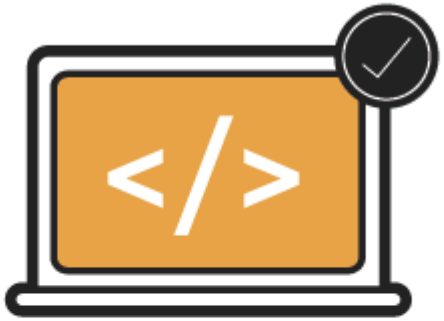
ACADGILD



FRONT END WEB DEVELOPMENT FUNDAMENTALS

Website : http://www.acadgild.com
LinkedIn : https://www.linkedin.com/company/acadgild
Facebook : https://www.facebook.com/acadgild

Brief Intro About AcadGild: CEO – Vinod Dham, Father of Pentium

- **AcadGild** is a technology education start-up which provides online courses in latest technologies.

ANDROID DEVELOPMENT

BIG DATA DEVELOPMENT

FRONT END DEVELOPMENT (WITH ANGULARJS)

CLOUD COMPUTING

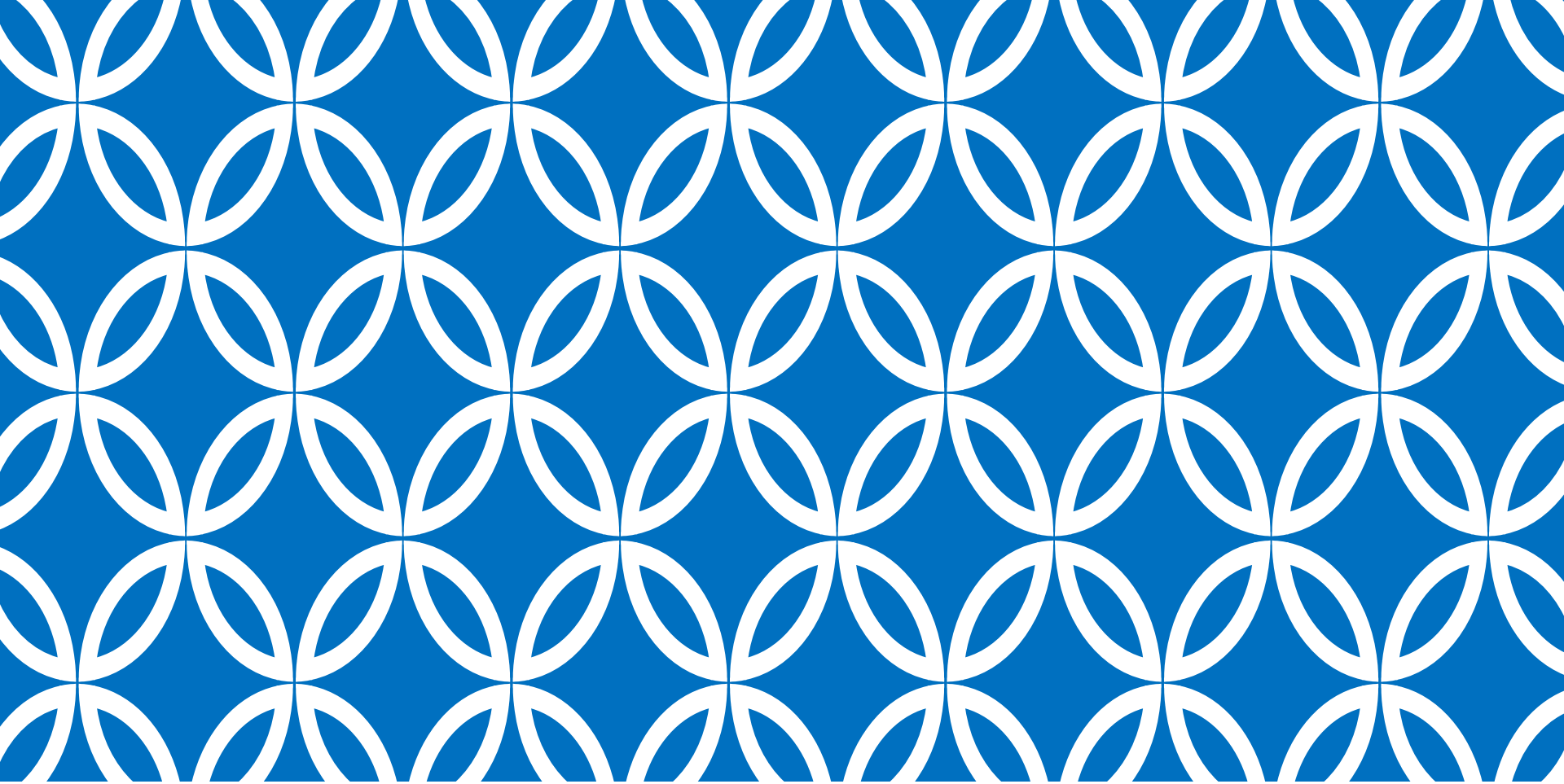
- **AcadGild** was started by IIT/IIM alumni.
- **Our aim** is to provide millions of high school graduates, college graduates and working professionals, skills to make them ready for jobs.

