UI FULL STACK WEB DEVLOPMENT

A to Z Modules

UI FULL STACK WEB DEVLOPMENT

What is Full Stack?

- Full Stack developers are now trending in software industry.

- Software developer who can handle end-to-end application development.

- Developer can manage database, server-side application, middleware, client-side tools, web server.

[Full Stack Developer]

- Front-end to Back-end.

What we are going learn in Full Stack?

- Complete frontend development. [HTML, CSS, Bootstrap, JS, JQ, Angular JS]

- Server-Side Technologies. [Node JS]

- Database. [MongoDB, MySql]

- Middleware. [Express.js]

- Web Servers. [IIS, Tomcat]

- Cross platform framework for Mobile apps. [Ionic],

- Angular JS

- React.js

What we can build by using Full Stack?

- Web Application

- Distributed Application

- Mobile Native Application

FAQ: What is difference between Full Stack, MEAN, MERN and UI?

- UI is all about "Front End Development"

- HTML
- CSS
- Bootstrap
- JavaScript
- jQuery
- Angular JS etc.
- No database, middleware and server side

- **MEAN Stack**

M	MongoDB	Database
E	Express	Middleware
A	Angular	Client Side
N	Node.js	Server Side

- No Front end tools or language

- **MERN Stack**

M	MongoDB	Database
E	Express	Middleware
R	React	Client Side
N	Node.js	Server Side

- No Front end tools or language

What is Single Page Application [SPA] and Progressive Web Application?

SPA
- In SPA user will stay on one page and get access to everything from the page.
- User will not Navigate from one page to another.
- In Web Applications, Distributed Applications and Mobile Native applications we are using SPA environment.
- We will do that by using "Angular JS, React.js and Ionic" for SPA

PWA [Progressive Web Applications]
- In PWA web site will have app like behaviors in browser.
- A web site looks like an "app" not like a page.
- We can do that by using "React.js and Ionic".

What is UI, UX?
- Front End development

- ➢ UI [User Interface]
- ➢ UX [User Experience]
- ➢ UX is part of UI.
- ➢ Web started in early 1990's
- ➢ **Tim Berners Lee** introduced the concept of Web.
- ➢ **For Early web development UI is enough.**
  HTML, CSS, JavaScript, jQuery etc.

Modern web development has several challenges

- ➢ 80% to 90% of internet users are using web from
  - smart devices.
- ➢ Our traditional websites are too large and can't
  - reach low bandwidth devices easily.
- ➢ Client wants an
  - **Unified UX**
    - Our application must provide same features and functionality across any device.

  Fluid UX
    - User will stay on one page and get access to everything.
    - Easy Extensibility
    - Simplified Deployment.

  Solution
    - Angular JS, React.js etc.

Front-end Development

What is front-end?

- Software applications follow tier and layer architecture.
- Applications are built in layers.
- Applications run in tiers.

Tier : run application
Layer : build application

- Developers work in 3 layers

- UI Layer
- Business Layer
- Database Layer

- Application will run in tiers

- Database Tier
- Application Tier
- Presentation Tier

- Front End development is all about designing the **UI**.

- In full stack we will learn building UI for 3 types of applications.

- Web Application
- Distributed Application
- Mobile Native Application

What we need to learn for developing UI?

- HTML
- CSS
- LESS
- SASS
- Bootstrap
- JavaScript
- jQuery
- Angular JS
- React
- Ionic

Why Description

HTML: It is a presentation language used for preparing presentation for web, distributed and mobile.

CSS It makes the presentation more attractive, interactive and responsive.

LESS It is a CSS pre-processor. Make your CSS better, simplified, reusable, and compact.

SASS It is also a CSS pre-processor. Make your CSS better, simplified, reusable, and compact.

JavaScript To reduce burden on server.

It manages several interactions client-side.

Security Issues, Blocked by Browsers

JavaScript is language used client-side, server-side, database.

jQuery It is a JavaScript library.
Write less and do more.
It contains set of pre-defined functions.
Functions are build with JavaScript.

Angular JS It is a Front-end Framework.
It can build and control the application flow.
Google.

React It is a JavaScript library.
It is used with front-end framework like Cordova, Ionic, NativeScript.
Facebook.
More work is on UI

Ionic Framework for building cross platform mobile application with React or Angular.

Web Application

- What is a network?

