Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP

Learn Anytime Anywhere

Andhra Pradesh State Skill Development Corporation

Web designing using ReactJs

Introduction to HTML

Web design vs Web development.

- In essence, web design refers to both the aesthetic portion of the website and it's usability. Web designers use various design programs such as Adobe Photoshop to create the layout and other visual elements of the website.
- Web Developers on the other hand, take a website design and actually make a functioning website from it. Web developers use HTML, CSS, JavaScript, PHP and other programming languages to bring to life the design files.
  Web developers are of 3 kinds:
- Front-end developer
- Back-end developer &
- Full stack developer

1. Front-End Developer (FED):

Front End typically refers to what you actually see on the website in the browser (and is often called "client-side"). This means that front end developers are responsible for everything that you see when you're navigating around the Internet, from fonts and colours to dropdown menus and sliders.

Skills needed: JavaScript, HTML, CSS, jQuery, HTML5, CSS3, Ajax, UI (User interface), UX (user experience), JavaScript Frameworks such as Angular-s, Angular.js, ReactJS, vue.js, Ember.js and Meteor.js

2. Back-End Developer

While the front end is everything the user interacts with directly, "Back End" typically refers to the guts of the application which live on the server (and is often called "server-side"). The back end of a website consists of a server, an application, and a database. Back end developers generally work with front end developers to make their code work within the site's design (or to tweak that design when necessary) and user interface.

Skills needed: JavaScript, HTML, CSS, jQuery, SQL, Java, Linux, Python, Ruby, PHP

3. Full-stack developer

Full Stack web development is a combination of both the front-end and back-end. "Stack," means layer in this case. A full-stack web developer has expertise in all layers of a website's development. This includes, but not limited to: the server, client and hosting, a form of data structuring or modelling, user interface and experience, as well as the needs of the actual business.

Skills needed: JavaScript, HTML, CSS, jQuery, PHP, SQL, Java, HTML5, Ajax, CSS3, Linux, Python, Ruby, UI (User interface)

Chapter - 1

HTML

1.1 Introduction to HTML5:

1.11 What is HTML?

- **HTML:** stands for Hypertext Mark-up Language
- HTML is a mark-up language used to create web pages (either static or dynamic).
- By default, a web browser will display exactly what you type
- HTML is the heart of web pages and HTML5 is the latest version to be approved by the World Wide Web Consortium (W3C)

Before going to creation let us know about normal web pages.

In general, the HTML structure consists of 3 parts.

- HTML version information
- Head &
- Body

HTML version information:

We can declare the HTML version by using **doctype.**

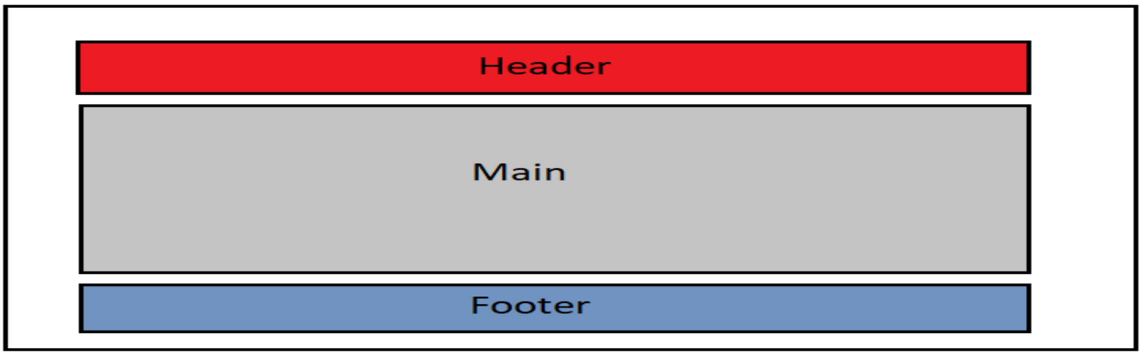
Head:

- In the head of web page contains a title, icon, Meta tags, Styles & Some Scripts related to styles
- The head part content is not visible on the screen. It's works on background of the webpage/website

Body:

A. The body session can be devices into 3 major parts/sessions

1. Header
2. Main
3. Footer



B. In the web browser entire visible text is written the body session

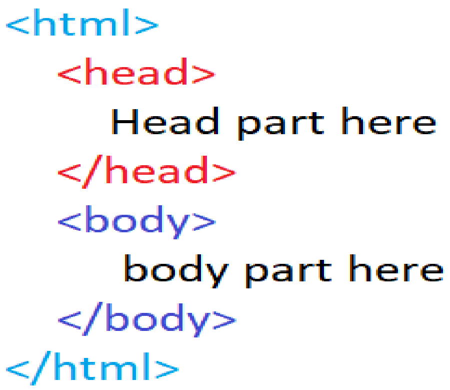
- Before Going to learn web development we need one of the editors like Visual Studio Code, Atom, Sublime Text, etc..,
- Without an editor we can't able to implement our website
- Html language uses a series of elements to display the content in a web page.
- Any text enclosed between less than (<) and greater than signs (>) is an HTML tag **<html>**
- Most of the HTML tags require a start Tag and an end tag, end tag differs with the opening tag by a slash "/"

Syntax:

```
<starting tag> content </ending tag>
```

Example:
<html> </html>

1.1.12 Structure of Html:



```
<html>
    <head>
        Head part here
    </head>
    <body>
        body part here
    </body>
</html>
```

Open Text Editor and create a folder inside our folder create 3 more folders

Example: Web design (Root folder)

Inside the Web design folder create 3 folders with names CSS, images, JS (In the CSS folder we will store all .CSS files, in the image folder will store all static images, in the JS folder will store all JavaScript files).

After creation of these folders it's time to create MainSource file i.e. index.html file.
(index is the filename and .html is the file extension; all html files will save with .html extension)

Why is the index.html file the main source file of our project?

Index is the most common name used for the default pages shown in the website. If we mention the name index.html in our website. The browser is considered as a default page. We can say that the index page is as the root file.
Basic Tags in Html:

1. <!DOCTYPE html>
2. <html lang=" en-US ">

These 2 tags are basic and primary tags

By using these tags even a disabled person can understand the document type is html, language we are using is English and the version is 5.

Head part:
Actions we are able to perform in head part:

- Able to give title to our webpage
- Able to keep an icon as favicon
- Allows us to use external CSS and manifest files.
- Moreover, we can use scripting languages by using script tags.
- Provides the way to use meta elements.

The HTML <head> element is a container for the following elements: <title>, <style>, <meta>, <link>, <script>, and <base>

By using title (<title> </title>) tags we can denote the title of the page that displays on the page tab.

Let us give a title and favicon to our webpage.

To give title the syntax is:

<title> any title </title> (title tag is used to keep a title to our webpage)

To keep an icon as a favicon to our webpage first of all let me introduce you to different types of images.

1. Raster Images
2. SVG Images

Raster Images: Raster Images are images which can be taken by camera or uploaded with mobile. If we Zoom Raster Images at a certain point Raster Images miss their clarity

SVG Images: SVG Images are the images which can be downloaded through google.

If we zoom SVG Images won't miss their clarity.

Download an image in SVG format and store that image in our project at the image folder.

Syntax to keep an image as favicon to our webpage:

<link rel=” icon” type=” favicon” href=” image/myimage.txt” />

Link attribute is used to navigate or link

Here **rel** stands for relation
Type used to define the type
Href stands for head reference path where we have to mention the path.

Body part:
The main content will be displayed in this body tag.

Let us learn about different types of elements in Html5:

1. Block-level Elements:
   The element which occupies the complete width of the screen is known as Block-level Elements.