Course Objectives

- Design Website using HTML (**H**yper **T**ext **M**arkup **L**anguage).
- Style the Website developed in HTML using CSS (**C**ascading **S**tyle **S**heets).
- Develop JavaScript codes.

Course Modules

1. HTML (Hyper Text Markup Language)
2. CSS (Cascading Style Sheets)
3. CSS3
4. JavaScript
5. HTML 5



Session 1 – HTML Basics

Agenda – HTML Basics

Sl No	Agenda Title
1	How Web Works
2	Name Server
3	Client Server
4	Understanding the URL
5	HTTP
6	HTTPs
7	Full Web Request Cycle
8	Request Response Cycle

Sl No	Agenda Title
9	Server Side Vs Client Side Languages
10	Languages Vs Framework
11	HTML
12	Tags, Attributes & Elements
13	Basic HTML Syntax
14	Basic HTML Tags
15	Doctype tag
16	Hello World Using HTML

How Web Works

- **To Understand web,** let us understand how a computer works.

  Scenario:

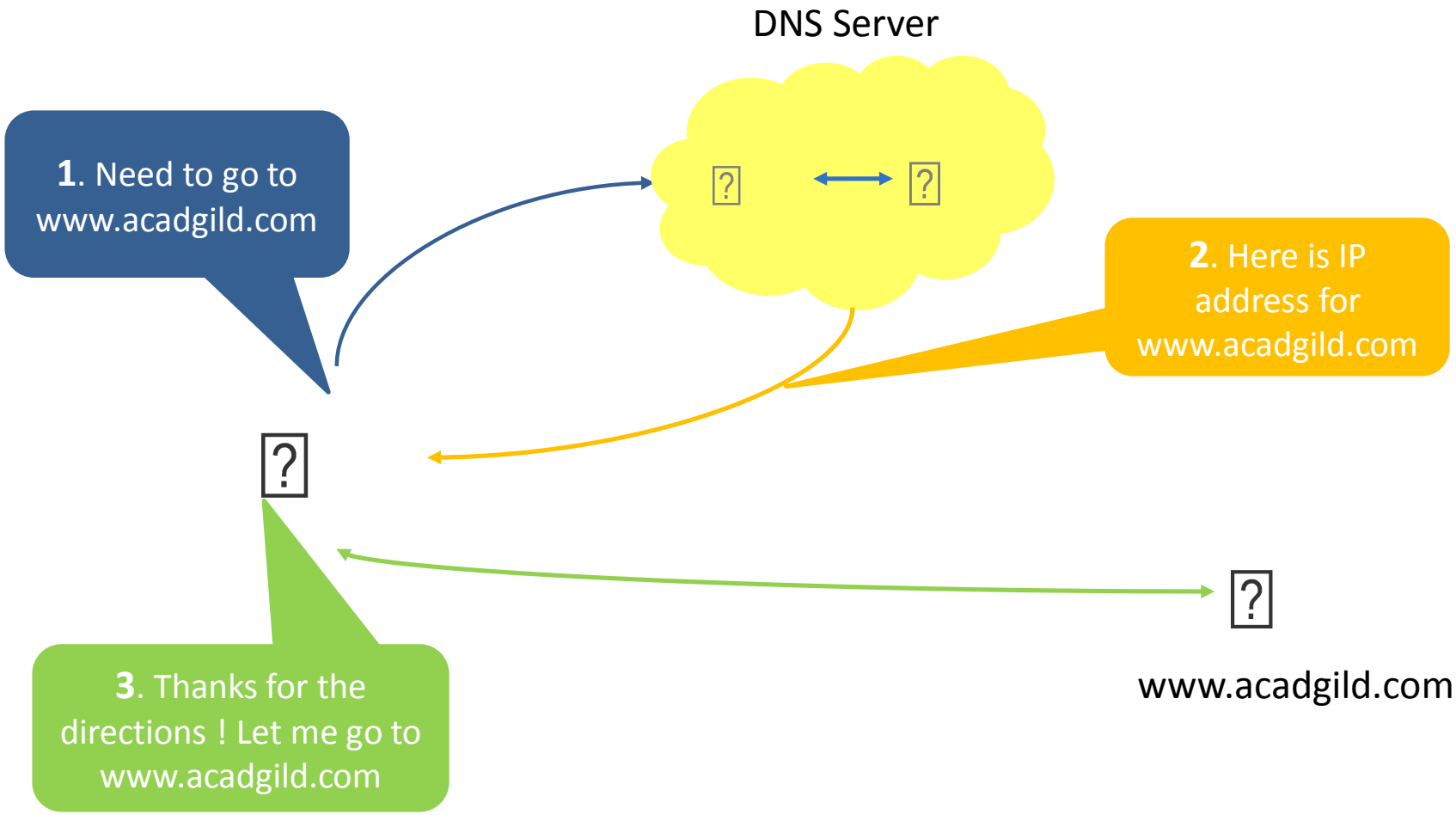
  - We want to access a document named details.pdf kept in C:/docs/details.pdf of Machine1 -> Browse through the folder's path and access it.
  - Let's say we want to access this document from a different machine, say Machine2. We assume sharing is enabled on Machine1. In that case we will need to know the IP address of Machine1.
  - Go to Explorer Address Bar -> Type the IP address -> Click on folder and access the details.pdf.

