoft Unit Testing Framework, NUnit	JUnit
Web Application Framework	ASP.NET MVC, Spring .NET	Spring
Web Server Scripting	ASP.NET	JSF
Access Data By	ADO.NET/oLeDB	JDBC
HTTP Engine	IIS	Application Servers from Multiple Vendors

Hyper Text Transfer Protocol (HTTP)

HTTP

- The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.
- Foundation Foundation for data communication communication for the World Wide Web since 1990
- Generic and stateless protocol .
- TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.)
- Default port is TCP 80.

Features of HTTP

❖HTTP is connectionless

- Browser (Client) initiates an HTTP request.
- Client waits for the response.
- Server processes the request and sends a response back after which client disconnect the connection.
- Client and server knows about each other during current request and response only.
- Further requests are made on new connection like client and server are new to each other.

Features of HTTP

❖HTTP is media independent

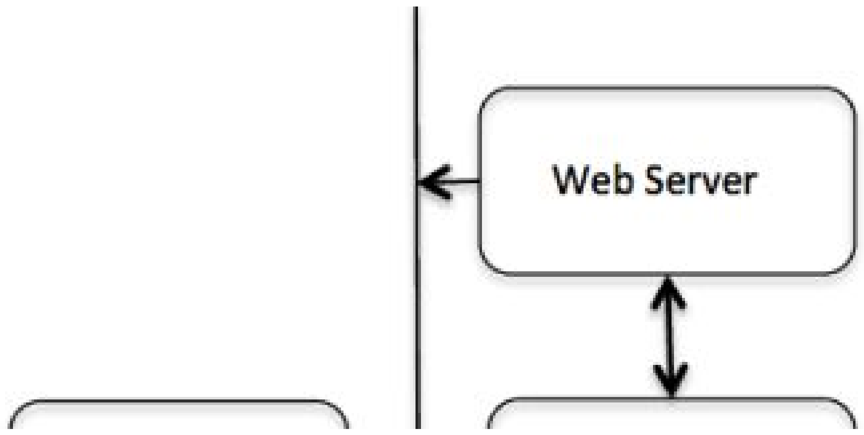
• Any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. • It is required for the client as well as the server to specify the content content type using appropriate appropriate MIME-type.

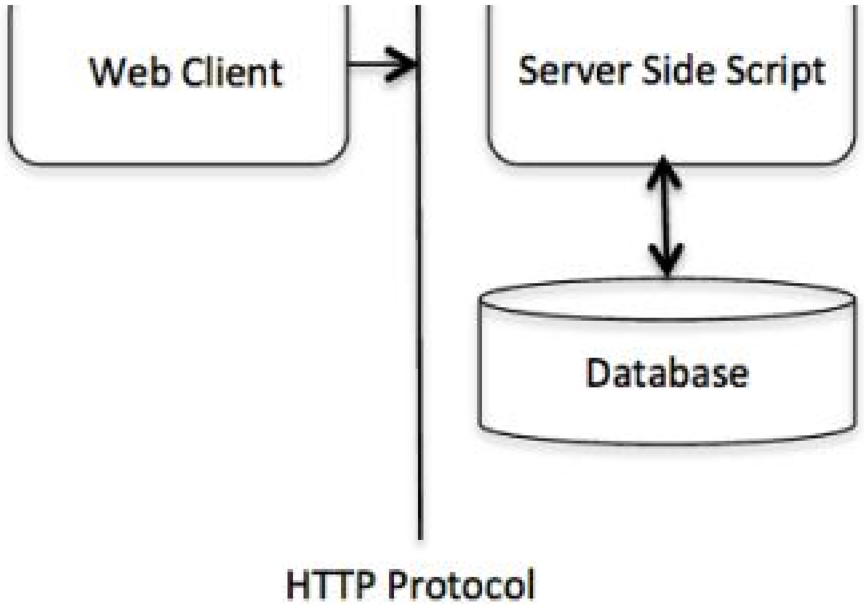
❖HTTP is stateless

- Server and client are aware of each other only during a current request.
- Neither the client nor the browser can retain information between different requests across the web pages.

Basic

Architecture





HTTP Header Files

• **General-header:** These header fields have general applicability for both request and response messages. • **Request-header:** These header fields have applicability applicability only for request request messages messages.