- It is a group of computers connecting with each other for sharing of information and resources.
- ARPANET [Advanced Research Projects Agency Network] is the first computer network introduced in 1960's by US-DOD.

- What are types of networks?

- **LAN**
- **MAN**
- **WAN**

- What is internet?

- It is a Wide Area Network connecting computers across the globe.
- International Network.

- What is Web?

- 1990's "Tim Berners Lee" introduced "Web".
- Web is a portion of Internet with restricted access.

- **Who maintains the standards of Web?**

- W3C
- WHATWG

- **Web Terminology**

- Server
- Website
- WebPage

Web Server

Server refers to a computer or software solution where we can host, process and handle requests.
- Web Server resembles both hardware and software.
- It satisfies the client request by sending and receiving data.
- Web Server is responsible for hosting your applications, process and handling requests in Web.
- Web Server is also known as Http Server.

- **The popular web server software**
  - Microsoft IIS [Internet Information Services Manager]
  - Apache Tomcat
  - Node.js
  - NGINX
  - Lighttpd etc.

- We are going to use the webserver locally our machine, web build application, test, and then we can deploy on live servers.

Windows Web Server

- Windows OS comes with a webserver called "Internet Information Services"
- Locate the Web Server on your windows PC.
  - Open windows control panel
  - Switch to "Large Icon" view
  - Go to "Administrative Tools"
  - Look for "Internet Information Services Manager"
  - Add web server to your windows PC if not available
  - Go to Control Panel
  - Open "Programs and Features"
  - Click on "Turn Windows Features ON or OFF"
  - Select "Internet Information Services"
  - Click OK
- Test Web Server
  - Open any browser
  - Request the following URL

http://localhost or http://127.0.0.1

Web Site

- Web is virtual directory on Web Server.
- It provides access to the resources.
- Web [Portion of Internet]
- Site [Location – usually in computing system location is referring to drive and directory]
- Create a new Website on your local server
  - Open IIS [run – inetmgr]
  - Expand local computer [http://localhost]
  - Expand "Sites" folder
  - Right click on "Default Web Site"
  - Select the option "Add Virtual Directory"

  Alias: Web Site Name [http://localhost/amazon]
  Physical Path: Resource Location [C:\amazon]

Note: You have to use "\" backslash for physical path and "/" forward slash for virtual path. For URL always use forward slash.

Web Page

- Web page provides a user interface (UI) from where user can interact with the resources in our application.
- Web page is a Hyper Text document that provides interface for interacting with resources.
- The term "Hyper" is derived from a Greek term, which means "beyond".
- Hyper Text document is a document that contains information beyond what is displaying.
- **The web pages are classified into 2 types**
  - Static Page
  - Dynamic Page

Static Page

- Static refers to continuous memory.
- The memory allocated for first request will continue for others.
- Static page contains information that will be same across any number of requests.
- Static page responds with the same content across any number of requests.
- **Static Pages will have extension**
  - .htm
  - .html

Dynamic Page

- Dynamic refers to discreet memory.
- The memory is newly allocated for every request.
- Dynamic page contains information that is customized according to the client request.
- Dynamic pages have extension
  - .aspx, .asp, .php, .jsp

Note: Every website that you designed starts with a default page called "index.html".

Elements in HTML

We need a markup language to design a web page. HTML

What is HTML?

Markup Language
Designing Static & Dynamic
Tim Berners Lee
WHATWG
HTML 4
HTML 5

How HTML Works?

Markup => Bytes => Chars => Tokens [Tags] => Node [Element] => DOM => Layout => Rendering => Painting

<p> => 0011011 => < p > => <p> => p [paragraph] => body->p => Layout [DOM] => generating output => painting output

- HTML is a collection of Elements arranged in a hierarchy called DOM.
[Document Object Model]

- HTML presents static DOM.

- JavaScript, jQuery, React, Angular, Vue etc. convert the static DOM into dynamic DOM.

- Tags are used to build element.

- Tags are Tokens

- Tag => Element [Tokenization]

Types of Elements in HTML

1. Normal Elements
2. Void Elements
3. RC Data Elements
4. Raw Text Elements
5. Foreign Elements

Normal Elements:

- They return a presentation directly on callback without using additional attributes
<b>
- It can't end presentation.
- You have to end presentation explicitly by using end tag. </b>
Start Tag <b>
End Tag </b>

Void Elements:

- Void means no-return type.
- These elements will not return any presentation directly on callback.

