

Module - 1

Front-end : It is a part of web development that involves creating visual and interactive aspects of a web site or applications. It focuses on what users see and interact with in web browsers.

Languages and Technologies

- 1) HTML : defines the structure and content of the web pages.
- 2) CSS : styles the layout.
- 3) javascript : It adds interactivity and dynamic behaviour to web pages.
- 4) framework : React, angular helps in streamlining and enhancing the web development.

HTML editors : Sublime, atom, VS code, Notepad++
Vim

Difference between HTML and HTML5

HTML	HTML5
→ does not support audio and video without the use of flash layer support.	→ it supports audio & video with the help of <video> and <audio> tags.
→ It uses cookies to store temporary data.	→ It uses SQL database to store offline data.
→ does not allow drag & drop effect	→ It allows drag and drop effect.
→ older version of html are less mobile friendly.	→ More mobile friendly.
→ doctype declaration is too long and complicated	→ Doctype declaration is easy and simple.

$\rightarrow <\text{script type}=\text{"text/javascript"}\text{ src}=\text{"abc.js"}></\text{script}>$	$\rightarrow <\text{script type}=\text{"text/javascript"}\text{ src}=\text{"abc.js"}></\text{script}>$
$\rightarrow <\text{link rel}=\text{"stylesheet"}\text{ href}=\text{"abc.css"}>$	$\rightarrow <\text{link href}=\text{"abc.css"}>$

Basic structure of html

```

<html>
  <head>
    <title> </title>
  </head>
  <body>
    <header> </header>
    <nav> </nav>
    <section> </section>
    <aside> </aside>
    <footer> </footer>
  </body>
</html>

```

Viewport

- It is the user's visible area of the web page. The viewport varies with the device. It is small on the phone than on the computer screen.
- HTML5 has introduced a method to allow web designers to take control over viewport, with help of meta tag.

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

device-width : sets width of the page to fit

device width.

initial-scale = 1.0 : it indicates the zoom level when web page is loaded by the browser.

Semantic elements : these are the elements that conveys the meaning and helps to organise and structure the code.

ex: <header>, <footer>, <article>, <section>, <figure>, <figcaption>, <aside>

Non- semantic elements:

ex: <h1>, <p>,
, <pre>

Webforms 2.0 : these are the extended forms of HTML5.

eg: <input type = {
- datetime-local → month
- time → number
- email → Range.
- url
- week}

program

<html>

<head>

<style>

.abc {

background-color: blue;
color: white;

}
fieldset {

margin-bottom: 20px;

}

```
fieldset > div {
```

```
    margin-bottom: 10px;
```

```
    display: flex;
```

{

```
input {
```

```
    border-radius: 5px;
```

}

```
label, input [type = "text"]
```

{

```
    width: 100%;
```

}

```
</style>
```

```
</head>
```

```
<body>
```

```
    <form>
```

```
        <fieldset>
```

```
            <legend class = "abc">
```

```
                Delivery address
```

```
            </legend>
```

```
        <div>
```

```
            <label> Name </label>
```

```
            <input type = "text" >
```

```
        </div>
```

```
    </fieldset>
```

```
    <fieldset>
```

```
        <legend class = "abc">
```

```
            Billing address
```

```
        </legend>
```

```
    <div>
```

```
        <label> same as delivery address
```

```
        </label>
```

```
        <input type = "checkbox" checked>
```

```
</div>
<div>
    <label> Name </label>
    <input type="text">
</div>
<div>
    <label> Address </label>
    <input type="text">
</div>
<fb>
<fieldset>
</form>
</body>
</html>
```

HTML Canvas

It is an HTML element that provides drawing surface for creating graphics and animations with javascript.

Canvas elements have attribute such as width and height that defines size of drawing area.

Code

```
<!DOCTYPE html>
<html>
<body>
<canvas id="mycanvas" width="300"
height="150" style="border: 2px solid black;">
Your browser doesn't support the HTML5
canvas tag </canvas>
```

```
<script>
var c = document.getElementById ("mycanvas");
var ctx = c.getContext ("2d");
ctx.fillStyle = "blue";
ctx.fillRect (20, 20, 150, 100);
</script>
</body>
</html>
```

CSS3

It is an extended version of CSS2 it supports transform property that helps to create 2D or 3D transformations.

The different transformation functions are

Translate(-20px, -100px)

rotate(-20deg)

scale(2, 3)

scaleX(-1)

scaleY(-1)

Code for translate

<html>

<head>

<style>

.rect {

height: 100px;

width: 100px;

background-color: black;

transform: translate(20px, -100px);

}

</style>

</head>

<body>

<p class="rect">Hello </p>

</body>

</html>

→ code

<html>

<head>

<style>

.div {

width: 500px;

height: 300px;

background-color: yellow;

border: 1px solid black;

}

.rotatediv {

transform: rotate(70deg); }

```
</style>
</head>
<body>
    <div> Normal div </div>
    <div class = "rotatediv" > div is
        rotated </div>
</body>
</html>
```

Note

- It will rotate element 20deg clockwise
- transform: scale(2, 3);
 - It will either increase or decrease the object
- transform: scaleX(-1);
 - it will flip the element horizontally.
- transform: scaleY(-1);
 - it will flip the element vertically.
- transform: skew(10deg, 20deg);
 - it skew the element horizontally and vertically by 10° and 20°

CSS color gradients

It will allow you to display smooth transition between 2 or more specified colors. There are 3 types of gradients:

linear - gradient

radial - gradient

conic - gradient

Code

<html>

<head>

Linear-gradient

To create a linear gradient you must specify atleast 2 colors, by default the color transition from top to bottom.

&

```
background: linear-gradient(red, green);
```

- * To customise the transition of color to right:-

```
background: linear-gradient(to right, red, yellow);
```

Radial-gradient

→ It is defined by its center.

```
→ background: radial-gradient(red, yellow);
```

Conic gradient

It is a gradient with color transition rotated around a center point.

```
→ background: conic-gradient(red, yellow);
```

Code (linear gradient)

<html>

<head>

<style>

```
background: linear
```

&

```
background: linear-gradient(red, black);
```

```
height: 100px;
```

```
width: 100px;
```

&

```
p: hover {  
    transform : scale(-1);  
}  
</style>  
</head>  
<body>  
    <p> Hello </p>  
</body>  
</html>
```

Code

```
<html>  
    <head>  
        <style>  
            a {  
                display : block ;  
                text-decoration: none ;  
                width : 200px ;  
                background - color : grey ;  
                padding : 10px 10px ;  
            }  
            a: hover {  
                background - color : blue ;  
            }  
        </style>  
    </head>  
    <body>  
        <a href= " " > home </a>  
        <a href= " " > Login </a>  
        <a href= " " > Signup </a>  
        <a href= " " > Contact us </a>  
    </body>  
</html>
```

http : http is unidirectional where the client sends the request and server sends the response. Each http establishes a new connection to the server & everytime and after getting the response the connections get terminated by itself.

websocket : It is bidirectional, full duplex protocol used in client server communication these are used in trading websites for displaying the price fluctuation

ex: gaming application, chat application
Web sockets are used if we want any real time data or continuous data that are being transmitted over network. but if you want to fetch old data we go with http protocol

Web storage

Local Storage

It is a storage that allow you to store data persistently (data persist even after the browser or tab is closed)

ex: Shopping cart data, game progress in web games.

Session Storage

It is similar to local storage but stores data only for duration of the session. Data will be lost after tab / browser is closed.

ex: Multipage form data

Cookies : These are small text file that website store on your computer / mobile device when you browse internet.

cookies help to remember the user preference.