  Note: Remote machine's IP address is required to connect to Machine2 and access the document.

How Web Works (contd.)

- **Web works** in the same manner as explained above.
- Documents like images, HTML files etc. are accessed from a remote machine with the help of browser.
- **Browser** does the job of connecting to the remote machine.
- Each machine is recognized with unique **IP address**.
- It is difficult to remember all IP addresses. Therefore instead of accessing the remote machine via IP address, we access them using a domain name(example: www.acadgild.com).
- The task of converting domain name to IP address is taken care by separate computers called **Name servers**.
- **Name Servers** are a cluster of servers containing domain to IP address mapping.

Name Server


DNS Server
1. Need to go to www.acadgild.com
2. Here is IP address for www.acadgild.com
3. Thanks for the directions ! Let me go to www.acadgild.com
www.acadgild.com

Client Server

- **Servers** - Machines that provide service to other machines.
- **Clients** - Machines that request data from them.
- In previous example Machine 2 becomes the client and Machine 1 becomes the server.

Understanding the URL

Example of URL: *http://acadgild.com/tutorials.php*

- **The URL begins with http** which is the protocol for communication. This is the normal mode of communication. If we want a secure communication, we should use “https” protocol. More about http in next slide.
- **Next component is acadgild.com** which is the domain name.
- A Domain name comprises of two parts - name and extension. For acadgild.com, the name is acadgild and extension is .com.
- **Next is the actual file requested namely tutorials.php** (note that some webservers mask these files for security).
- In tutorials.php the extension **php** represents the server side language.

Note that many sites hide the extension or change the URLs for security purposes.

HTTP

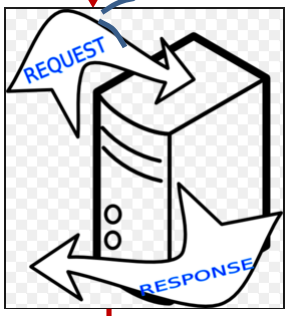
- **HTTP** stands for ***H***yper ***T***ext ***T***ransfer ***P***rotocol.
- **HTTP** is the protocol used by World Wide Web or in other words by Internet.
- **HTTP** defines how messages are formatted and transmitted
- **HTTP** also defines what actions Web servers and browsers should take in response to various commands.
- When a URL is entered in browser it sends a **HTTP** command to the web server directing it to fetch and transmit the requested web page.
- **HTTP** is also called as a stateless protocol because each command is executed independently, without any knowledge of the commands that it encounters.