1. All heading tags (h1 to h6), paragraph tag (p)
2. All semantic elements
3. Form tag

1.All heading tags:

To define heading we have to use h1 to h6 tags.
The font size decreases gradually from h1 to h6.
The syntax goes here:

```
<h1> content </h1>
<h2> content </h2>
<h3> content </h3>
<h4> content </h4>
<h5> content </h5>
<h6> content </h6>
```

All the heading tags occupy the complete width of the screen.

Paragraph tag:

To display the content in the paragraph we will use paragraph tag

Syntax:

```
<p> content </p>
```

This paragraph tag occupies the complete width of the screen.

Before going to semantic tags/elements in the html5 version all tags are divided into 2 types i.e.,

1. Semantic (Header, main, footer, section, article etc.,)
2. Non-Semantic (div, span)

Semantic Elements:

Semantic Elements are similar to div tags used to divide the content into sections.
But div tags don't contain any Description.
Semantic Tags clearly explains its meaning to both the user and browser.

Semantic Elements are:

1. Section
2. Article
3. Aside
4. Nav
5. Header
6. Footer etc.,

Section:

The <section> element defines a section in a document.
Syntax goes here:

```
<section>
    content
</section>
```

Article:

The <article> element is used to define a independent section in a webpage
Example: Blog in a Newspaper

Syntax:

```
<article>
    Content
</article>
```

Header:

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

Syntax:

```
<article>
   <header>
     <h1> content </h1>
     <p> content   </p>
  </header>
</article>
```

Footer:
The <footer> element defines a footer for a document or section.
A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.
Syntax:

```
<footer>
      <p> content </p>
      <p> content 1 </p>
</footer>
```

Nav element:
The <nav> element defines a set of navigation links like home, contact etc.,
Syntax:

```
<nav>
   <a href = “path” >
</nav>
```

Aside:
The <aside> element defines some content aside from the content it is placed in (like a sidebar).
The <aside> content should be indirectly related to the surrounding content.
Syntax:

```
<aside>
   <h1> content </h1>
   <p> content </p>
</aside>
```

Inline Elements:
The Elements which don't occupy the complete width of the screen (we are able to see the output side by side).

1. Span tag
2. Image tag
3. Anchor Tag
4. Button
5. Input etc.,

Span tag:
The <span> tag is an inline container used to mark up a part of a text, or a part of a document.
The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.
Syntax:

```
<span> content </span>
```

Image Tag:
Image Tag is used to display an image in the body part.
Syntax:

```
<img src=" path of the image">
```

Anchor Tag:
The <a> tag defines a hyperlink, which is used to link from one page to another page
Syntax:

```
<a href="{ resource page link }" Name for displaying purpose </a>
```

A linked page is normally displayed in the current browser window, unless you specify another target
Button:
It defines a clickable button. It tell the browser what type of button is
Syntax:

```
<button > Click Here..! </button>
```

Input: <input>
It’s the most important element in the form data. By using this we can display it in several ways depending on the type attribute.

Navigation Elements:

All Navigation Elements are inline Elements where we are able to see the output side by side.
Navigation Elements are used to navigate
All Navigation Elements uses <a,href> (anchor tag)

1. Inbound navigation
2. Outbound navigation
3. Tel
4. Mailto

Inbound navigation:
Inbound navigation is used to navigate to the content present in the same file .
(we need an identifier selector to select that particular tag).
Syntax:

```
<a href="#one">h3 content </a>
<h1> Hai all </h1>
<h2>welcome </h2>
<h3 id="one"> to APSSDC </h3>
```

Outbound navigation:

Outbound navigation is used to navigate to the content present in another using <a,href> (create another file “resume.html” and write some content using h1 tag)

Syntax:

<a href=”resume.html>Content in resume.html </a>

Mailto:

Mailto navigation element is used to send a mail to a particular person (after doing some configuration settings we are able to send the mail within the web page).

Syntax:

<a href=”mailto: xyz@gmail.com”> mail </a>

Tel:

Tel navigation element is used to call a particular person.

Syntax:

<a href=”tel: +91 9123456789”> phone </a>