**TUTORIAL 4 – HTML 5 & WEB Basics**

Repo link: <https://github.com/tinkerness/S7_Web_Programming/tree/main/tutorials/tut4>

1. **What is a URL? Explain the different parts of URL with example.**

A URL, which stands for Uniform Resource Locator, is a reference or address used to access resources on the internet. It specifies the location and access method for a resource, such as a web page, file, or service. URLs are essential for navigating the web and identifying specific resources online. A typical URL consists of several components, each serving a specific purpose:

1. **Scheme (or Protocol):** This part specifies the protocol or method used to access the resource. Common schemes include "http," "https," "ftp," "mailto," "tel," and more. The scheme is followed by a colon and two forward slashes (e.g., https://). Example: <https://www.example.com> . In this case, https is the scheme, indicating that a secure HTTP connection is used.
2. **Domain (or Host):** The domain name or IP address identifies the server where the resource is hosted. It may also include a subdomain (e.g., "www") and multiple levels of domains (e.g., "example.com"). Example: <https://www.example.com> . Here, www.example.com is the domain.
3. **Port**: The port number, if specified, follows the domain and is separated by a colon. It indicates the specific port on the server that should be used for the connection. Ports are usually assigned default numbers for specific services (e.g., HTTP uses port 80, HTTPS uses port 443). Example: <https://www.example.com:8080> . In this case, 8080 is the port number.
4. **Path**: The path specifies the location of the resource on the server's file system or within the web application. It typically starts with a forward slash and may include subdirectories and the filename. Example: <https://www.example.com/products/electronics/laptop.html> . Here, /products/electronics/laptop.html is the path, indicating the location of a specific HTML file.
5. **Query Parameters**: Query parameters are often included after a question mark "?" in the URL. They are used to pass data to the server as key-value pairs. Multiple parameters are separated by ampersands "&." Example: <https://www.example.com/search?q=example&page=1> . In this URL, q=example and page=1 are query parameters used for search and pagination.
6. **Fragment Identifier (or Anchor):** A fragment identifier is indicated by a hash "#" symbol followed by a name or identifier. It points to a specific section within a web page. When the URL is opened, the browser will scroll to the identified section. Example: <https://www.example.com/page#section2> . Here, #section2 points to a specific section within the web page.
7. **Explain HEAD, PUT and POST HTTP methods.**

**HEAD Method:**

* The HEAD method is used to retrieve only the headers (metadata) of a resource, not the actual content.
* It's useful for checking resource availability and getting information like content type and length.
* It is a read-only and safe method, meaning it doesn't modify data on the server.
* Example:

HEAD /example/resource HTTP/1.1

Host: www.example.com

**PUT Method:**

* PUT is used to update or create a resource at a specific URL.
* It sends the full representation of the resource in the request body to update the resource or create it if it doesn't exist.
* PUT requests are idempotent, meaning multiple identical requests should have the same effect.
* Example:

PUT /example/resource HTTP/1.1

Host: www.example.com

Content-Type: application/json

Content-Length: 123

{"key": "updated\_value"}

**POST Method:**

* POST is a versatile method used to submit data to a server for various purposes.
* It doesn't specify the exact operation to perform on the resource; it's often used for form submissions, data processing, and creating new resources.
* POST requests are not idempotent, meaning multiple identical requests may have different effects.
* Example:

POST /example/resource HTTP/1.1

Host: www.example.com

Content-Type: application/x-www-form-urlencoded

Content-Length: 25

key1=value1&key2=value2

1. **Distinguish between cell spacing and cell padding attributes of table tag with a suitable example.**

**Cell Spacing (cellspacing):**

* cellspacing defines the space between the individual cells (table data cells and table header cells) of the table.
* It adds space between adjacent cells, both horizontally and vertically.
* cellspacing is specified as a non-negative integer value (measured in pixels or other length units) or simply as "0" to indicate no spacing between cells.
* Example :

<table cellspacing="10">

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

</tr>

<tr>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</table>

In this example, there will be a 10-pixel space between adjacent cells.

