```
transition: 2s;  ->when hover slowly zooming
fieldset
legend
details
summary
dl
dt
dd
14:)================================================(:14
OrderdList
<ol type="" start="" reversed>
NestedList

<ol type="I">
            <li>HTML is ________language</li>
            <ol type="a" id="optionsList">
                <li>Programming</li>
                <li>Functional</li>
                <li>presentaion</li>
                <li>structural</li>
            </ol>
        </ol>

List Items In Multiple Columns:-
================================
<style>
      #optionsList{
                display: grid;
                grid-template-columns: 6fr 6fr;
             margin-top: 15px;
            }
            li{
                margin-right: 30px;
            }
</style>
List items in side by side:-
============================
<style>
      #optionsList{
                display: flex;

            }
            li{
                margin-right: 30px;
            }
</style>

Custom list as Bullet:-
-----------------------
ul{
```

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

Library for symbols in your project:-
=====================================
-Fontawesome
-Bootstrap Icons

Download and install Bootstrap icons for your project:
=====================================================
1.open project terminal
      [ctrl+`]
2.type the follwing commend
        >npm install bootstrap-icons --save
3.All library files are copied into "node_modules" folder.
4.go to html file and link "bootstrap-icons" css file.
       <head>
          <link rel="stylesheet"
href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
      </head>
5.Every icon is accessed by using its class name,which you can apply for any
container,<span> <div> <p> <dd> <li> etc..

       <span class="bi bi-bell"></span>

     https://icons.getbootstrap.com
NOTE: to remove numbering or bullet symbols for list items
           ul,ol {
              list-style:none;
           }
EX:
Nav Bar:
========

      <head>
        <title>Icons Demo</title>
    <link rel="stylesheet"
href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
    <style>
        ul{
            list-style: none;
            position: fixed;
            bottom: 0px;
            right: 20px;

        }
        li{
            margin-bottom: 20px;
            font-size: 30px;
            ;background-color: darkcyan;
```

```
            color: white;
            padding: 10px;
            border-radius: 10px;
        }
    </style>

      <div>
            <ul>
                  <li class="bi bi-facebook"></li>
                  <li class="bi bi-linkedin"></li>
                  <li class="bi bi-twitter"></li>
                  <li class="bi bi-youtube"></li>
            </ul>
        </div>

15:)text Effects(:15
====================
font
   face,size,color
font styles
    bokd,italic
font Effects
   underline,strikeout,super script,subscript

<font>  it is used to define face,size,color
     face=Arial,Times New Roman,Script
  EX: <h1><font face="monospace">Welcome to HTML</font></h1>

16:)img(:16:
============

<img src="assets/IMG_20220505_103748-removebg-preview.jpg" alt="myImg: 1"
height="20%" width="20%"
             align="left" vspace="20" hspace="20" crossorigin="anonymous"
decoding="sync" border="2px solid dotted" importance="low">


cards:
======
  <div id="container">
        <div id="card">
            <div id="card-header">
                <img src="assets/nit/Asp.Net.jpg" alt="">
            </div>
            <div id="card-body">
                <h3>GoProgramming  Trainning</h3>
                <p>
                <span class="bi bi-star-fill"></span>
                <span class="bi bi-star-fill"></span>
                <span class="bi bi-star-fill"></span>
```

```
                <span class="bi bi-star-fill"></span>
                <span class="bi bi-star-half"></span>
            </p>
            </div>
            <div id="card-footer">
                <button> <span class="bi bi-search"></span>View Course</button>
            </div>
     </div>
  </div>

17:)Links in HTML(:17
====================
FAQ.How to target any reference within the page with style?
=>By using  ":target" selector

   Ex:-
            .topic{
                border: 2px solid black;
                padding: 20px;

                color: black;
                margin-bottom: 20px;
            }
           .topic:target{
                background-color: black;
                color: white;
            }
FAQ.How to change effects for element on mouse over?
=>By using ":hover" selector
        Ex:
            .topic:hover {
                background-color: yellow;
                color: red;
            }
FAQ:How to remove underline for hyperlink?
=>By using "text-decoration :none" with style
      Ex:
         a{
        text-decoration:none;
        }
FAQ:How to change color for HyperLink?
=>a:link{
       effects for normal link;
  }

  a:visited {
       effects for visited link;
  }
  a:active {
       effects for active link;
```

```
  }
  a {
    //effects for all phases [visited,normal and active]
  }

18:)Links in HTML(:18
=====================
-Intra Document Links
-Inter Document Links
     -Handling navigation from one page to another page.
     -Navigates to any specific file,url,application

FAQ:How to open linked document in a new Tab?
=> <li><a href="http://www.amazon.in" target="_blank">Amazon Shopping</a></li>

FAQ:How to open linked document in a new Window?
=> By using javaScript
            "window.open()"

Ex: <li><a
href="assets/NikeCasuals_files/8-15386898-nike-black-original-imaghzr3kjg7cfee.jpeg
">Nike
         Casuals</a></li>

FAQ:How to open linked document in the same page embeded at specific location?
=>By using "<iframe>"

syntax:
       <iframe name="frameBody"></iframe>
       <a href="assets/shoe.jpg" target="frameBody">shoe</a>

FAQ:How to create a link for email,skype and phone call?
=>Links href uses
              "mailto:"        for email
              "tel:"           for phone call
              "skype:"         for skype
              "javascript:"    for executing a javascript function

   Ex: <a href="mailto:hr@nareshit.in">mail</a>
       <a href="tel:+917873719196">Call : +917873719196</a>
       <a href="skype:nareshit@outlook.com">Skype Us</a>

FAQ:How to create a link embeded video or presentation?
=>By using "<iframe>"

  EX:>
https://www.youtube.com/watch?v=LWynsfHNyiI&list=RDCMUC4o8Fdpv3g_AjgShAeivqpA&start
_radio=1
        &rv=LWynsfHNyiI&t=0
  set to:>
```

```
https://www.youtube.com/embed/LWynsfHNyiI&list=RDCMUC4o8Fdpv3g_AjgShAeivqpA&start_r
adio=1
                                        -----
        &rv=LWynsfHNyiI&t=0

syntax:
 <a href="embededUrl" target="frameBody">Watch Video</a>

view url:       watch?v=code
embed url:      /embed/code

19:)Links in HTML(:19
=====================
Note: iframe can use defult sourse by using "src" attribute
   syntax:
         <iframe src="home.html">
         <iframe src="http://site.com"></iframe>

