

HTML

→ HTML stands for hypertext markup language.

Q What is internet?

→ The internet is a vast network that connects all over the world.

Q What is network?

→ A network consists of two or more computers that are linked in order to share resources (such as printers and CDs), exchange files, or allow electronic communications.

Q What is protocol?

→ Protocol, in computer science, a set of rules or procedures for transmitting data between electronic devices, such as computers.

Q What is URL and example?

→ A URL (Uniform resource locator) is a unique identifier used to locate a resource on the internet. It is also referred to as a web address. URLs consists of multiple parts - including a protocol and domain name - that tell a web browser how and where to retrieve a resource.

Date _____
Page _____
<https://www.website.com/blog/blog-post-one/>

Protocol

Sub-Domain

Second Level Domain (Domain)

Top level Domain (TLD)

Folder / path

page

↳ What is HTTP and HTTPS difference?

→ HTTPS is HTTP with encryption. The only difference between the two protocols is that HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses.

↳ What is HTML?

→ HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. Hypertext is text which is not constrained to be linear. Hypertext is text which contains links to other texts.

Markup is what HTML tags do to the text inside of them.

The first version of HTML was written by Tim Berners-Lee in 1993.

- The `<!doctype html >` declaration defines that this document is an HTML5 document.
- The `<HTML >` element is the root element of an HTML page.

- The `<Head>` element contains meta information about the HTML page.
- The `<Title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab).
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

↳ What is an HTML document?

An HTML element is defined by a start tag, some content, and an end tag.

`<tag name>` content goes here `</tag name>`

↳ Web browsers.

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags.

↳ A simple text editor is all you need to learn HTML.

Step 1: Open Notepad (pc)

Step 2: Write some HTML.

Step 3: Save the HTML page.

Step 4: View the HTML page in your browser.

↳ The `<! Doctype >` declaration is not case sensitive

HTML Headings

HTML Paragraphs

↳ HTML Basic:

The `<! Doctype >` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The `<! Doctype >` declaration is not case sensitive

The `<! Doctype >` declaration for HTML5 is:

```
<! Doctype html >
```

↳ HTML Headings:

HTML headings are defined with the `<h1 >` to `<h6 >` tags.

`<h1 >` defines the most important heading. `<h6 >` defines the least important heading.

↳ HTML paragraphs:

HTML paragraphs are defined with the `<p >` tag.

↳ HTML links:

HTML links are defined with the `<a >` tag:
``
This is a link ``

Date _____
Page _____

The link's destination is specified in the href attribute. Attributes are used to provide additional information about HTML elements.

↳ HTML Images:

HTML images are defined with the `` tag. The source file (`src`), alternative text (`alt`), width, and height are provided as attributes.

example:

```

```

↳ Nested HTML Elements:

HTML elements can be nested (this means that elements can contain other elements). All HTML documents consist of nested HTML elements.

↳ Empty HTML elements:

HTML elements with no content are called empty elements.

The `
` tag defines a line break, and is an empty element without a closing tag:

example:

```
<p> This is a <br> paragraph with a line break. </p>
```

↳ HTML Attributes :

- All HTML elements can have attributes.
- Attributes provide additional information about elements.
- Attributes are always specified in the start tag.
- Attributes usually come in name / value pairs like:
Name = " value".

↳ The href Attribute.

The `<a>` tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to.

example :

```
<a href = "https://www.google.com" > Google </a>
```

↳ The Src Attribute.

The `` tag ~~should~~ is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed.

example :

```
<img src = "meezan.jpg" >
```

↳ The width and height Attributes.

The `` tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):

example :

```
<img src = "meezan.jpg" width = "200px" height = "300px" >
```

↳ The alt Attribute :

The required alt attribute for the tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

example

```

```

↳ HTML Comments :

HTML comments are not displayed in the browsers, but they can help document your HTML source code.

You can add comments to your HTML source by using the following syntax :

```
<!-- write your comments here -->
```

↳ HTML block and Inline elements :

There are two display values : Block and inline.