- They require additional attributes.
- They can return specified content and end implicitly.
- They dosen't require end tag.

Ex. <img>

RC Data Elements:

- These are the container elements used to present plain text content.
- Rich Content is not allowed.

<textarea> </textarea>

Raw Text Elements:

- These are the elements presented by using raw text instead of tokens. [tag]
- They don't have tags or tokens.

&#8337;
&copy;

- They are presented by using "&".

Foreign Elements:

- These elements are not native to HTML, but can be used in HTML.
- They require additional library. Without library you can't use in a page.

SVG, Canvas, MathML

FAQ: What is difference between <img> and <img/>?

Ans: There is no difference HTML or Browser can't understand <img />.

It is just used by developers to supress warnings in editor.

<img /> indicates that it is a self ending element.

HTML Page Extenstion

- HTML page can be defined with 2 extensions

a) .htm b) .html

FAQ: What is difference between ".htm" & ".html"?

Ans: No difference, both are same.

".htm" is the extension given to file as per standards of Operating System.

Structure of HTML Page

- Every HTML page comprises of 2 sections at high level

1. Document Declaration
2. Document Scope

Document Declaration:

- It specifies the version of HTML used for Web Page.
- If document declaration is not defined then it is in HTML 4.
- To indicate that the page is designed in HTML 5, we need document declaration in the first line of page.

```
<!DOCTYPE html>

<!              comment - not a tag
```

- Document declaration contains meta data, information about HTML document like

a) Its version

b) Its culture

c) License etc.

FAQ: What is the purpose of <!DOCTYPE html>?

Ans: It describes HTML document meta data.
It comprises of version.

Document Scope:

- It specifies the scope of HTML document in page.
- It is defined by using <html> tag.

Syntax:

```
<!DOCTYPE html>
<html>
</html>
```

FAQ: Can a page display multiple document?

Ans: Yes

- It is mandatory to define the culture type used in document. So that browser engine can understand the format of content in page.
- The language culture in page is defined by using "lang" attribute.

```
<html lang="en-in">

</html>
```

lang - is an attribute of <html> tag

Sections in Document Scope

- Every document scope in HTML comprises of 2 sections at high level
  a) Head <head> </head>
  b) Body <body> </body>

Head Section:

- It comprises of content that is intended to load into browser memory. So that it can accessed and used by browser or by page whenever required.
- It is defined by using

```
<head>
</head>
```

- Typically head section of web page contains following elements
  1. <title>
  2. <link>
  3. <meta>
  4. <script>
  5. <style>

Title : It defines the title to display for page in the title bar of browser window. It is also used for bookmarking the page.

Syntax:

```
<!DOCTYPE html>
<html lang="en-in">
<head>
 <title> Amazon Shopping | India </title>
</head>
</html>
```

Link: It is used to link any external document to web page, usually we link favicon or stylesheet.

Ex: Link Favicon
1. Every favicon must have extension ".ico"
2. Make sure that favicon file size is between (16 x 16) to (32 x 32) pixels.
3. Goto project and add a new file in public folder by name "favicon.ico"
4. Open favicon.ico with any photo editing tool [Paint]
5. Design your icon and save.
6. Go to "index.html" and link

```
<head>
    <link rel="shortcut icon"  href="favicon.ico">
```

```
</head>
```

rel : It specifies the relation type of external file.
[stylesheet or shortcut icon]

href : It sepcifies the path and name of icon file.

<u>Meta</u>:
- Meta refers to "Meta Data".
- Meta Data means information about your page given to Web Spiders and Web Crawlers used in SEO. [Search Engine Optimization]
- Meta is one of the options used for SEO.
- Meta is also used for responsive design.

Syntax: **Responsive Meta Element**

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
```

Meta in Head Section:

- Responsive Viewport

```
<meta name="viewport" content="width=device-width, initial-sacle=1">
```

- Request Handling

```
<meta http-equiv="refresh" content="4">
```

This reloads entire page at regular time interval.

- Character Set

utf-8, utf-16, utf-32
Unicode Transformation Format 8, 16, 32 bit size

English - 8 Bit
Chinese, Jap, French, Korean - 16 bit
Arabic, Regional Lang - 32 bit

```
<meta charset="utf-8">
```

- Keywords

They are used to find your page on Web.

