

Web Technology (PS02CMCA53)

Unit 1: Client-side Web Technologies – I

Introduction to HTTP and HTML5:

HTTP:

HTTP is a protocol which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser.

(HyperText Transfer Protocol) The communications protocol used to connect to Web servers on the Internet or on a local network (intranet). The primary **function** of **HTTP** is to establish a connection with the server and send HTML pages back to the user's browser.

It is stateless protocol.

OR

HTTP is the protocol used to transfer data over the web. It is part of the Internet protocol suite and defines commands and services used for transmitting webpage data. **HTTP** uses a server-client model. A client, for example, may be a home computer, laptop, or mobile device.

HTML5:

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and last major HTML version that is a World Wide Web Consortium recommendation. The current specification is known as the HTML Living Standard.

HTML5 is a programming language whose acronym stands for Hyper Text Markup Language. It is a system that allows the modification of the appearance of web pages, as well as making adjustments to their appearance. It also used to structure and present content for the web.

HTML5 is the latest version of HTML programming that allows better management of the web application or the website contents. While HTML doesn't allow support for

Video and Audio data in the programming language, **HTML5 allows** any kind of data to be incorporated into the program.

HTML5 may contain HTML, JavaScript, CSS.

URL format:

A URL is another word for a web address.

Web browsers request pages from web servers by using a URL.

A Uniform Resource Locator (URL) is used to address a document (or other data) on the web.

- **scheme** - defines the **type** of Internet service (most common is **http or https**)
- **prefix** - defines a domain **prefix** (default for http is **www**)
- **domain** - defines the Internet **domain name** (like google.com)
- **port** - defines the **port number** at the host (default for http is **80**)
- **path** - defines a **path** at the server (If omitted: the root directory of the site)
- **filename** - defines the name of a document or resource

HTML5 document structure:



DOCTYPE - A basic HTML page starts with the Document Type Declaration or doctype. That is a way to inform the browser what type of document it is. The doctype is always the first item at the top of any HTML file. Then sections and subsections come, each possibly has its heading and subheading. These heading and sectioning elements help the reader to perceive the content meaning.

The <html> element

The `<html>` element follows the doctype information, which is used to inform the browser that this is an HTML document.

he <head> section

The next part is the `<head>` section. The `<head>` element contains metadata (document title, character set, styles, links, scripts), specific information about the web page that is not displayed to the user.

The <body> element

The `<body>` of a document contains the content of the document. The content may be presented by a user agent in different ways. E.g., the content can be text, images, links, colors, graphics, etc.

Headers:

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

Example:

A header for an <article>:

```
<article>
  <header>
    <h1>A heading here</h1>
    <p>Posted by John Doe</p>
    <p>Some additional information here</p>
  </header>
  <p>Lorem Ipsum dolor set amet....</p>
</article>
```

Body:

The <body> tag defines the document's body.

The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

Example:

```
<html>
<head>
  <title>Title of the document</title>
</head>

<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>

</html>
```

Declarations:

All HTML documents must start with a <!DOCTYPE> declaration.

The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document.....
</body>

</html>
```

HTML elements:

An HTML element is defined by a start tag, some content, and an end tag.

The HTML **element** is everything from the start tag to the end tag:

`<tagname>Content goes here...</tagname>`

Examples of some HTML elements:

`<h1>My First Heading</h1>`

`<p>My first paragraph.</p>`

Element ID:

The HTML **id** attribute is used to specify a unique id for an HTML element.

You cannot have more than one element with the same id in an HTML document.

Using The id Attribute:

The **id** attribute specifies a unique id for an HTML element. The value of the **id** attribute must be unique within the HTML document.

For example:

`<h1 id="head">Demo</h1>`

Name:

The name attribute specifies a name for an HTML element.

For example: `<input type="text" name="txttest" />`

Attributes: HTML attributes provide additional information about HTML elements.

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: `name="value"`

For Example:

```
<a href="https://www.google.com">Visit Search Engine</a>

<p style="color:red;">This is a red paragraph.</p>
```

HTML Events:

HTML has the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.

1. **Onclick:** A function is triggered when the button is clicked.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<button onclick="myFunction()">Click me</button>
```

```
<p id="demo"></p>
```

```
<p>A function is triggered when the button is clicked. The function outputs some text
in a p element with id="demo".</p>
```

```
<script>
```

```
function myFunction()
```

```
{
```

```
    document.getElementById("demo").innerHTML = "Hello World";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

2. Ondblclick: A function is triggered when the button is double-clicked.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<button onclick="myFunction()">Double-click me</button>
```

```
<p id="demo"></p>
```

<p>A function is triggered when the button is double-clicked. The function outputs some text in a p element with id="demo".</p>

```
<script>
```

```
function myFunction() {
```

```
    document.getElementById("demo").innerHTML = "Hello World";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```


3. OnFocus: A function is triggered when one of the input fields get focus.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

<p>A function is triggered when one of the input fields get focus. The function changes the background-color of the input field.</p>

First name: <input type="text" id="fname" onfocus="myFunction(this.id)">

Last name: <input type="text" id="lname" onfocus="myFunction(this.id)">

```
<script>
```

```
function myFunction(x) {
```

```
    document.getElementById(x).style.background = "yellow";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

4. On Submit: When you submit the form, a function is triggered which alerts some text.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

<p>When you submit the form, a function is triggered which alerts some text.</p>

```
<form action="/action_page.php" onsubmit="myFunction()">
```

```
  Enter name: <input type="text" name="fname">
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

```
<script>
```

```
function myFunction() {
```

```
  alert("The form was submitted");
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