HTTPS

- **HTTPS** stands for **H**yper**T**ext **T**ransfer **P**rotocol **S**ecure
- **HTTPS** is a protocol for secure communication over a computer network which is widely used on the internet
- **HTTPS** consists of communication over HTTP with in a connection encrypted by Transport Layer Security(TLS) or Secure socket layer (SSL)
- Motivation of **HTTPS** is authentication of website and to protect the privacy of exchanged data
- **HTTPS** encrypts the message before the transmission and decrypts a message on arrival
- **HTTPS URL example**: https://www.acadgild.com

Full Web Request Cycle

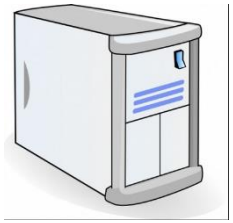
1. The client opens the URL http://acadgild.com/tutorials.php.



2. The request is sent to name servers to convert the domain name to an IP address.

DNS Server

IP **Address**

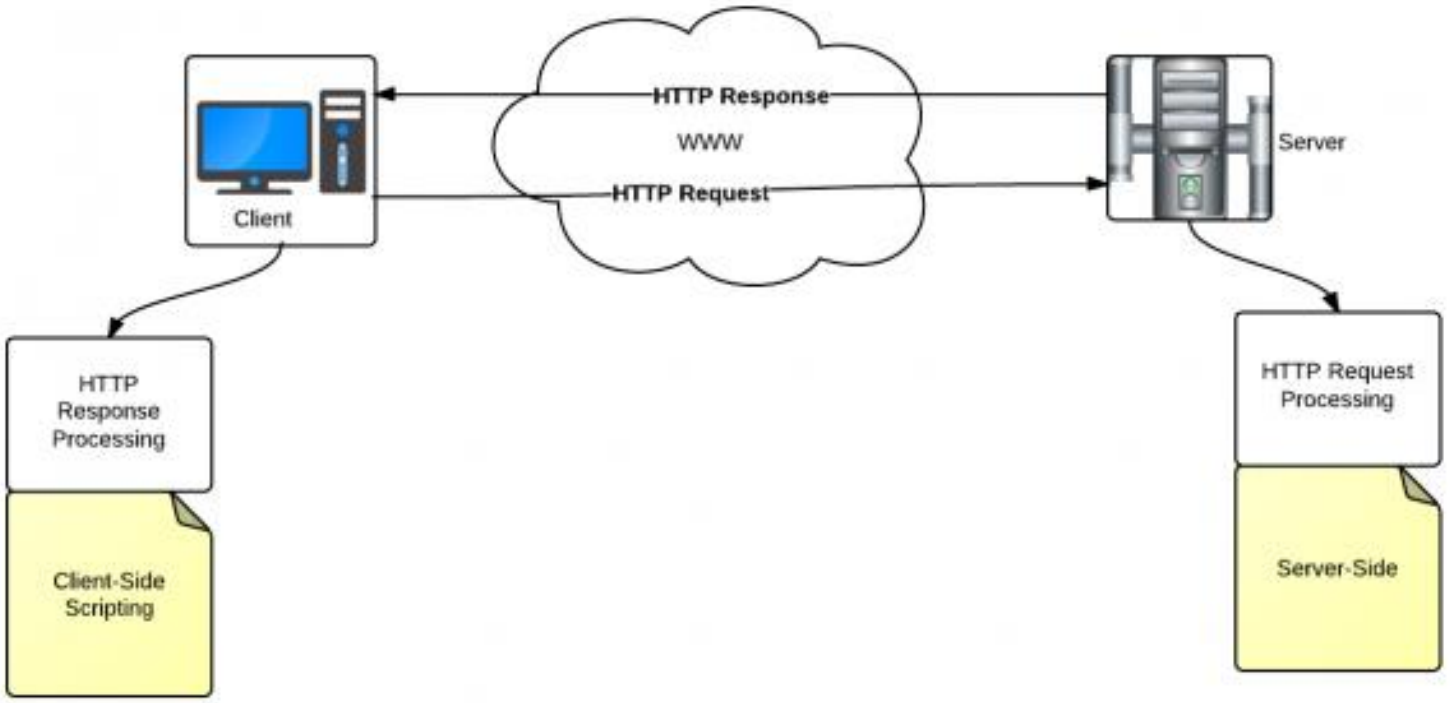


3. The server checks for the file type and finds it to be a php file type and sends the request to **php interpreter.**

4. Based on the request, interpreter connects to either database, file system, mail server or external network.

5. After fetching data, php interpreter accumulates the output into a HTML, CSS, and JavaScript code which is sent to browser.