Center your content on page Horizontally and vertically
=======================================================
1.create a container-parent
2.create a container-child
3.keep your content in child container
       <div class="parent">
         <div class="child">
            your content
        </div>
       </div>
4.set following effects to parent
      .parent {
         display: flex;
         justify-content: center;
         align-items: center;
         height: 400px;
       }

Background Image for body section:
==================================
  <body>
        <div class="container">

        </div>
  </body>
styles:
------
   body{
      background-image: url("assets/netflix.jpg");
      background-size: 100% 100%;
      background-repeat: no-repeat;
  }
```

```
 .container {
      background-color:rgba(0,0,0,0.3);
      height: 600px;
 }

20:)Tables in HTML(:20
======================
Table tags:
     <table>,<thead>,<tbody>,<tfoot>,<tr>,<th>,<td>,<colgroup>,<caption>

 <table frame="box" border="1" rules="all" cellspacing="20px" cellpadding="10px>

 </table>
1.cellspacing and cellpadding:-
  cellspacing     :it sets space between cells in table
  cellpadding     :it sets space around table cell content

<table cellspacing="20" cellpadding="10">

2.background color and image

     bgcolor    :it sets backgound color for table,group,row,cell
     background :it sets background image for table,group,row,cell

  <table bgcolor="red">
  <table bgcolor="red">
  <tr bgcolor="red">
  <td bgcolor="red">
  <th bgcolor="red">

<table background="your image path">

3.Allignments:
   align   :horizontally left,center,right and justified
   valign  :vertically top,center and bottom
 <table align="" valign="">
  <td align="" valign="">

4.Merging of rows and columns:
:- colspan  :it can merge multiple columns into single cell
   rowspan :it can merge multiple rows into single cell

22:)Forms in HTML(:22
=====================
:-HtML provides <form> container that comprises of various elements.

The form Container Attributes:
------------------------------
1.id      :it defiens a unique id for form
```

2.name   :it defines a referenece name for form
3.class   :it specifies a css class name

Syntax:
```
         <form id="frmRegister" name="frmRegister" class="form-container">
```

4.method :it defines the type of action to perform.Every html form comprises of 2 action method

Http Method	Http	Purpose
a.	GET	(fetch data)
b.	POST	(submit data)
c.	PUT	(modify all data)
d.	PATCH	(modify partial data)
e.	DELETE	(remove details)

```
       <form method="GET">
       <form method="POST>
       <form>        --default method GET
```

FAQ:Can we POST|SUBMIT data on GET request?
a:- yes. But not recommended.

FAQ:What is difference between GET and POST?
a:-

GET:
-It submit data to server.
-it appends data into url as query string.
-it can be viewed any user.
-so it is not safe.
-it is easy to hack your data.
-any one can bookmark your data.
-it is stored in browser histry/logs.
-Data is appended into url,hence you can't submit complex data.
-you can't submit binary type data [image,graphics]
-you hava limit for data.
-you can submit max 2048 chars.

POST:
-post submits data to server.
-data is submitted as "form body".
-it is not appended into url.
-it is safe.
-it is hard to hack your data.
-it is not in browser history.
-it can't be bookmarked.
-there is no limit for data,you can submit any amount of data.
-you can submit any complex data.even binary type data.

NOTE:
-always use GET method for fetching data
-always use POST method for submmiting data

FAQ:Where data is present,When it submitted on GET request?
->Query String

FAQ:Where data is present,When it submitted on POST request?
->form body

5.action :it indicates the target page where to submit the data.Usually it refers to server side
  pages.[jsp,php,asp,aspx etc..]
      syntax :
            <form method="POST" action="register.php">

6.novalidate :
   -HTML5 intoduced form validations.
   -form validation verifies the details before submitting.
   -it allows submitting only when all fields are valid.
   -"novalidate" is used to ignore validation.it will by-pass the validation

23:)Form Elements(:23
=======================
1.text box:

   -it allows to view,input and edit text.
   -it can handle a string [group of chars-alphabet,numeric and special chars]

 <input type="text">
Attributes:
-name
-id
-class
  Note:Form will not submit value of any element if it is not defined with a name.
       "name" attribute is mandatory for element in a form.
-value   :it indicates the default value to display in textbox.
-placeholder:it indicates the watermark text to display in textbox.

syntax:
   <input type="text" name="UserName" placeholder="subhalaxmi">
-readonly   :it will not allow to change the value but can submit the value
-disabled   :it will disable the element ,form can't submit the value of any disabled element

-size   :it defines the size of textbox. default size is 20.
-autofocus :it sets focus to textbox automatically on page load.
-minlength :it validates the minimum no of characters that can be entered into text box

-maxlength :it restricts the maximum no of chars that can be entered.
-required   :it used to verify and ensure that textbox is having a value and it is not empty .it is used for
             mandatory fields.
-pattern   :it is used to verify the format of input value.it uses a regular expression.
              [A-Z]{4,10}
              [0-9]{10}

            <input type="text" name="mobile" pattern="[0-9]{10}">
               <input type="text" name="userName" pattern="[A-Z]{4,10}">
-list   :it is used to define a datalist <datalist> for textbox which contains options to display as
              auto complete.

   Ex: <input type="text" size="40" list="topics">
        <datalist id="topics">
            <option>HTML Tuturial</option>
            <option>JavaScript Examples</option>
            <option>CSS Examples</option>
            <option>CSS Projects</option>
            <option>HTML Projects</option>
            <option>CSS Tutorial</option>
        </datalist>

24:)Forms in HTML(:24
=====================
2.Password Box:

-it is similar to text box, but will display text masked with password characters("*").
     <input type="password">

3.Number Input:

-it allows the user to input only numeric value. it restricts the input to number.
       <input type="number">
8.File Input:

-it allows to browse and select a file for upload .it will not upload the file.it is just selecting a file.
     <input type="file">
 To select multiple file,you havae to use the option "multiple".

25:)Forms in HTML(:25
=====================
FAQ:How to restrict specific file type in selection?
-By using "accept" attribute

```
 syntax-  <input type="file" accept="MIMEType/Extention">
          <input type="file" accept="image/jpeg">
          <input type="file" accept=".jpg">
          <input type="file" accept=".jpg,.png" multiple>

9.COlor Input:
--------------
-it allows the user to select a color from color panel.
     <input type="color" name="color" value="#ff0000">

10.Radio Buttons:
-----------------
-Allows to select one or more options.
-Radio button once checked can't be unchecked.
-Radio button is used with mutex mechanism.
  [Mutex-Mutual Exclusion]
    syntax:
           <input type="radio" name="Gender" checked> Male
           <input type="radio" name="Gender"> Female
-"checked" attribute is used to select a radio.
-By default radio button submit "on" as value.
-you have to define a value.
    <input type="radio" name="Gender" value="female"> Female

11.Check Box:
-------------
-Check box allows to check and uncheck any option.
-it can be used to select one or more multiple.
-Even you define same name,every check box is individual.