↳ Block-level elements.

A block-level element always starts on a new line and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available.

The <p> element defines a paragraph in an HTML document.

```
<h1> - <h6>   <li>   <ol>   <p>   <ul>
```

↳ Inline elements:

An inline element does not start on a new line.
An inline element only takes up as much width as necessary.

```
<a> <button> <img> <input> <label>
<textarea>
```

↳ HTML class Attribute:

```
<h2 class = "name" > Alfurqan </h2>
```

↳ Multiple classes:

```
<h2 class = "name main" > welcome to alfurqan computer
</h2>
```

↳ Different elements can share same class.

```
<h2 class = "name" > haries </h2>
<p class = "name" > follow @ malik-haris -1 </p>
```

↳ HTML Id Attribute.

The HTML id attribute is used to specify a unique id for an HTML element.

You cannot have more than one element with the same id in an HTML document.

```
<h1 id = "my header" > my header </h1>
```

Note: The id name is case sensitive.

Note: The id name must contain at least one character. Cannot start with a number, and must not contain

whitespaces (spaces, tabs, etc).

↳ Difference Between class and ID.

A class name can be used by multiple HTML elements, while an ID name must only be used by one HTML element.

↳ Ordered HTML list:

An ordered list starts with the `` tag. Each list item starts with the `` tag.

The list items will be marked with numbers by default.

Example:

```
<ol>
<li> coffee </li>
<li> Tea </li>
<li> milk </li>
</ol>
```

↳ Ordered HTML list - The type Attribute

`type="1"`, `type="A"`, `type="a"`, `type="I"`, `type="i"`

↳ Control list counting.

```
<ol start="50">
```

↳ Nested HTML lists:

```
<ol>
<li> coffee </li>
<li> Tea
<ol>
```

```
<li> Black Tea </li>
```

```
<li> Green Tea </li>
```

```
</ol>
```

```
</li>
```

```
</li> milk </li>
```

```
</ol>
```

↳ Unordered HTML list.

An unordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with bullets (small black circles) by default:

Example:

```
<ul>
```

```
<li> havis </li>
```

```
<li> Faizan </li>
```

```
<li> mazaan </li>
```

```
</ul>
```

↳ HTML Forms.

An HTML form is used to collect user input.

↳ The `<input>` Element.

An `<input>` element can be displayed in many ways depending on the type attribute.

Radio, text, password.
checkbox, date, submit.

↳ The <label> element:

The for attribute of the <label> tag should be equal to id attribute of the <input> element to bind them together.

↳ The Submit button:

The <input type = "Submit"> defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's action attribute.

```
<form action = "/action-page.php">
```

```
<input type = "Submit" value = "Submit">
```

```
</form>
```

↳ The Name Attribute for <input>:

Notice that each input field must have a name attribute to be submitted.

If the name attribute is omitted, the value of the input field will not be sent at all.

↳ HTML TABLES:

HTML tables allow web developers to arrange data into rows and columns.

Example:

Company	Contact	Country
MS	Umee	American
Apple	Bill	American.

<table>

<tr>

<th> company </th>

<th> contact </th>

<th> country </th>

</tr>

<tr>

<td> ms </td>

<td> umes </td>

<td> American </td>

</tr>

<tr>

<td> Apple </td>

<td> Bill </td>

<td> America </td>

</tr>

</table>

th for headings, tr for table row, td for table data.

Introduction to Internet Technology.

1. Introduction : The internet is a collection of computers around the world connected to each other via a high speed series of networks. The internet becomes the main method in exchanging cultures and transferring knowledge between people. All connected computers and networks exchange information and use various services. Hence, the internet is not the world wide web (WWW or W3). The world wide web (or web) is just one of the services that the internet offers to its users although it is the most commonly used service. The internet is the global system of interconnected computer networks. The internet carries an unlimited range of information resources and services, such as the interlinked hypertext documents and applications of the web, electronic mail, telephony, and file sharing.