Request Response Cycle



Server Side Vs Client Side Languages

- **Server side languages** have code that run on the web server.
- All sensitive computations like user login are done by server side languages. Example - PHP, Ruby, Python.
- **Client side languages** execute at the client browser.
- JavaScript, HTML, CSS are examples of client side languages.
- As a rule, sensitive business logic is not kept at the client side because users can read the source code and modify client side code.
- With Node JS, the above boundaries are getting blurred.

Languages Vs Framework

Languages	Framework
• Different languages have different syntax and run time environments. • Examples - JavaScript, PHP and HTML are all languages.	• Framework is not a language. • It is collection of classes, functions and common business logic to speed up the development. • Examples - AngularJs, NodeJs, BackboneJs. All Frameworks are designed using JavaScript language.

HTML

- **HTML** stands for **Hyper Text Markup Language**, which is the most widely used language on Web to develop web pages.
- **HTML** is interpreted by browsers and hence we don't need to compile it.
- **HTML** documents are described by HTML tags.
- **HTML** describes the structure of the web page along with cues of presentation.

Tags, Attributes & Elements

Data Types	Description	Syntax
Tags	• **Tags** are used to mark up the start and end of an HTML element. • **Tags a**re predefined keywords inside angular brackets.	<tagName> …… </tagName> **or** <tagName…/>
Attributes	• **Attributes** are extra bits of information. • **Attributes** appear inside the opening tag and their value sit inside opening tag.	tagName attribute="value">Margarine</tagName>
HTML Elements	• **Elements** are the bits that make up web pages. • **Elements** are written with **start** tag, **end** tag and the **content** in between start tag and end tag.	<tagName>content</tagName>

Basic HTML Syntax

- **Header:** Provides information about the page. e.g. title.
- **Body:** Provides actual content of the page.
- **Document** starts with <HTML> and end with </HTML>

Syntax:

```
<html>
<head>
<title>….</title>
</head>
<body> … </body>
</html>
```

Basic HTML Tags

Tag	Description
<h**n**>…</h**n**>	To delimit a level in heading
…	To set the text between the tags to boldface
…	To set the text between the tags to italics
…	To align the text between the tags in center of the page horizontally
…	To start a paragraph
 	To insert a line break
<hr>	To create a horizontal line in HTML page
<img src="…">	To display image
…	To define a hyperlink

Doctype Tag

- It declares which version of HTML is being followed by the document.
- **In HTML version 4.0, there are three types of DOCTYPES:**
  - Strict
  - Transitional
  - Frameset

Note :

- <!Doctype> **should be the first tag in HTML code.**
- **doctype is not a HTML tag**; it is just used to tell the browser about the version of the HTML to follow for the current code.

DOCTYPES	Description
Strict	• Does NOT INCLUDE presentational or deprecated elements (like font). • Framesets are not allowed.
Transitional	• INCLUDEs presentational and deprecated elements (like font). • Framesets are not allowed.
Frameset	• INCLUDE presentational and deprecated elements (like font). • Framesets are allowed.

Hello World Using HTML

```
<html>  // opening tag for HTML
        <head> //opening tag for head
                <title>   //opening tag for Title
                    Hello World
                </title>   //closing tag for head
        </head> //closing tag for head
        <body>  //opening tag for Body
              Hello World
        </body> //closing tag for Body
</html>    // Closing tag for HTML
```

Agenda – HTML

Sl No	Agenda Title
1	HTML Forms
2	Form Element – input
3	Form Element – textarea
4	Form Element – select
5	HTML Headings
6	HTML Formatting Elements
7	HTML Lists
8	HTML Table
9	Example to Create HTML Table
10	HTML Images
11	HTML iframes
12	Meta Tag

HTML Forms

- **HTML Forms** are required when you want to collect some data from the site visitor.
- The HTML **<form>** tag is used to create an HTML form and it has following syntax:

Syntax:

<form action="Script URL" method="GET|POST"> (form elements like input, textarea etc.)

</form>

Form Element - input

- The **<input>** element is the most important form element.
- The **<input>** element may vary depending on the type attribute.

Input Tag	Description
<input type="text">	Defines normal text input. It defines a one-line input field for **text input**. This can also have a value attribute, which sets the initial text in the textbox.
<input type="password">	Is similar to the textbox but the characters are masked. (Text will be shown as asterisks or circles)
<input type="checkbox">	Defines a checkbox, which can be toggled on and off by the user. This can also have a checked attribute (<input type="checkbox" checked> - the attribute doesn't require a value and makes the initial state of the check box to be switched on).