26:)Forms in HTML(:26
=====================
Modern CheckBox
-download and install "bootstrap"
       >npm install bootstrap --save

-Link Bootstrap CSS file to your web page.
    node_modules/bootstrap/dist/css/bootstrap.css

-Apply the following classes for checkbox
   .form-switch
   .form-check-input
 Ex:   <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
        <div class="form-switch">
        <input class="form-check-input" type="checkbox" checked id="accept"><span>I
Accept</span>
     </div>

Dropdown List:
```

```
--------------
-it allows the user to select any one option from list.
         The <select> element defines dropdown list and
             <option> defines items in dropdown list.
syntax:
  <select>
     <option>item-1</option>
     <option>item-2</option>
  </select>

-Every option comprises of following attributes.
  -value     :it defines the value to submit
  -selected  :it makes the option selected[default selection]
  -disabled  :it will not allow to select any specific option

-Every option comprises of following properties[javascript]
  -value
  -selected
  -disabled
  -text     :it indicates the text to display for item in dropdown
              <option value="">Text </option>
  Ex:
        <select name="category">
                <option value="all">All</option>
                <option value="ele101" selected >Electronics</option>
                <option value="foot102">Footware</option>
                <option value="fashion" disabled >Fashion</option>
            </select>

 -you can group the options in a dropdown by using "<optgroup>"
      <optgroup label="Electronics">
            <option>TV</option>
             <option>Mobile</option>
           </option>
NoTE:
 -option Group is used to classify the items into categories visually not
logically.

ListBox:
--------
-it is similar to dropdown but allows the user to select one or more multiple
options from a group of choices.
 -All elements are same as dropdown
    <select>
<optgroup>
<option>

To transform into listbox you hava to defined the following attributes for <select>
element.
a.multiple
```

b.size

syntax:
<select size="3" multiple>
-if size or multiple attributes are not defined then it is a dropdown.

Meter :

-it is used as grade meter.
-it can display a value range .
-it is read only , it will not allow to change maually,but it can change dynamically.

Attributes: min,max,value,low,high
       above minimum and below high
       below maximum and above low
syntax:
  <meter min="1" max="100" value="100" low="20" high="60">
NOTE:meter can hava contextual colors
   warning   -gold yellow
   success   -green
   danger   -red
if low and high=0   :sucess
if low and high range value is minimum   :danger
if low and high range value is maximum   :maximum

Always the value must be set to max for contextual colors.

<meter min="1" max="100" value="100" low="0" high="0"></meter>   :green
<meter min="1" max="100" value="100" low="20" high="80"></meter> :yellow
<meter min="1" max="100" value="100" low="60" high="80"></meter> :red

27:)Forms in HTML(:27
=====================
Progressbar:
-it is used to display status of any task performed by webpage.
-downloading,uploading,copying etc.

 syntax: <progress></progress>
Attributes: min,max,value

    <progress min="1" max="100" value="70"></progress>

Textarea:
-it is used for multiline text.it is an RC data type.
  [only plain text]
    <textarea></textarea>
attributes:
-rows

-cols
-readonly
-disabled
EX:
```
   <textarea name="" id="" cols="30" rows="10" disabled>
               Your texts...
           </textarea>
```

Buttons:
-Used to confirm user actions.
-HTML provides 2 types of buttons
   a.Generic
   b.Non-Generic
-Generic buttons hava pre-defined functionality.
  HTML4
```
 <input type="submit">
 <input type="reset">
```

 HTML5
```
 <button type="submit"></button>
 <button type="reset"></button>
```

Note:HTML 5 Button allows symbols and images.
     HTML 4 Button uses plain text and special chars.
     In html5 if button type is not defined by default it acts as submit.

```
 <button>Register</button>    Submit
```
     If generic buttons are not placed in <form> container,they will not hava generic functionality.

-Non-Generic buttons are static and not defined with any functionality.
```
         <input type="button">       HTML 4
         <button type="button">      HTML 5
```

FAQ:Can we configure additional fuctionality for submit and reset button?
A:-Yes. <form> element uses
  a.onsubmit
  b.onreset

 syntax:
```
   <form onsubmit="function()"  onreset="function()">
  </form>
```
EX:
```
    <form onsubmit="alert('Form submitted...')" onreset="alert('form will reset')">

    <button type="submit">Submit</button>
       <button type="reset">Reset</button>
```

FAQ:Can we define multiple submit buttons in a form?
A:Yes.

FAQ:Why a form need multiple submit buttons?
A:To handle various functionalities

FAQ:How to identify specific submit click?
A:By using name and value attributes.
 EX:

```
    <form>
           <fieldset>
               <legend>Database operations</legend>
               <button type="submit" name="action" value="Insert">Insert</button>
               <button type="submit" name="action" value="Update">Update</button>
               <button type="submit" name="action" value="Delete">Delete</button>
           </fieldset>
   </form>
```

FAQ:Can we define multiple forms in a page?
A:Yes.Every form will have unique ID and Name

FAQ:Can we define nested forms?A form within another form?
A:No.