**Cell Padding (cellpadding):**

* cellpadding defines the space between the content of each cell and the cell's border.
* It controls the padding within each cell, effectively adding space between the content and the cell's border.
* Like cellspacing, cellpadding is specified as a non-negative integer value (measured in pixels or other length units) or "0" for no padding.
* Example :

<table cellpadding="5">

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

</tr>

<tr>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</table>

In this example, there will be a 5 pixels of padding inside each cell.

To summarize, cellspacing controls the space between cells in a table, while cellpadding controls the space between the content within each cell and the cell's border.

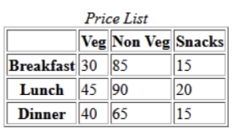
1. **Describe the use of MIME by explaining the various type and subtypes.**

**MIME (Multipurpose Internet Mail Extensions)** is a standard that extends the format of email messages and other internet content beyond simple text. MIME types (also known as media types or content types) are used to label and describe the nature of data files. MIME types are a way of specifying what kind of data a file contains, so that software applications can handle the data appropriately.

MIME types are composed of two parts: a type and a subtype, separated by a slash (/). The type indicates the general category of data, while the subtype provides more specific information about the data within that category. Here are some common MIME types and their subtypes:

1. **Text Types:**
   * **text/plain**: Unformatted text, typically containing human-readable content.
   * **text/html**: Hypertext Markup Language, used for web pages.
   * **text/css**: Cascading Style Sheets, used for styling web documents.
   * **text/javascript**: JavaScript code for client-side scripting.
   * **text/xml**: Extensible Markup Language, used for structured data.
2. **Image Types:**
   * **image/jpeg**: JPEG image format, commonly used for photos.
   * **image/png**: Portable Network Graphics, used for lossless image compression.
   * **image/gif**: Graphics Interchange Format, often used for animations.
   * **image/svg+xml**: Scalable Vector Graphics, for vector images.
3. **Audio Types:**
   * **audio/mpeg**: MPEG audio format, commonly used for music.
   * **audio/wav**: Waveform Audio File Format, for uncompressed audio.
   * **audio/ogg**: Ogg Vorbis audio format, an open and free audio codec.
4. **Video Types:**
   * **video/mp4**: MPEG-4 video format, widely used for video streaming.
   * **video/webm**: WebM video format, an open and royalty-free format.
   * **video/ogg**: Ogg Theora video format, often used with Ogg audio.
5. **Application Types:**
   * **application/json**: JavaScript Object Notation, used for data exchange.
   * **application/pdf**: Portable Document Format, for documents.
   * **application/xml**: XML data format, often used for configuration files.
   * **application/zip**: ZIP archive format, for compressing files.
6. **Other Types:**
   * **multipart/form-data**: Used for file uploads in HTML forms.
   * **message/rfc822**: Format for email messages.
   * **multipart/alternative**: Used for alternative representations of email content (e.g., plain text and HTML versions).

MIME types help ensure that software applications can correctly interpret and display data, making them crucial for web browsers, email clients, and other internet-related services. They are specified in the **Content-Type** header of internet messages to inform recipients about the type of data being transmitted.

1. **Write the code for the following output: **

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Price List</title>

</head>

<body>

    <table border="1">

        <caption><em>Price List</em></caption>

        <tr>

            <th></th>

            <th>Veg</th>

            <th>Non Veg</th>

            <th>Snacks</th>

        </tr>

        <tr>

            <th>Breakfast</th>

            <td>30</td>

            <td>85</td>

            <td>15</td>

        </tr>

        <tr>

            <th>Lunch</th>

            <td>45</td>

            <td>90</td>

            <td>20</td>

        </tr>

        <tr>

            <th>Dinner</th>

            <td>40</td>

            <td>65</td>

            <td>15</td>

        </tr>

    </table>

</body>

</html>

1. **Demonstrate the use of size and multiple attributes for select tag with help of code. A selection list containing four items, “CS201”, “CS202”, “CS203”, “CS204”, first two of which are always visible. Draw the relevant output also.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Selection List</title>