```
<meta name="keywords" content="your keywords, words, words">
```

- Description

The summary of your website to display in search results.

```
<meta name="description" content="something about your site">
```

- Author

It specifies the author name if it is a blog.

```
<meta name="author" content="AuthorName">
```

Note: Web comprises of various types of content provides

a) Web Site b) Web Application c) Blog d) Vlog
e) MicroBlog f) Wiki g) Podcasting h) Widget

Web Site: It is a virtual directory. It comprises only static resources.

Web Application: It comprises of both static and dynamic resources.

Blog : The term blog is derived from "Web-Log" [blog]
It is published and maintained by individual users and updated periodically.

Vlog : Video Logs [YouTube Channel]

MicroBlog : Allows multiple users to post they personal information on to single page.
Ex: Twitter

Wiki : The term wiki means "Quick" [Hawain]
A wiki allows any user to edit its content.
EX: Wikipedia, imdb etc..

Podcasting: Allows to broadcast media content on to network devices.
Ex: YouTube, Media Player, iTunes etc..

Widgets : Gadgets for web application or site.
Ex: Currency Converter, EMI Calculator etc..

Summary: Head Section
- title
- link
- meta
- **script :** It is used to embed client or server side script.
- **style :** It is used to embed styles in web page.

Ex:

```
<!DOCTYPE html>
<html lang="en-in">
  <head>
     <title>Amazon Shopping | India</title>
     <link rel="shortcut icon"  href="favicon.ico">
     <meta name="viewport" content="width=device-width, initial-scale=1">
     <meta http-equiv="refresh" content="5">
     <meta charset="utf-8">
     <meta name="keywords" content="your keywords">
     <meta name="description" content="something about your site">
  </head>
</html>
```

Body Section

- Body section comprises of content to render directly into browser workspace.
- Body section element comprises following attributes.

Body Section Attributes

Attribute	**Description**
bgcolor	sets background color.
text	sets the text color.
background	sets background image.
align	aligns content left, center, right or justify.
leftmargin	]
rightmargin	] set space between content and page
topmargin	]
bottommargin	]
alink	sets color for active link.
vlink	sets color for visited link.

<body attribute=value> // valid **- not recommended**
<body attribute="value"> // valid - recommended
<body attribute='value'> // valid - recommended
<body attribute=`value`> // valid - **recommended with scripting language.**

"outer 'inner string' string"
(or)
'outer "inner string" string'

<u>HTML Colors:</u>

- In HTML colors can be defined in 2 ways
a) Color Name b) Hexa Decimal Code

- Color Name define the generic name of color
bgcolor="red"
bgcolor="lightred"
bgcolor="lime"

- Color Hexa Codes
a) 3 char code with "#" b) 6 char code with "#"

- 3 Char Hexa Code

#RGB	[Red, Green, Blue]
R	: 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
G	: 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
B	: 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
0	: darkest shade
f	: bright shade
#000	: black
#fff	: white

- 6 Char Hexa Code
#RRGGBB
#ff0000 - red

```
        #ff00ff                 - pink

        <body bgcolor="red">          3 chars
        <body bgcolor="#f00">         4 chars
```

Syntax:

```
        <body bgcolor="#f00" text="#fff"> </body>
```

Background Image:

- Every image that you want to use for project must be imported into project.
- Don't ever try to access image directly from your PC locations.

```
        <body  background="c:\mypictures\photo.jpg">  // Invalid for server
```

- Always you have to define virtual path.

```
        <body  background="./assets/photo.jpg">
```

Note: "assets" is a folder created in "public" folder to keep all static resources like images, text document, pdf, audio, video etc.

Margins for Page:

- You have to apply margins according to the device you are targetting.
- The units for margin can be
    a) Pixels
    b) Percentage
    <body leftmargin="20%">
    <body leftmargin="200">
- Units in % will give fluid experience [UX]

Link Colors:
=========
- Page comprises links.
- Links are used in navigation.
- Link colors can be changed by using
    a) alink
    b) vlink

Syntax:

```
        <body alink="gray" vlink="red">
```

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Index Page</title>
    </head>
    <body vlink="gray" alink="red">
      <a href="home.html">Home</a>
      |
      <a href="http://www.microsoft.com">Microsoft</a&gt;
      |
      <a href="http://www.amazon.in">Amazon</a&gt;
    </body>
</html>
```