```
  <form>
      <form>
     </form>       //invalid
  </form>
```

28:)Regular Expressions(:28
==========

- Regular Expression is used to validate the format of input value.
- Regular expression is built by using
    a.Meta Characters
    b.Quantifiers

Meta Characters:

? :zero or one occurance of character

syntax:

```
 <input type="text" pattern="colou?r" name="test">
        color       -valid
        colour      -valid
```

+ :One or more occurance of character

syntax: pattern="colou+r"

```
       color  -invalid
       colour -valid
       colouur -valid
```

* :zero or more occurance

```
syntax:     pattern="colou*r"
       color  -valid
       colour -valid
       colouur -valid
       colouuur -valid

. [dot] :any single char, every specific character occurance is mandatory.
syntax:   pattern=".at"
       allowed- cat,bat,rat
         pattern=".o."
       allowed- toy,boy,dos
         pattern="..at"
      cat  -invalid
     chat  -valid

|  [pipe]  :Logical OR allows multiple chars or words
   syntax:     pattern="red|green|blue"
          valid-red,green,blue

\   :it is used as escape sequence character

d   :it specifies only numeric value
       pattern="d"

  \d    -any single digit numeric value type
  \d\d  -exactly 2 digit numeric
  \d\d\d -exactly 3 digit numeric

D   :it specifies non-numeric value
   pattern="\D"

  a   -valid
  9   -invalid
  $   -valid

w    :alpha numeric with underscore
       [a-z,A-Z,0-9,_]
 syntax :    pattern="\w"
   valid    =c,D,7,_
   invalid  =$,%

W       :non-word any special character other than a-z,A-Z,0-9,_

 syntax:       pattern="\W"

  invalid -c,D,5,_
   valid  -$,%,#,@

s       :it refers single blank space.
         [space or tab]
```

```
         space is mandatory
syntax:     pattern="\d\s\w"
        4A  -invalid
        4 A -valid
        4 6 -valid

i      :Ignore case

[a-z]         :only lower case alphabet
[A-Z]         :only uppercase alphabet
[a-zA-Z]      :Both upper and lower case
[a-Z]         :Both upper and lower case
[a,d,s]       :Only specified chars
[a-d,m-s]     :Only chars in specifed range allowed
[^a,d,s]      :excluding specified chars all other allowed
[^a-d,m-s]    :excluding specified range of chars others allowed
[0-9]         :Only numeric [similar to \d]
[a-zA-Z0-9]   :alpha numeric
[a-zA-Z0-9_]  :similar to \w
[^a-zA-Z0-9_] :similar to \W
\^            :starts with
\$            :Ends with
()            :Union of chars

Quantifiers:
-{n}     exactly specified number of chars
       {4}   exactly 4
-{n,m}   minimum -n and maximum -m
     {4,10}  between 4 to 10
-{n,}    minimum -n and maximum any
     {4,}  minimun 4 and maximum any
syntax:
   <input type="text" pattern="\d{10}">
   <input type="text" pattern="\d{2}-\d{2}-\d{4}">
        32-00-2342
         10-11-2023

29::29
------
Query-1: Name only uppercase letters 4 to 15 chars
 <input type="text" pattern="[A-Z]{4,15}

Query-2: Password alpha numeric with _ and special chars,range 4 to 15.
 <input type="password" pattern="\w[@#$%]">

Query-3: Bank IFSC code SBI3424HY
  <input type="text" pattern="SBI\d{4}[A-Z]{2}">
  <input type="text" pattern="SBI[0-9]{4}[A-Z]{2}">

Query-4: 10 digits mobile number starting with +91
```

```
  <input type="text" pattern="\+91\d{10}">

Query-5: Password 4 to 15 chars alpha numeric with atleast one upercase letter.
  <input type="text" pattern="(?=.*[A-Z])\w{4,15}">

       (?=.*[A-Z])
        (?=.*[0-9])
             ?     :zero or one
             ?=.   :atleast 1
             *     :many

Query-6: Password 4-15 chars alpha numeric with atlest one special char.

  <input type="password" pattern="(?=.*[!@#$%])\W[!@#$%^]{4,15}">

Query-7: password 4 to 15 chars alphanumeric with atleast one uppercase
letter,number and special char.
  <input type="password" pattern=
           "(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$^])\w[!@#$%]{4,15}">

 Upper
 Number
 Special
 (?=.*[A-Z0-9!@#$%])\w[!@#$%]{4-15}

Query-8: Write regular expression to validate Email address

Input Group for Form Elements:
------------------------------
.form-control   :textbox,email,date,url,password
  <input type="text" class="form-control">
  <input type="password" class="form-control">

.form-select    :dropdown
     <select class="form-select">

.btn
.btn-primary
.bit-danger       <button class="btn btn-danger>
.btn-sucess       <button class="btn btn-warning>
.btn-warning

.input-group       <div class="input-group input-group-lg">
.input-group-lg      your controls
.input-group-text  </div>

Netflix Register:
-----------------
        <div id="container">
        <div class="input-group input-group-lg">
```

```
            <input type="email" placeholder="Your email address"
class="form-control">
            <button class="btn btn-danger">
                Get Started
              <span class="bi bi-chevron-right"></span>
            </button>
        </div>
    </div>

Amazon search Bar:
------------------
     <div class="input-group">
        <select name="" id="" class="input-group-text">
           <option value="">All</option>
           <option value="">Electronics</option>
           <option value="">Footware</option>
        </select>
        <input type="text" class="form-control">
       <button class="btn btn-warning">
           <span class="bi bi-search"></span>
       </button>
   </div>

30:=====:30
-----------
                        Styles
                       ========
-Styles are attributes defined for HTML elements to make them more interactive and
responsive.

-styles can be defined in 3-ways
        1.Inline Styles
        2.Embedded Styles
        3.External style Sheets[CSS]
          [Cascade style sheets]


31:=====:31
-----------
1.Inline Styles  :
-Styles are defined for every element individually by using "style" attribute.
     <div style="attribute:value;"></div>
-they are faster.
-you can't reuse.

2.Embeded Styles :
-Styles are defined by using <style> container.
-you can reuse from any location within page.
-styles are not accessible to other pages.
      <style>
```

```
          selector
         {
           attribute: value;
         }
       </style>
```

FAQ:Where to embed the style container,in head or body section?
A:You can embed in both locations

FAQ:What is difference between head and body section?
A:styles in body section are intended to load into memory and apply when ever required.
   styles in body section are intended to apply on body load.

FAQ:What is MIME type for styles?
A: Multipurpose Internet Main Extension

```
          "text/css"
         <style type="text/css">


        </style>
```

FAQ:What is media type for styles?
A: Media type is used to target styles for printer,screen or speech.