- **Response-header:** These header fields have applicability only for response messages.
- **Entity-header:** An entity header describes the content of the body of the message. Entity headers are used in both, HTTP requests and responses.

HTTP Message

- The message body part is optional for an HTTP message but if it is available, then it is used to carry the entity body associated with the request or response.

HTTP Request

S.N.	Method and Description
GET	The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
HEAD	Same as GET, but it transfers the status line and the header section only.
POST	A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
PUT	Replaces all the current representations of the target resource with the uploaded content.
DELETE	Removes all the current representations of the target resource given by URI.
CONNECT	Establishes a tunnel to the server identified by a given URI.
OPTIONS	Describe the communication options for the target resource.

TRACE Performs a message loop back test along with the path to the target resource.

❖CONNECT Method

- The CONNECT method is used by the client to establish a network connection to a web server over HTTP.

❖OPTIONS Method

- The OPTIONS method is used by the client to find out the HTTP methods and other options supported by a web server.

❖TRACE Method

- The TRACE method is used to echo the contents of an HTTP Request back to the requester which can be used for debugging purpose at the time of development.

URL

(**Uniform Resource Locator)**

Uniform Resource Locator (URL) is the address of a resource on the Internet. URL indicates the location of a resource as well as the protocol used to

access it.

A URL contains the following information:

- The protocol protocol used to a access the resource resource.
- The location of the server (whether by IP address or domain name)
- The port number on the server (optional)
- The location of the resource in the directory structure of the server
- A fragment identifier (optional)

Hyper Text Markup Language(HTML)

HTML

- HTML is the language in which most websites are written. HTML is used to create pages and make them functional. The code used to make them visually appealing is known as CSS.

HTML

Why HTML is called “Hyper Text” and “Markup”?

- Hypertext is text which contains links to other texts and pages. As in HTML page by clicking the link in a page one can navigate to another page, so it is Hyper Text.

- A markup language is a computer language that uses tags to define elements within a document. It is human-readable, meaning markup files contain standard words, rather than typical programming syntax.

Example

```
<!DOCTYPE html>
    <html>
    <head>
    <title>Page Title</title>
    </head>
    <body>

    <h1>This is a
    Heading</h1> <p>This is a
    paragraph.</p>

    </body>
    </html>
```

<!DOCTYPE>

- The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag.

- The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

- In HTML 4.01, the <!DOCTYPE> declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.

HTML Tags

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

HTML tags normally come **in pairs** like <p> and </p>

- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name

HTML Attributes

- All HTML elements can have **attributes** • Attributes provide **additional information** about an element

- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs

like: **name="value"**

▪ The href Attribute

 <a href="https://www.ABC.com">This is a

link</a> ▪ The src Attribute

 <img src=“test.jpg“>

▪ The width and height

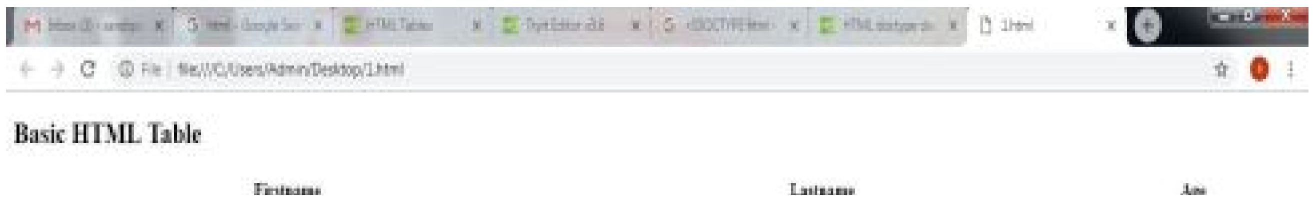
Attributes <img src=“test.jpg" width="500"

height="600"> ▪ The style Attribute

- Title Attribute

HTML Tables

<table



file:///C:/Users/Admin/Desktop/1.html

Basic HTML Table

Firstname	Lastname	Age

```
 style="width:100%"> <tr>
   <th>Firstname</th>
   <th>Lastname</th>
   <th>Age</th>
 </tr>
 <tr>
   <td>Jill</td>
   <td>Smith</td>
   <td>50</td>
 </tr>
 <tr>
   <td>Eve</td>
   <td>Jackson</td>
   <td>94</td>
 </tr>
</table>
```

HTML List

Unordered • Item

- Item
- Item

Ordered

- First item

• Second item

• Item • Third item • Third item

• Item • Fourth item

<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li> </ul>

An unordered HTML list • Coffee

• Tea

• Milk

<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li> </ol>

• **An ordered HTML list** 1.Coffee

2.Tea

3.Milk

HTML frames

- HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document.