Example with Styles – Netflix page

Styles in Web Page

1. Styles are defined by using <style> element in head section.

```
        <head>
                <style>
                </style>
        </head>
```

2. Styles require a selector to apply effects

```
    selector
  {
        your effects;
  }
```

3. Currently the selector we are using is "TypeSelector", which means we are using tag name.

```
     header { }
     aside   {  }
```

4. CSS have lot of attributes

Attributes of Colors and Effects:

background-color	: set background color
background-image	: set background image
background-size	: sets background image size
color	: sets color for text
height	: sets height in pixes or %
width	: sets width
border	: sets border for content.
padding	: space around content.
margin	: space between page and content.
font-size	: sets text size in pixels
font-weight	: sets bold
font-family	: Arial, Times New Roman
rgb()	: red green blue colors value will be between 0 to 255; rgb(0,0,255); rgb(R,G,B);
rgba()	: red, green, blue, alpha[0 to 1], rgba(0,0,0,0.5);

Ex-1:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Netflix | India</title>
      <style>
         #bgshade
         {
            background-color: rgba(0,0,0,0.6);
            width: 120%;
            height: 768px;
            margin-top: -20px;
            margin-left: -20px;
         }
      </style>
   </head>
   <body background="./assets/netflixback.png"  >
     <div id="bgshade">       </div>      </body>           </html>
```

Arranging contents flexible on Page:

```
============================
dispaly: flex
justify-content: space-between
```

Arranging Contents center on Page:

```
============================
display: flex
justify-content:center;           // horizontal center
align-items:center;                 // vertical center
```

Note: The container must be defined with height if you want vertical center.

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Netflix | India</title>
        <style>
            body {
                background-image: url("./assets/netflixback.png");
            }
            #bgshade
            {
                width: 100%;
                height: 768px;
                background-color: rgba(0,0,0,0.5);
            }
            #brand {
                font-size: 30px;
                color:red;
            }
            header{
                padding:20px;
                display: flex;
                justify-content: space-between;
            }
            #lang {
                width: 150px;
                height: 30px;
                background-color: gray;
                color:white;
                font-size: 20px;
                border:2px solid black;
                padding: 10px;
            }
            #signin
            {
                width: 150px;
                height: 30px;
                background-color: red;
                color:white;
                font-size: 20px;
                border:2px solid black;
                padding: 10px;
            }
```

```
      section {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 500px;
      }
      main {
        text-align: center;
        width: 650px;
        font-weight: bold;
        font-family: Arial;
      }
      #title {
        font-size: 60px;
        color:white;
      }
      #subtitle {
        font-size: 20px;
        color:white;
        margin-top: 20px;
      }
    </style>
  </head>
  <body>
   <div id="bgshade">
    <header>
      <div id="brand">
        NETFLIX
      </div>
      <div>
        <span id="lang">Language</span>
        <span id="signin">Signin</span>
      </div>
     </header>
     <section>
       <main>
        <div id="title">
          Unlimited movies, TV shows and more.
        </div>
        <div id="subtitle">
          Watch anywhere. Cancel anytime.
        </div>
       </main>
    </section>
   </div>
  </body>
</html>
```

Arrange in columns:

===============
display: grid
grid-template-columns: 3fr 3fr 3fr 3fr [12 fr]

Basic Elements

Basic Elements and Entities in Body Section

==================================

1. Line Breaks <br> <br />

2. Blank Space &nbsp; [non-breakble space]

3. Tooltip | Titles "title" attribute is used for tooltip.
Syntax: <span title="some description">Text</span>

4. Preformatted Text <pre> </pre>
It can present content without loosing the from.

5. Address Blocks <address></address>
It makes your contact details SEO friendly.

6. Code Blocks <code> </code>
It makes SEO easy to understand the code block in your page.
A code block can be any computer language code.