```
            <style type="text/css" media="print">

            <style>

          <style type="text/css" media="screen">

           </style>
```

3.External Style Sheets:
-You can define styles in a separate style sheet with extension ".css"
-you can link to any page.
-Using external style sheet will increase the number of request for page that increases page load time.

step-1:create a new file with extension ".css"
step-2:link to your HTML page

```
         <link rel="stylesheet" href="effects.css">
```

FAQ:What is Minification?

FAQ:What is CDN?
A:COntent Distribution/Delhivery network
  it keeps the repository in a server and gives access from any remote location.

CSS Rules:

```
----------
Style syntax:

-inline style syntax
       <h1 style="styleAttribute: value; styleAttribute:value">

-Embeded or External File
     selector
        {
          styleAttribute: value;
              ;;
              ;;
        }

32:-====-:32
------------

Primary/Basic CSS Selectors:
----------------------------
1.Type selector    :it defines the element tag name.
                   :it applies to every element that occurs in page.
                   :you can't ignore for any specific element.
              h2 {

              }
              p {

              }
              table {

             }
                   :you can group elements
               h2 , p {

             }

2.ID selector       :ID is defined by using "#".
                    :it is accessed by using "id" attribute.
           #Name {

           }
           <h2 id="Name"></h2>
                    :you can apply effects only to the required emements.
                    :Every element can have only one ID.you can't apply multiple
groups of effects to single
                     element.
3.Class selector    :it is defined by using "."[dot]
                    :it is accessed by using "class" attribute.
                    :Dynamically it is accessed by using "className" property.
                    :you can apply multiple classes to single element.
```

```
       syntax:   .class1 {

                 }
                     .class2 {
                  }

              <h2 class="class1 class2"></h2>
Rules:
-if element is defined with type,id and class then which selector is used?
    1st priority : ID selector
    2nd priority : Class selector
    3rd priority : Type selector
-if all selectors are having different attributes,then all attributes are applied
to element.

Rational Selectors or Combinators:
---------------------------------
Apply effects based on parent and child hierarchy or adjacent or general sibilings.

1.Decendent selector :
       -it applies effects to all child elements under specific parent.
       -it can apply to any level.
            parent child {

            }
2.child selector  :
 -it applies effects to direct child.
 -it will not span effects to multiple levels.
            parent > child {

            }
            tbody> tr >td {

            }

3.Adjacent Sibling  :
  -it defines effect for immediate adjacent sibling [element] after the specifed
element.
            h2 + p {

            }
-it will apply only to the first paragraph that occurs after h2.

4.General Sibbling:
  -it defines effect for every elements that occurs after the specified.
  -it defines any level.
      h2 ~ p {

        }
```

```
  -it will apply for every paragraph after h2.

Attribute Selectors:
====================
-several HTmL elements are presented by using attribute of tag.
-Hence we have to apply effects based on attribute and its value.

 syntax:        tag[attribute="value"]
                 {

                }
         EX:     input [type="button"] {

                }
                 tag[attribute] {

                 }
     ex: p[id] {

           }
      p[class] {

     }
-Attribute selectors use conditions for values

Condition                         Description
-------------------------------------------
[attribute="value"]             Equal to
[attribute^="value"]            starts with
[attribute$="value"]            ends with
[attribute*="value"]            all occurance
[attribute|="value"]            uses space
[attribute~="value"]            uses space and "-"

33:----------------:33
======================
Dynamic Pseudo Classes:
-----------------------
-Dynamic allows to change according to state and situation.
-Pseudo refers to a "name" that maps to different element.
-Class is a pre-defined template that comprises of data and logic,which you can
implement
 and use with out re-writing.
  EX:      h2:ClassName {

           }
       #id:className {

         }
        .effect:className {
```

```
      }

1.:link           :effects for hyperlink at normal state.
2.:hover          :effects for eny element when mouse is hover.
3.:active         :effects for hyperlink at active state.
4.:visited        :effects for hyperlink when it is visited.

Syntax:   img {
            width: 100px;
            height: 100px;
          }
        img : hover {
             width: 200px;
             height: 200px;
            transition: 2s;
        }

Dynamic Element State Pseudo Classes:
-------------------------------------
-Element state refers to enabled,disabled,readonly,checked & focus.

- :enabled         :defines effects when enabled
  :disabled        :effects on disabled state
  :read-only       :effects on readonly
  :checked         :effects when checked
  :focus           :when element get focus.

syntax: button :enabled{
          cursor:grab;
       }
         button:disabled{
     cursor: not-allowed;
       }
    #UserName:read-only{
         backgroung-color:gray;
   }

EX: focus
        #userName:focus+span
        {
            display: inline;
        }
       #userName+span
        {
          display: none;
        }
             <input type="text" id="userName">
             <span>Name in Block Letters</span>
```

```
Dynamic Element Validation State Pseudo Classes:
------------------------------------------------
-validation state of element refers to required,optional,
valid,invalid,in-range,out-of-range etc.

:required        -for mandatory fields
:optional        -for optional fields
:valid           -if all validation attributes are valid
:invalid         -if any one validation attribute is invalid
:in-range        -if value is with an specified numeric range
:out-of-range    -if value is out of specified numeric range

synax:  #userName:required
        {
          border: 2px solid black;
         }
       #userName:required+span {
             display: block;
             color:red;
      }
      #userName:valid+span {
     display:none;
       }

  <input type="text" id="userName" required>
    #userName:optional {
              border: 2px solid gold;
      }

34:------------:34
==================
EX: pattern,minlength,Email,Url =>valid & invalid

        #Mobile:valid {
           border: 2px sloid green;
           box-shadow: 2px 2px 3px green;

        }
        #Mobile:invalid {
            border: 2px solid red;
            box-shadow: 2px 2px 3px red;
        }
        #Mobile:valid+span {
            display: none;
        }
        #mobile:invalid+span {
            display: block;
            color: gold;
        }
    <h3>Register User</h3>
```

```
    <dl>
        <dt>Mobile</dt>
        <dd>
            <input type="text" id="Mobile" pattern="\+91\d{10}">
            <span>Invalid Mobile</span>
        </dd>
    </dl>

EX:Numeric Range:
    <dl>
        <dt>Age</dt>
        <dd>
            <input type="number" value="15" id="Age" min="15" max="25">
            <span>Age 15-25 only</span>
        </dd>
    </dl>
  <style>
     #Age:in-range {
        border: 2px solid green;
        box-shadow: 2px 3px 3px green;
     }
      #Age:out-of-range {
        border: 2px solid red;
        box-shadow: 2px 3px 3px red;
       }
       #Age:in-range+span {
        display: none;
       }
       #Age:out-of-range+span {
        display: inline;
       }
  </style>

Stuructural Pseudo Classes:
---------------------------
-These classes are used to apply effects based on their occurance.