- A collection of frames in the browser window is known as a frameset.

Disadvantages of Frames

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different different computers computers due to different different screen resolution.
- The browser's *back* button might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Example

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows = "10%,80%,10%">
<frame name = "top" src = "/html/top_frame.htm" />
<frame name = "main" src = "/html/main_frame.htm" /> <frame name =
    "bottom" src = "/html/bottom_frame.htm" />
<noframes>
    <body>Your browser does not support frames.</body>
</noframes>
</frameset>
</html>
```

FORM

- The HTML <form> element defines a form that is used to collect user input.

```
<form>
.
form elements
.
</form>
```

The <input> Element

<input type="text">	Defines a one-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

Example

```
<form>

  First name:<br>
 <input type="text" type="text"
  name="firstname firstname"><br> Last
  name:<br>
 <input type="text"
```

name="lastname"> </form>

<form>

<input type="radio" name="gender" value="male" checked>

Male<br> <input type="radio" name="gender" value="female">

Female<br> <input type="radio" name="gender" value="other">

Other </form>

Dynamic Web Page

- A dynamic web page is a web page that displays different content each time it's viewed. For example, the page may change with the time of day, the user that accesses the webpage, or the type of user interaction. There are two types of dynamic web pages.

 CLIENT-SIDE SCRIPTING

Web pages that change in response to an action within that web page, such as a mouse or a keyboard action, use client-side scripting.

Client-side scripts generate client-side content. Client-side content is content that's generated generated on the user's computer computer rather than the server. In these cases, the user's web browser would download the web page content from the server, process the code that's embedded in the web page, and then display the updated content to the user.

Scripting languages such as JavaScript and Flash allow a web page to respond to client-side events.

 SERVER-SIDE SCRIPTING

Web pages that change when a web page is loaded or visited use server-side scripting. Server-side content is content that's generated when a web page is loaded. For example, login pages, forums, submission forms, and shopping carts, all use server-side scripting since those web pages change according to what is submitted to it.

DHTML

- Dynamic HyerText Markup Language (**DHTML**) is a combination of Web development technologies used to create dynamically changing changing websites websites. Web pages may include include animation, dynamic menus and text effects. The technologies used include a combination of HTML, JavaScript or VB Script, CSS and the document object model (DOM).

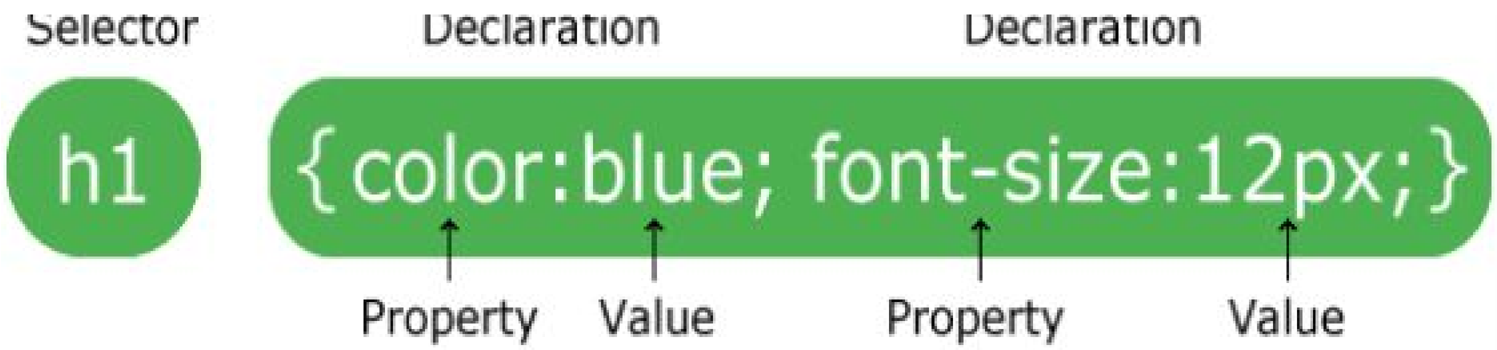
Cascading Style Sheet (CSS)

- **CSS** stands for **C**ascading **S**tyle **S**heets • CSS describes **how HTML elements are to be displayed on screen, paper, or in other media** • CSS **saves a lot of work**. It can control the layout of multiple multiple web pages all at once
- External stylesheets are stored in **CSS files** *The*

name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

CSS syntax

- A CSS rule-set consists of a selector and a declaration block:



```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```

CSS Selectors

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

The element Selector

The element selector selects elements based on the element name.

The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element! element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- The style rule below will be applied to the HTML element with id="para1":

```
#para1 {
  text-align: center;
```

```
        color: red;
      }
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

The class Selector

- The class selector selects elements with a specific class attribute.

- To select elements with a specific class, write a period (.) character, followed by the name of the class.

- In the example below, all HTML elements with

class="center" will be red and center-aligned:

.center{

text-align: center;

: red;

}

color

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

• We can also specify that only specific HTML elements should be affected by a class. • In the example below, only <p> elements with class="center" class="center" will be center-aligned aligned: • p.center {

```
    text-align: center;
    color: red;
  }
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1> <p
class="center">This paragraph will be red and center-aligned.</p>

</body>
</html>
```

- In the example below, the <p> element will be styled according to class="center" and to class="large":

<p class="center class="center large">This large">This paragraph paragraph refers to two

classes.</p>

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}

p.large {
  font-size: 300%;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>

</body>
</html>
```

Grouping Selectors

- If you have elements with the same style definitions, like this:

```
h1, h2, p {
    text-align: center;
    color: red;
  }
```

CSS

External Style Sheet

- With an external style sheet, we can change the look of an entire website by changing just one file!
- Each page must include a reference to the external external style sheet file inside the <link>

  element element. The <link> element goes inside

  the <head> section:

- <head>
  <link rel="stylesheet" type="text/css"

href="mystyle.css"> </head>

- <u>mystyle.css</u>

```
body {
  background-color:
lightblue; }

h1 {
  color: navy;
  margin-left: 20px;
}
```

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the <style> element, inside the <head> section

of an HTML page:

```
<head>
    <style>
    body {
      background-color:
    linen; }

    h1 {
      color: maroon;
      margin-left:
    40px; }</style>
</head>
```

Inline Styles

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the

relevant element. The style attribute can contain any CSS property.

- The example below shows how to change the color and the left margin of a <h1> element:

<h1 style="color:blue;margin-left:30px;">This is a heading</h1>

Multiple Style Sheets

- Assume that an external style sheet has the following style for the <h1> element:

```
h1 {color: navy;
   }
```

then, assume that an internal style sheet also has the following style for the <h1> element: h1 {color: orange;
}

- If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange":

```
<head>
  <link rel="stylesheet" type="text/css"

  href="mystyle.css" ><link rel="stylesheet"
```

```
    type="text/css" href="mystyle.css"

    >
     <style>
    h1 {
      color: orange;
    }
    </style>
</head>
```

If the internal style is defined before the link to the external style sheet, the <h1> elements will be "navy":

```
<head>
    <style>
    h1 {
      color: orange;
    }</style>
    <link rel="stylesheet" type="text/css"
    href="mystyle.css" >
</head>
```

Cascading Order

What style will be used when there is more than one style specified for an HTML element? • All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest highest priority priority:

• Inline style (inside an HTML element) • External and internal style sheets (in the head section)

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

 Background Color

```
<h1 style="background-color:Orange;">Hello
World</h1> <p
style="background-color:Tomato;">HI...</p>
```

Text Color

```
<h1 style="color:Tomato;">Hello
World</h1> <p style="color:Orange...</p>
```

CSS Backgrounds

CSS background properties:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

```
body {
    background-color:
  lightblue; }


h1 {
```

background-color:
green; }

Background Image

- body {
  background-image:
  url("paper.gif"); }

- body {
  background-image:
  url("img_tree.png");
  background-repeat: no-repeat;

```
}
```