7. Variable Blocks <var> </var>
It is used to define variables in programming. Variables are used to store values.

Ex: Preformat

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
            pre {
                background-color: lightgray;
                padding: 20px;
                width: 350px;
            }
        </style>
    </head>
    <body>
       <div>Sample C Program</div>
       <div>
            <pre>
                #include &lt;stdio.h&gt;;
                main()
                {
                    printf("Welcome to HTML");
                }
            </pre>
        </div>
     </body>
</html>
```

Ex: Address

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
```

```
            pre {
                background-color: lightgray;
                padding: 20px;
                width: 350px;
            }
        </style>
    </head>
    <body>
       <address>
           Topitcourses <br>
           hr@topitcourses.in <br>
           www.topitcourses.com <br>
           Ph:9995848372
       </address>
    </body>
</html>
```

Ex: Code Block

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
            code {
               color:red;
            }
        </style>
    </head>
    <body>
        <div>Code Blocks</div>
        <code>404: Page you requested not found</code>
    </body>
</html>
```

Ex: Variables

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
            var {
                color:gray;
            }
        </style>
    </head>
    <body>
        <div>Variables</div>
        var <var>x</var> = 10; <br>
        var <var>y</var> = 20; <br>
        <var>z</var> = <var>x</var> + <var>y</var> ;
    </body>
</html>
```

Headings in HTML

- HTML provides <hn> element for headings.
- n refers to level number of heading from 1 to 6.

Syntax:

```
        <h1> </h1>
        <h2> </h2>
        ................
        <h6> </h6>
```

- It is used to make your topic SEO friendly.
- SEO can make a summary of topics in page.
- Never use headings to highlight any word or sentence in page.
  [headings have a line break before and after]
- Never use too many headings in page. This will SPAM your page.

FAQ: Can we change heading font color, size, style?
Ans: Yes. By using styles.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
        </style>

    </head>
    <body>
        <h1>Web Technologies</h1>
        <h2>HTML</h2>
        It is a markup language.
        <h2>CSS</h2>
        It is used to define styles.
        <br><br>
        <font size="5"><b>JavaScript</b></font>
        <br><br>
        It is a language.
    </body>
</html>
```

Paragraphs and Blockquotes:

<p> for paragraph
<blockquote> for blockquote, which is similar to paragraph but have left and right indentation.

Attribute: **align** : It aligns content left, center, right and justify.

Note: Blockquotes are used to make page content SEO friendly.
SEO can get the summary of content published from blockquote.

Syntax:

```
        <p>  your content </p>
        <blockquote> your content </blockquote>
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <title>Home</title>
        <style>
           p {
               text-align: justify;
           }
           blockquote {
               text-align: justify;
               font-style: italic;
               margin-left: 200px;
               margin-right: 200px;
           }
        </style>

    </head>
    <body>
        <h2>HTML</h2>
        <blockquote>           Your use of this software is subject to the terms and conditions of the license
agreement by which you acquired this software.          </blockquote>
        <p>           Your use of this software is subject to the terms and conditions of the license agreement by
which you acquired this software          </p>
        <p>           Your use of this software is subject to the terms and conditions of the license agreement by
which you acquired this software.          </p>
    </body>
</html>
```

Data Terms and Definitions

<dl>	Data list with terms and definitions.
<dt>	Data Terms
<dd>	Data Definition

Syntax:

```
        <dl>
           <dt> Term-1 </dt>
           <dd> Definition </dd>
           <dt> Term-2 </dt>
           <dd> Definition </dd>
         </dl>
```

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
           dt {
               font-weight: bold;
               background-color: gray;
               color:white;
               margin-bottom: 10px;
           }
        </style>
    </head>
    <body>
        <dl>
          <dt>HTML</dt>
          <dt>Hyper Text Markup Language</dt>
          <dd> It is a markup language. </dd>
          <dd> It is used for presentation.</dd>
          <dd> Introduced by Tim Berners Lee.</dd>
          <dt>CSS </dt>
          <dt>Cascade Style Sheet</dt>
          <dd>It defines styles.</dd>
          <dd>Styles make HTML more interactive.</dd>
        </dl>
    </body>
</html>
```

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
          dl {
              display: grid;
              grid-template-columns: 3fr 9fr;
          }
          dt{
              width: 200px;
              background-color: black;
              color:white;
              margin-bottom: 20px;
              padding: 5px;
          }
          dd {
              background-color: gray;
              color:white;
              margin-bottom: 20px;
              padding:5px;
          }
        </style>