:first-child  -to first child
:list-child   -to last child
:nth-child()  -specific child element
               (2)    -2nd child
               (2n)   -Every 2nd child
               (2n+1) -Every 2nd child starting with 1
               (odd)  -Every odd occurance
               (even) -Every even occurance

       syntax:  li:nth-child(odd) {
                   color: red;
                }
:nth-of-type       -it is for every nth occurance
```

```
:nth-last-of-type  -for every nth occurance from bottom
:root              -
:empty             -
  syntax: Empty and Root
     td:empty {
      background-color:red;
  }
:root {
  font-family:Arial;
 }

Pseudo Element Classes:
-----------------------
::      -These are the classes applied to element that occurs to elements
before,after or first
    and last of existing hierarchy.
        -these classes are applied to any element using "::",which means implement
and inherit.

   ::     Adding any functionality with specific rules
The element classes are
::before         -specifies content to display before existing
::after          -specifies content to display after existing element
::first-line     -effects for first line in a container
::first-letter   -effects for first letter in a container
::placeholder    -effects for place holder
::selection      -effects for selected content

EX: for before and after
       ol {
        list-style: none;
        display: flex;

       }
       li::before{
        content: "-->";
       }
       li:first-child::before{
          content: "";
       }

   <ol>
    <li>Home</li>
    <li>About</li>
    <li>Contact</li>
    <li>Blog</li>
   </ol>

EX:DropCap
     p::first-letter{
```

```
        font-size:40px;
        font-weight:bold;
        color:red;
     }
   p::first-line {
    color:blue;
   }
  p::selection {
    background-color: yellow;
   }

EX:placeholder
 #userName::placeholder {
    color:red;
 }

                          CSS Units
                          ---------
-Absolute Units  :will apply for any specified element
         in  -inch
         cm  -centimeter
         pt  -points
         px  -pixles
         %   -fluid
-Relative Units  :will apply for any element based on its parent.
        em   -element
        rem  -root element

35:----------------:35
======================
                CSS Colors
                ----------
-color Name
-Hexa Decimal COde
-RGB()
-RGBA()

color Name:
      h2{
         color:red;
      }
Hexa Code:
     h2{
        color: #ff0000;  [#foo]
     }

RGB() :
  h2{
   color: rgb(255,0,0);
  }
```

```
RGBA():
   h2{
      color: rgba(255,0,0,0.6);  [A=0.0 to 1]
   }

                      CSS Inheritance
                      ---------------
-inherit  :implements the parent effects
-initial  :sets the default effects
-unset    :clears all effects

->Inheritance is the process of extending functionality and allowing to re-use the
existing values.

 syntax:  body {
              color:red;
          }
           p{
            color: initial;
          }

                  CSS Box Model
                 ---------------
-Margins   :sets space between the container and page margins.
  margin             all directions[short hand]
  margin-left
  margin-right
  margin-top
  margin-bottom

-padding   :sets space around contnt
 padding           short hand -all direction
 padding-left
 padding-right
 padding-bottom
 padding-top

  syntax:
        p{
          border: 2px solid black;
          padding: 10xp 10px 5px 5px;
        }

-border    :sets border for container
 border         shorthand
 border-left    left shorthand
 border-right   right shorthand
 border-top     top shorthand
 border-bottom  bottom shorthand
```

```
border-left-style  double,dashed,solid,groove
border-right-style
border-top-style
border-bottom-style

border-left-color  name,code
border-right-color
border-top-color
border-bottom-color

border-left-width   units[px]
border-right-width
border-top-width
border-bottom-width

border-radius   :it can define rounded corners [shorthand]
border-top-right-radius
border-top-left-radius
border-bottom-left-radius
border-bottom-right-radius

border-image     :it sets img as border
Note: You can't apply border-image without configuring border.
syntax :
          p {
           border: 10px solid red;
           border-image: url("path");
          }
Border image requires following values to configure
a.url()   :it specifies the path of image to display
b.style   :it defines the display style stretch or round
c.offset  :it defines the offset size for display style.
   syntax:
       body {
         border: 30px solid red;
         border-image: url("assets/border.png") stretch 50;
         padding: 20px;
       }

              CSS Postiton
              ------------
-static
-relative
-sticky
-fixed
-absolute
 syntax:
       div {
         position: fixed;
      }
```

```
static position  :it keeps the element according to normal document flow.
                 :it is the default position.
                 :it is not effected by left,right,top or bottom values.

fixed position   :it removes element for normal document flow.
                 :it keeps the element at specified position using left,right,top
and bottom.
                 :its position is fixed and locked from scrolling.

Absolute position :same like fixed but will not lock scrolling.
   syntax:  #offer{
               position: fixed;
               right: 10px;
               top: 100px;
            }

sticky position  :it keeps the element to normal flow of document.
                 :it locks scrolling after reaching specifeid position
top,left,bottom or right.
Relative position  :it sets position of element relative to its parent.
                   :it is according to normal document flow.
                   :it is effected by using left,right top and bottom values.
        #parent {
          width:300px;
          height:300px;
          position:static;
        }
   #child{
     width:100px;
     height:100px;
     position:relative;
   }
        <div id="parent">
           <div id="child">
       </div>
      </div>
                              CSS Float
                              ---------
-Float allows to keep the content absolute left or right.
-it provides only left and right values.
           #container {
              float: left;/flaot: right;
          }
                            CSS Display
                            -----------
-it controls the display style of content on page.
-it comprises of values.
```

```
      -none
      -inline
      -block
      -grid
      -flex
Display:None  -it hides the element.
Display:block -it will display element in next line.
Display:inline-it will display element in same line.
Display:grid  -it configures grid to arrange information into rows and columns.
Display:flex  -it defines a flexible grid,that can adjust according to page and
device.

EX: hide element   :  img {
                         display: none;
                       }

Ex: display inline  :
         .form-group {
          display: inline;
        }
       <div class="form-group">
            <div>
                 username
            </div>
           <div>
                password
            <div>
         </div>
ex: display: block;
      .form-group div {
           display: block;
      }
EX: display: flex;
   -it uses a flex container
   -the flex container can configure
         a.flex-direction
         b.flex-wrap
syntax:  .container {
             display: flex;
             flex-wrap: wrap;
             flex-direction: column/row/column-reverse/row-reverse;
          }

 Note: You can align items in a flex container horizontally and vertically by using
the attributes
         a.justify-content
         b.align-items

  justify-content: space-between,around,evenly,center.
  align-items: center,baseline,stretch.
```

EX: Display grid:
-it organises information into specific row and column.
-it creates flexible and responsive layout.
syntax:
```
div {
  display: grid;
 grid-templet-column:200px 2fr[12fr];
 grid-row: rowNumber;
 grid-column: columnNumber;
}
```
-display grid allows to organized information into row by using "grid-row".
-it specifies the row number.
-the attribute "grid-column" arranges into specific column.
-the values for row and column are defined in points.