</head>

<body>

<h3>The select size attribute</h3>

<label for="courses">Select Courses:</label>

<select id="courses" name="courses" size="4" multiple>

<option value="CS201">CS201</option>

<option value="CS202">CS202</option>

<option value="CS203">CS203</option>

<option value="CS204">CS204</option>

</select>

</body>

</html>

1. **Write HTML code to implement a form which has the following elements:**

**i. A textbox which has label “NAME” accept a maximum of 30 characters.**

**ii. A textbox which has label “PASSWORD” accept a maximum 8 characters and it mask characters when user enters**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<form action="#">

<label for="fname">NAME: </label><br>

<input type="text" id="fname" name="fname" maxlength="30" required />

<br><br>

<label for="fname">PASSWORD: </label><br>

<input type="password" id="fname" name="fname" maxlength="8" required/>

<br><br>

<input type="button" value="submit"/>

</form>

</body>

</html>

1. **Why do you call MIME as an extension feature? Justify with suitable statements**

MIME is an e-mail extension protocol, i.e., it does not operate independently, but it helps to extend the capabilities of e-mail in collaboration with other protocols such as SMTP. MIME is a kind of add-on or a supplementary protocol that allows non-ASCII data to be sent through SMTP. It allows the users to exchange different kinds of data files on the Internet: audio, video, images, application programs as well.

Features of MIME Protocol

1. It supports multiple attachments in a single e-mail.
2. It supports the non-ASCII characters.
3. It supports unlimited e-mail length.
4. It supports multiple languages.

Advantage of the MIME

The MIME protocol has the following advantages:

1. It is capable of sending various types of files in a message, such as text, audio, video files.
2. It also provides the facility to send and receive emails in different languages like Hindi, French, Japanese, Chinese etc.
3. It also provides the facility of connecting HTML and CSS to email, due to which people can design email as per their requirement and make it attractive and beautiful.
4. It is capable of sending the information contained in an email regardless of its length.
5. It assigns a unique id to all e-mails.
6. **Explain following html tags with proper example.**
7. **<textarea> 2. <span> 3. <tr> 4. <form> 5. <a>**

**<textarea> Tag:**

* The <textarea> tag is used to create a multi-line text input field within an HTML form.
* It allows users to input and edit multiple lines of text, such as comments or descriptions.
* Example:

<form>

<label for="comments">Comments:</label>

<textarea id="comments" name="comments" rows="4" cols="50"></textarea>

</form>

**<span> Tag:**

* The **<span>** tag is an inline HTML tag used to apply styles or manipulate individual sections of text within a larger text block.
* It doesn't add any line breaks or structural changes; it's primarily used for styling or scripting purposes.
* Example:

<p>This is <span style="color: blue;">blue</span> text.</p>

**<tr> Tag:**

The <tr> tag is used to define a table row within an HTML table (<table>).

It is used in combination with <td> (table data) or <th> (table header) elements to create table cells within the row.

* Example:

<table>

<tr>

<th>Header 1</th>

<th>Header 2</th>

</tr>

<tr>

<td>Data 1</td>

<td>Data 2</td>

</tr>

</table>

**<form> Tag:**

* The **<form>** tag is used to create an HTML form that allows users to input data and submit it to a server for processing.
* It acts as a container for various form elements like text fields, checkboxes, radio buttons, and buttons.
* Example:

<form action="/submit" method="post">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<br>

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

</form>

**<a> Tag:**

* The **<a>** tag (anchor tag) is used to create hyperlinks, allowing users to navigate to other web pages, resources, or locations.
* It is used with the **href** attribute to specify the destination URL.
* Example:

<p>Visit our <a href="https://www.example.com">website</a> for more information.</p>

1. **Write the equivalent HTML code to implement the following in a web page:**

**• An image titled “flower.jpg” with proper attribute to set height, width and message text.**

**• Unordered list with values tea, coffee and milk**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Example</title>

</head>

<body>

    <h2>Image</h2>

    <img src="flower.jpg" alt="Flowers" width = "300px" height="200px">

    <h2>Beverage List</h2>

    <ul>

        <li>Tea</li>

        <li>Coffee</li>

        <li>Milk</li>

    </ul>

</body>

</html>