    </head>
    <body>
        <h2>Product Details</h2>
        <dl>
          <dt>Name</dt>
          <dd>Samsung TV</dd>
          <dt>Price</dt>
          <dd>56000.55</dd>
          <dt>Stock</dt>
          <dd>Available</dd>
          <dt>Rating</dt>
          <dd>4.3 [3,000]</dd>
        </dl>
    </body>
</html>
```

Ex:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Home</title>
      <style>
        nav {
           width: 200px;
        }
        dt {
           background-color: black;
           color:white;
           padding: 5px;
           position: sticky;
           top: 0px;
        }
      </style>

    </head>
    <body>
       <nav>
          <dl>
            <dt>Electronics</dt>        <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>          <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>         <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>          <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>          <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>          <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>          <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>          <dd>Mobiles</dd>       <dd>TV</dd>        <dd>Watch</dd>
            <dd>Speakers</dd>          <dd>Mobiles</dd>      <dd>TV</dd>        <dd>Watch</dd>
          <dd>Speakers</dd>          <dt>Footwear</dt>      <dd>Sneakers</dd>   <dd>Casuals</dd>
            <dd>Boots</dd>          <dd>Sneakers</dd>      <dd>Casuals</dd>    <dd>Boots</dd>
          <dd>Sneakers</dd>          <dd>Casuals</dd>      <dd>Boots</dd>     <dd>Sneakers</dd>
            <dd>Casuals</dd>         <dd>Boots</dd>       <dd>Sneakers</dd>    <dd>Casuals</dd>
            <dd>Boots</dd>          <dd>Sneakers</dd>      <dd>Casuals</dd>    <dd>Boots</dd>
            <dd>Sneakers</dd>          <dd>Casuals</dd>      <dd>Boots</dd>     <dd>Sneakers</dd>
            <dd>Casuals</dd>         <dd>Boots</dd>       <dd>Sneakers</dd>    <dd>Casuals</dd>
            <dd>Boots</dd>
          </dl>
       </nav>
   </body>
</html>
```

Details and Summary

<details> A container for displaying content, which can be collapsed dynamically.

<summary> It is caption for details defined in container.

Syntax:

```
<details>
    <summary> Caption </summary>
              your content...
</details>
```

Attribute: **open**: It is used to keep details open on page load.

Note: Details have browser dependency issues. [Not supported on IE <11]

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
        <style>
        </style>

    </head>
    <body>
        <nav>
            <details open>
                <summary>Electronics</summary>
                <dl>
                    <dd>Televisions</dd>
                    <dd>Mobiles</dd>
                    <dd>Speakers</dd>
                    <dd>Laptops</dd>
                </dl>
            </details>
            <details>
                <summary>Fashion</summary>
                <dl>
                    <dd>Shirts</dd>
                    <dd>Jeans</dd>
                    <dd>Shoes</dd>
                </dl>
            </details>
        </nav>
    </body>
</html>
```

Fieldset and Legend

===============

<fieldset> It is a container with frame.
<legend> It is a caption for container.

Syntax:

```
 <fieldset>
       <legend> Summary </legend>
       your content....
 </fieldset>
```

CSS Attribute for Shadow:

====================

box-shadow: horizontalUnits, verticalUnits, blurUnits, color;
box-shadow: 3px 4px 2px red;
text-shadow: 3px 4px 2px red;

Ex:

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Home</title>
    <style>
        dl {        display: grid;
                    grid-template-columns: 3fr 9fr;                    }
        dt {
            background-color: gray;
            color:white;
            padding: 5px;
            margin-bottom: 20px;                        }
        dd {
            margin-bottom: 20px;
            padding: 5px;                }
        fieldset {    margin-bottom: 30px; }
        legend {
            background-color: black;
            color:white;
            text-align: center;
            padding: 4px;
            box-shadow: 4px 4px 2px gray;  }
        #title {
            font-size: 80px;
            font-weight: bold;
            text-align: center;
            text-shadow: 4px 4px 2px black;
            color:yellow;                               }
        body {
            padding: 20px;
            box-shadow: 10px 10px 2px gray;
            margin: 30px;
            border:2px solid black;                }
    </style>

</head>
<body>
    <div id="title">
        RESUME
    </div>
    <fieldset>
        <legend>Personal Information</legend>
        <dl>
            <dt>Name</dt>
            <dd>some name</dd>
            <dt>Fathers Name</dt>
            <dd>some name</dd>
            <dt>Date of Birth</dt>
            <dd>some value</dd>
        </dl>
    </fieldset>
    <fieldset>
        <legend>Contact Information</legend>
        <dl>
            <dt>City</dt>
            <dd>city name</dd>
            <dt>State</dt>
            <dd>state name</dd>
```

```
                <dt>Email</dt>
                <dd>your email</dd>
            </dl>
        </fieldset>
    </body>
</html>
```