```
grid-row:2;
grid-row:1;

grid-column: 1/4   1 to 4 columns
grid-column:1/1    only first column
```

NOTE:You can re-arrange your content into row and column by changing the row and column numbers.

style:
```
<style>
  section {
    height: 500px;
  }
  header, footer {
   background-color: tomato;
   color: white;
   text-align: center;
   padding: 10px;
  }
  .container {
   display: grid;
   grid-template-columns: 3fr 6fr 3fr;
  }
  header {
   grid-row: 1;
   grid-column: 1/4;
  }
  footer {
   grid-row: 3;
   grid-column: 1/4;
  }
  section {
   grid-row: 2;
   grid-column: 1/4;
   display: grid;
   grid-template-columns: 3fr 6fr 3fr;
  }
```

```
    </style>

    <div class="container">
        <header>
            <h1>Amazon Shopping</h1>

        </header>
        <section>
            <nav>
                <ul>
                    <li>Electronics</li>
                    <li>Footware</li>
                    <li>Fashion</li>
                </ul>
            </nav>
            <main>
                <h3>Shop | Online</h3>
                <p>Special Offers On Electronics</p>
            </main>
            <aside>
                50% OFF
            </aside>
        </section>
        <footer>
            <p>&copy; copyright 2023</p>
        </footer>
    </div>
```

CSS Columns

-CSS Columns are used to display continous content.
-the content of one column will span to next column automatically after specific height.

```
syntax :  section {
            columns-count:4;
            column-gap:20px;
             column-rule: 2px dotted black;
            column-width: 200px;
           }
```

FAQ:What is difference between grid and columns?
A:grid keeps content fixed to a specific row and column.
  columns allow to span into next column.

CSS Z-Index Positi

-it arranges your content from bottom to top on z-axis.
-it uses index position 1=bottom,2=over 1,3=over 2 etc.

```
syntax:  .div-1 {
            width: 100px;
            height: 400px;
```

```
            z-index:1;
          }
         .div-2 {
             height: 100px;
             width: 500px;
             z-index: 2;
         }
                                CSS Background
                               ----------------
-background   short hand
-background-color
-background-image           -url(""),url("")
-background-position        -horizontal-position v-position [pixels or
values-left,right,center,top,center,bottom]
-background-size            -100px 200px;
-background-attachment      -fixed,scroll
-background-repeat          -repeat|no-repeat|repeat-x|repeat-y

FAQ:Can we display multiple images in background?
A: Yes

38:_--------------------_:38
============================
                                  CSS Text Effects
                                 ------------------
-formating text
1.color                  -specifies color for text
2.font-family            -specifies the font name
3.text-shadow            -sets shadow for text [horizonatal,vertical,blur]
4.font-style             -sets italic
5.font-weight            -sets bold
6.text-transform         -changing the capitalization [effect-size]
7.font-variant           -changing the capitalization [without changing the size]
8-text-indent            -first line padding
9.line-height            -sets line spacing
10.word-spacing          -sets spacing between words
11.white-space           -it defines wrapping of text
12.text-overflow         -it control the text that overflow
13.text-direction        -better use "text-orientation" [along with CSS 2D and 3D
effects]
14.font-size             -set size for chars.
15.letter-spacing        -sets space between chars.
16.text-align            -Aligns left, center, right or justify.
17.word-break            -it controls the display of lengthy word in paragraph.
18.text-decoration       -it controls underline,overline,line through

FAQ:What are the web safe fonts?
A:serif,sans-serif,monospace

syntax:Text Overflow
```

```
      p{
         border: 2px solid black;
         width: 300px;
         height: 80px;
         white-space:nowrap;
         overflow: hidden;
         text-overflow: ellipse;
      }

NOTE: CSS Attributes have browser dependency issues.few attrbutes are not available
      for various browsers. hence we have to use plugins suitable for browser
engine.
            webkit   :edge,chrome,safari,opera
            gecko    :firefox
            ms       :internet explorer
            os       :opera


           word-break : break-all;
           webkit-word-break: break-all;
           moz-word-break: break-all;
           ms-word-break: break-all;

Ex:text decoration

text-decoration       -short hand
text-decoration-style -inline style[dotted,wavy,dashed,solid,groove]
text-decoration-color -color for underline,line through,over line
text-decoration-line  -underline,line through,overline

                                    CSS 2D Transforms
                                    -----------------
-two dimensional effects
-along x-axis and y-axis
-they are used to control width,height,orientaion etc.
-2d transforms are defined by using "transform" attribute.
-use with transform attribute the following things.
-      syntax:
                    transform:   ;
-2D effects are
       scale()
       skew()
       rotate()
       matrix()

-scale()        :it is used to control the size dynamically.[height and width]
                scaleX()          width
                scaleY()          height
                scale()           both
```

```
-rotate()       :it is used to rotate by specified angle.
                rotateX()
                rotateY()
                rotate()
             45  :45 deg clockwise
            -45  :45 deg counter-clockwise

-skew()          :it is used to tilt element by specified angle
                skewX()
                skewY()
                skew()
-translate()   :it is used to move the element along x and y axis.
               translateX()
               translateY()
               translate()

39:-----------------------------:39
====================================
                          CSS 2D Matrix
                          -------------
-it allows to define multiple transforms.

syntax: matrix(scaleX(),skeyX(),skewY(),scaleY(),translateX(),translateY())

Note:
-the values defined for matrix are configured by using points as unit.
-the order dependency of effects for matrix is standard.
-you can't ignore any specific effect,you have to define zero as value.

syntax:
        transform: matrix(1,0,0,1,100,100);
 -you can define fractions as units.
         transform: matrix(1.5,20,0.5,100,100);
  scale=1.5
  skew=20 deg
  translate= 100px
 EX:
       img {
        width: 100px;
        height: 100px;
        transition: 2s;
      }
      img:hover {
        transform: matrix(1.5,15,15,1.5,0,0);
        transition: 2s;
      }
 1.5  width
 15   skewX
 15   skewY
 1.5  height
```