Form Element – input (contd.)

Input Tag	Description
<input type="radio">	Defines a radio button. It lets a user select ONLY ONE value from a list of values. This can also have a checked attribute.
<input type="submit">	Defines a button for submitting form input to a form-handler. It is a button that when selected will submit the form. You can control the text that appears on the submit button with the value attribute. For example: <input type="submit" value="Update"> (Text on the button – Update)

Form Elements - textarea

- The **<textarea>** element is basically a large, multi-line textbox.
- The size of text area can be specified by cols and rows attributes or through CSS's height and width properties.

Syntax:

```
<textarea rows="5" cols="20">(A big load of text)
</textarea>
```

Form Element - select

- The **<select>** element works with the **<option>** tag to make drop-down list.
- The list will normally show the first item to be selected

Syntax:

```
<select name = “name”>
<option value=“first option”>Option 1</option>
<option value=“second option”>Option 2</option>
<option value="third option">Option 3</option>
</select>
```

HTML Headings

- **HTML** heading are defined with <h1> to <h6> tags
- **<h1>** defines as the most important heading and **<h6>** defines the least important heading
- Search engines use your headings to index the structure and content of your web pages:
  - <h1>This is a H1 heading</h1>
  - <h2>This is a H2 heading</h2>
  - <h3>This is a H3 heading</h3>
  - <h4>This is a H4 heading</h4>
  - <h5>This is a H5 heading</h5>
  - <h6>This is a H6 heading</h6>

HTML Formatting Elements

- **HTML** defines special elements for defining texts with a special meaning
- **HTML** uses elements like <b> , <I> for formatting output
- Formatting elements were designed to display special types of texts :
  - Bold text
  - Important text
  - Italic text
  - Emphasized text
  - Marked text
  - Small text
  - Deleted text
  - Inserted text
  - Subscripts
  - Superscripts

HTML Lists

- **Lists** are used to organize items in the browser window.
- **HTML supports two types of list:**
  - **Unordered list**: Bulleted list. Is the most popular type of list. It lists items in no particular order. e.g. bullets
  - **Ordered list**: Numbered list. This type of list is used when the items need to be listed in a particular order. e.g. 1,2,3

Syntax:

```
<UL>
<LI>Item 1</LI>
<LI>Item 2 <LI>
</UL>
```

HTML Table

- The **HTML Table** allows web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.
- The **HTML Table** is created using:
  - **Table tags:** <TABLE> </TABLE>
  - **Row tags:** <TR> </TR>
  - **Cell tags:** <TD> </TD>

Colspan and Rowspan attributes:

- **colspan** attribute is used when you want to merge two or more columns into a single column
- **rowspan** attribute is used when you want to merge two or more rows into a single row.

How to Create HTML Table

```
<TABLE>
    <TR>
      <TD>Data 1</TD>
      <TD>Data 2</TD>
    </TR>

    <TR>
      <TD>Data 3</TD>
      <TD>Data 4</TD>
    </TR>

    <TR>
      <TD>Data 5</TD>
      <TD>Data 6</TD>
    </TR>
</TABLE>
```

Output once the code is executed

Data 1	Data 2
Data 3	Data 4
Data 5	Data 6

HTML Images

- In HTML, **images** are defined with the **<img>** tag.
- The **<img>** tag is empty. It contains attributes only and does not have a closing tag.

  Syntax:

  ```
  <img src="url" alt="some_text">
  ```

HTML iframe

- An iframe is used to display a web page within a web page.

  <iframe src="URL"></iframe>

Example:

```
<iframe src=“demo_iframe.htm” name=“iframe_a”><iframe>
<p>
<a href=“http://www.w3schools.com”> target=“iframe_a”>
 W3Schools.com
</a>
</p>
```

Meta Tag

- HTML lets you specify **metadata**.
- **Metadata** is information about data. Additional important data about a document can be specified using metadata in several ways.
- The **<meta>** tag is kept inside the **<head>** element.
- The **<meta>** tag provides metadata about the HTML document.
- Metadata is not displayed on the web page.
- Meta elements are used to specify page description, keywords, author of the document, last modified and other metadata.
- It is used to provide information about data to browsers, web services and search Engines.
- Metadata can be used by browsers, search engines or other web services.

Lets Discuss Assignments

Assignment