5. **Onblur:** When you leave the input field, a function is triggered.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
  Enter your name: <input type="text" name="fname" id="fname"
  onblur="myFunction()">
```

<p>When you leave the input field, a function is triggered which transforms the input text to upper case.</p>

```
<script>

function myFunction() {

    var x = document.getElementById("fname");

    x.value = x.value.toUpperCase();

}

</script>


</body>

</html>
```

HTML Forms:

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The HTML `<form>` element is used to create an HTML form for user input:

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

elements

Example:

```
<form>
  <label for="fname">First name: </label> <br>
  <input type="text" id="fname" name="fname"> <br>
  <label for="lname">Last name: </label> <br>
  <input type="text" id="lname" name="lname">
</form>
```

HTTP Verbs:

- GET: It is a default method. Passes user input with URL query string. That is why, it is not considered as secure method.
- POST: It is not a default method. It never pass user input with URL query string. That is why, it is considered as secure method.

HTML5 media:

Multimedia on the web is sound, music, videos, movies, and animations.

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

How it Works:

The **controls** attribute adds video controls, like play, pause, and volume.

It is a good idea to always include **width** and **height** attributes. If height and width are not set, the page might flicker while the video loads.

The **<source>** element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the **<video>** and **</video>** tags will only be displayed in browsers that do not support the **<video>** element.

HTML **<video>** Autoplay

To start a video automatically, use the **autoplay** attribute:

```
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
```

```
<source src="movie.ogg" type="video/ogg">
```

Your browser does not support the video tag.

```
</video>
```

Add **muted** after **autoplay** to let your video start playing automatically (but muted):

```
<video width="320" height="240" autoplay muted>
```

```
<source src="movie.mp4" type="video/mp4">
```

```
<source src="movie.ogg" type="video/ogg">
```

Your browser does not support the video tag.

```
</video>
```

The HTML **<audio>** element is used to play an audio file on a web page.

The HTML **<audio>** Element

To play an audio file in HTML, use the **<audio>** element:

```
<audio controls>
```

```
<source src="horse.ogg" type="audio/ogg">
```

```
<source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

HTML Audio - How It Works

The **controls** attribute adds audio controls, like play, pause, and volume.

The **<source>** element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the **<audio>** and **</audio>** tags will only be displayed in browsers that do not support the **<audio>** element.

HTML <audio> Autoplay

To start an audio file automatically, use the **autoplay** attribute:

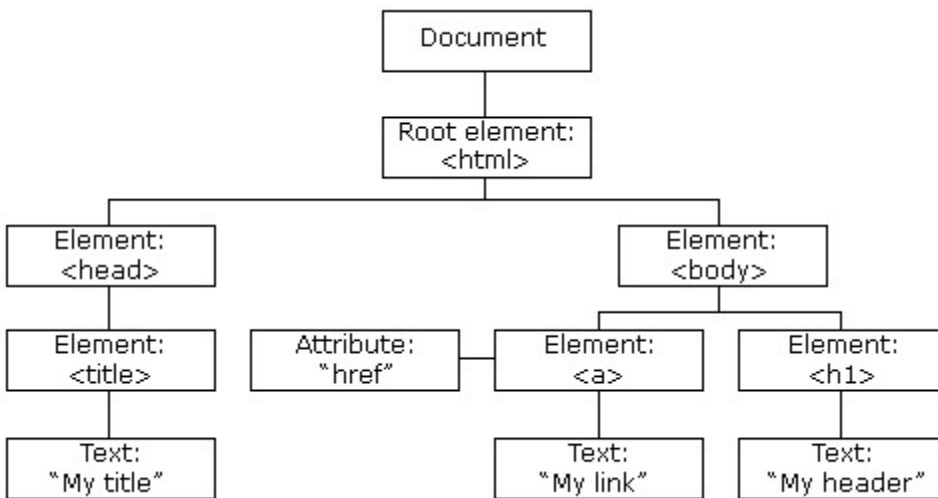
```
<audio controls autoplay>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

Add **muted** after **autoplay** to let your audio file start playing automatically (but muted):

```
<audio controls autoplay muted>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

Introduction to HTML DOM:

The Document Object Model (DOM) is a **programming interface for HTML and XML(Extensible markup language) documents**. It defines the logical structure of documents and the way a document is accessed and manipulated.



Introduction to CSS3:

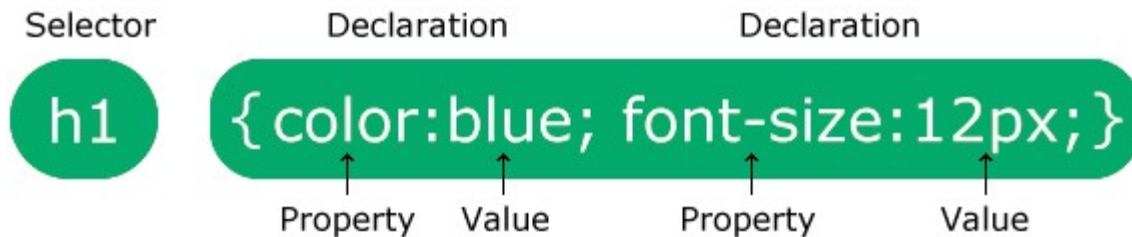
What is CSS?

- CSS stands for Cascading Style Sheets
- 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 web pages all at once
- External stylesheets are stored in CSS files

CSS3 Syntax:

A CSS rule consists of a selector and a declaration block.

CSS Syntax as below:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS (Different properties, value, units and specifying colors):

Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
  color: red;
```

```
  text-align: center;
```

```
  font-size:50px;
```

```
}
```


</style>

</head>

<body>

<p>Hello World!</p>

<p>These paragraphs are styled with CSS.</p>

</body>

</html>