```
 0    translateX
 0    translateY
```

CSS 3D Transforms

```
-3 dimensional effects along X,Y,Z axis.
-all attributes are similar to 2D but contains a new dimension z-axis.
1.translate3d()
2.scale3d()
3.skew3d()
4.rotate3d()
5.matrix3d()

Note:The 3rd dimension is possible to view only when border, shadow and background
are defined.
syntax:
        scale3d(xPosition, yPosition,zPosition);

EX:
      <style>
      img {
        width: 100px;
        height: 100px;
        box-shadow: 12px 12px 2px black;
        border:1px dotted yellow;
        transition: 2s;
      }
      img:hover {
        transform: scale3d(3,3,5);
        transition: 2s;
      }
      .back {
          background-image: url("assets/shirts.jpg");
          backgoround-size: 200px 200px;
          border: 2px solid black;
          width: 200px;
          height: 200px;
      }
      </style>

       <div class="back">
            <img src="assets/shirt.jpg">
       </div>
```

CSS Transition

```
-transition define animation,duration,delay and timing function.
-the transition attributes

 transition                    :short hand
 transition-delay              :the delay time for starting animation
```

```
 transition-duration           :the total duration of animation
 transition-timing-function    :pre defined animation effects.
 transition-property           :it defines the property to effect.
                                [by default animation will effect for all
                                 properties] width, height

syntax:
          .box{
             width: 100px;
             transition-duration: 2s;
             transition-delay: 5s;
             transition-color: red;
         }
          .box:hover {
             width: 1200px;
             transition-duration: 5s;
             transition-delay: 2s;
             transition-property: width;
             background-color: yellow;
          }

-transition-timing-function:
syntax:
       transition-timing-function: step(3);
       transition-timing-function: ease-in;
                         CSS Keyframes/Animations
                        -------------------------
-it alows to design animation for every frame.
-frame-by-frame animation
-keyframes are used to define animation effects.
-animation attributes are used to apply and configure keyframes to any element.

FAQ:What is keyframe?
->Animations are designed by using frames.
-usually animation contains 30fps[frames per second] or 60fps.
-frames are two types
  a.static frame
  b.key frame
-static frame will not define any animation for element [unchanged].
-key frame will define animation for element [changes according to time interval].

@keyframes referenceName
{
   inital-state{

   }
   final-state {

   }
}
```

```
 selector {
    animation-name:referenceName; [keyframe name]
 }

 initial-state        :form
 final-state          :to
 intermediate-state   :percent or pixels

-Animation attributes:
----------------------
-animation-name              :keyframe name
-animation-duration          :animation time
-animation-delay             :delay time to start animation
-animation-property          :properties to effect
-animation-direction         :animation direction-reverse,alternate
-animation-iteration-count   :number of times to display count in number infinite
-animation-timing-function   :pre-defined effects easeln, steps,easeOut etc.

EX: <style>
        @keyframes imageEffects {
            from {
                margin-left: 100%;
            }
            20% {
                transform: rotate(30deg);
            }
            50% {
                height: 300px;
                width: 300px;
                transform: rotate(360deg);
            }
            80% {
                transform: rotate(3deg);
            }
            to {
                margin-left: 0%;
            }
        }
        body {
            overflow: hidden;
        }
        img{
            animation-name: imageEffects;
            animation-duration: 3s;
            animation-direction: alternate;
            animation-iteration-count: infinite;
        }
   </style>
```

```
                                        ------------------------
-content must fit according to device.
-responsive designs are created by using
   a.fluid images
   b.fluid grids
   c.media properties
-fluid images are defined with height and width in % persentage.
  <img src="" width="50%" height="30%>
-fluid grid are defined by using "display:flex" and "flex:container" options.
   {
   display: flex;
   flex-wrap: nowrap;
   flex-direction: row;
   }

-media Properties use "viewport" to control the appearance of content on screen
according
 width,height and orientation.
               @media mediaType and mediaProperties
               {

               }
             mediaType       :screen,print,speech
             mediaProperties :orientaion,width,height,min-width,max-width
-Media Queries are configured using conditions
  @media screen and (orientaion: landscape)
 {

 }
 @media screen and (orientaion: portrait)
 {

 }
 @media screen and (min-width: 500px)
 {

 }
 @media screen and (max-width: 500px)
 {

 }

FAQ:What is difference between width and "min-width" or "max-width"?
A:   width             :defines exact width
     min-width         :defines from specified width
     max-width         :defines upto specified width

  Ex: <head>
       <style>
            ul {
```

```
    list-style: none;
    border: 2px solid darkcyan;
    padding: 5px;
    border-radius: 5px;
    background-color: darkcyan;
    color: white;
}
@media screen and (orientation: landscape) {
  li {
    display: inline;
    margin-right: 30px;
  }
}
@media screen and (orientation: portrait) {
    li {
        display: block;
        margin-bottom: 30px;
    }
    ul {
        width: 30px;
        margin-left: -40px;
        transition: 3s;
    }
    ul:hover {
        margin-left: -10px;
        transition: 3s;
    }
    .menu-title {
        display: none;
    }
}
@media screen and (max-width: 500px){
    body {
        background-color: yellow;
    }
    li {
        display: block;
        margin-bottom: 30px;
    }
    ul {
        width: 30px;
        margin-left: -40px;
        transition: 3s;
    }
    ul:hover {
        margin-left: -10px;
        transition: 3s;
    }
    .menu-title {
        display: none;
```

```
                }
            }

        </style>
    </head>
    <body>
        <nav>
            <ul>
                <li><span class="bi bi-facebook"></span><span
class="menu-title">Facebook</span></li>
                <li><span class="bi bi-twitter"></span><span
class="menu-title">Twitter</span></li>
                <li><span class="bi bi-instagram"></span><span
class="menu-title">Instagram</span></li>
                <li><span class="bi bi-linkedin"></span><span
class="menu-title">Linkedin</span></li>
            </ul>
        </nav>
    </body>
```

CSS Gradients

- Gradient colors are a combination of multiple colors with linear and radial orientation.
- linear will be towards left,right,top or bottom.
- Radial will be from center
- Gradient colors can be applied only with background as image

a. linear-gradient() to right,to left,to top,to top left,to top right
b. radial-gradient() to center,to offset

NOTE: you can apply multiple color with color name,hexa code,rgb(),rgba()
you can also define the % of color.