Ordered List and Unordered List

Ordered List:

- It defines auto numbering for items in a list.
- It is defined by using

<ol>	Ordered List
<li>	List Item

```
Syntax:     <ol>
                <li> Item-1 </li>
                <li> Item-2 </li>
            </ol>
```

- Numbering update automatically when you add or remove items.

Attributes:

type : It defines the numbering type, which can be
1, A, a, i, I

```
<ol  type="a">
```

start : It defines the level number to start with.
[only number]

```
<ol  type="a"  start="5">
```

reversed : It is used to reverse numbering style.

```
<ol  type="1"  start="2" reversed>
```

- **Numbering be nested**

```
<ol type="1">
   <li> Item-1
        <ol type="a">
            <li> Child-1 </li>
        </ol>
          </li>
</ol>
```

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>List Demo</title>
    </head>
    <body>
        <ol>
            <li>
                HTML
                <ol type="a">
                    <li>Normal Elements
                        <ol type="i">
```

```
                            <li>Bold</li>
                            <li>Italic</li>
                        </ol>
                    </li>
                    <li>Void Elements</li>
                </ol>
            </li>

            <li>
                JavaScript
                <ol type="a">
                    <li>Variables</li>
                    <li>Data Types</li>
                </ol>
            </li>
        </ol>
    </body>
</html>
```

FAQ: How to display items side by side?
Ans:

a) display:inline for list items

```
        li {
            display:inline;
         }
```

b) display:flex for ordered list

```
        ol {
            display:flex;
        }
        li {
          margin-left: 30px;
        }
```

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>List Demo</title>
        <style>
            ol {
                display: flex;
            }
            li {
                margin-left: 30px;
            }
        </style>
    </head>
    <body>
        <ol>
            <li>HTML</li>
            <li>CSS</li>
            <li>JavaScript</li>
        </ol>            </body>          </html>
```

FAQ: How to display Items in multiple columns?
Ans: By using
display:grid for ordered list.

Syntax: Columns Side by Side

```
ol {
     display:grid;
     grid-template-columns: 6fr  6fr;
   }
```

Syntax: Columns with items one below another

```
ol {
    columns:2;
  }
```

Ex:

```
<!DOCTYPE html>
<html>
   <head>
      <title>List Demo</title>
      <style>
         .options {
             columns:2;
             margin-top: 20px;
          }
       </style>
   </head>
   <body>
       <ol type="1">
           <li>The ________elements will not have end tag.
              <ol type="a" class="options" >
                  <li>Normal Elements</li>
                 <li>Void Elements</li>
                 <li>RC Data Elements</li>
                 <li>Raw Text Elements</li>
              </ol>
          </li>
          <li>HTML is _______language.
              <ol type="a" class="options">
                 <li>Programming Language</li>
                 <li>OOP Language</li>
                 <li>Presentation Language</li>
                 <li>Functional Language</li>
              </ol>
          </li>

       </ol>
    </body>
</html>
```

Unordered List:

- It defines a bulleted list.
- You can configure list of items with bullet symbol.
- Symbols are defined with "type" attribute for <ul>.

Syntax:

```
<ul  type="disc, circle, square">
<li> Item-1 </li>
<li> Item-2 </li>
</ul>
```

- List can be nested.
- List can be inline or columns etc..

FAQ: How to define custom [user desired] symbol?
Ans: By using
list-style-image : url ("path");

Syntax:

```
 ul {
              list-style-image: url("images/bullet.png");
         }
```

Note:

- This is a **preview UI FullStack eBook** containing **only 30 pages**.
- It is provided to help you understand **how the full UI FullStack eBook looks and is structured**.
- The **complete complete UI FullStack eBook** includes detailed concepts, real-world examples, and career guidance.
- **Purchase the full UI FullStack eBook for just ₹349**.
- Buy now from our official website: https://topitcourses.com/ui-full-stack-web-developer/