2. The web concepts : The web was developed to be a pool of human knowledge, and human culture, which would allow collaborators in remote sites to share their ideas and all aspects of a common project. The web consists of a vast assortment of files and documents that are stored on these computers and written in some form of Hyper text markup language (HTML) that tells browsers how to display the information. The computers that store the files are called servers because they can serve requests from many users at the same time. Users access these HTML files and documents via applications called browsers.

↳ The main web concept are:

web page the web page is a space of information on the internet that presents information about a particular person, business, or organization or cause

- The web consists of files, called web pages (documents)
- It is containing links to resource (text, images, audios, videos, and other data), throughout the internet.

↳ website:

website can be defined as a collection of web pages which are grouped together and usually connected together in various ways. often called a "web site" or simply a "site". The website usually presents information about a particular person, business, organization or cause. Generally there are two types of website style, static and dynamic website.

- A static website is one that has web pages stored on the server in the same form as the user will view them.
- A dynamic website is one that does not have web pages stored on the server in the same form as the user will view them. Instead, the web page content changes automatically and/or frequently based on certain criteria. There are two meaning for dynamic website. The first is that the web page code is constructed

Date _____
Page _____

dynamically, piece by piece. The second one is that the web page content displayed varies based on certain criteria. The criteria may be predefined rules or may be based on variable user input.

↳ web hosting :

After we create a website, it is necessary to store it in a place where it is always available for users. We use web hosting services / companies for this purpose. They own web servers that have the ability to store content.

↳ Search engine :

Search engines allow us to search in the internet for information, images and other types of files stored in various locations available throughout the network. Search engines work according to certain algorithms that show the user relevant, requested search results. At first, information from the various websites is collected and stored and then analyzed to organize and save them in the database for future use. When a user enters an inquiry into the search engine, the database is organized by an Index and the user is presented with results that match best the entered search terms. Therefore, it is important to understand that by using the search engine one does not search the entire internet but the search engine's database. Therefore, we might get different results each time we use different web engines.

↳ Internet protocols:

- **TCP/IP protocol:** This protocol is the main protocol for internet communication. It defines the rules that computers need to follow to communicate with other computers via a network.
- **HTTP (Hypertext Transfer protocol):** is a network protocol that ensures the exchange of almost all types of resources on the web. With resources, we mean files, pages, images, search results, etc. It is basically, the web browser's language to send requests to the server. There is also the HTTPS version, which is a secure and encrypted version for HTTP communications.
- **FTP (File Transfer protocol):** is used to load (upload) or retrieve (download) files between the client and server communication, or between two computers on the internet. In other words, this protocol is responsible for file transfer on the internet.
- **Internet Service Provider (ISP):** is an organization that provides individuals and other companies access to the internet and other related services such as email.

↳ Web Design : Client - Server Architecture :

1. Client - Server Architecture computers on the internet use client / server architecture. This means that the remote server machine provides files and services to the user local client machine.

1.1 Web Server : A web server is a computer purposed to runs special serving software. That software "serves" HTML pages and the files associated with those pages when requested by a client, usually a web browser. The computer is secured so that only authorized people can access it to make changes to the data. So, if a person is on the same network as the web server, he or she may be able to save the data directly onto the web server computer (if authorized).

↳ Server - Side :

- Jsp (Java - Server pages)
- Asp (Active - Server pages)
- ASP. Net (next generation of Asp)
- PHP.
- Python.

1.2 Client.

1.3

1.4. The client (front end) or user side of the web, it typically refers to the web browser in the user's machine. It may

also refer to plug-ins and helper applications that enhance the browser to support special services from the site. The term may imply the entire user machine or refer to a handheld device that provides web access.

1.5 client - side:

1.6.

1.7 • HTML / XHTML (Extensible Hypertext Markup Language)

1.8 • CSS (Cascading Style Sheets).

1.9 • JavaScript / VB Script (client-side scripting)

1.3 Web Browsers Generally.

1.4.

1.5 A web browser is a software application or program for retrieving, displaying, and traversing information resources on the world wide web. An information resource is identified by a uniform resource identifier (URL) and may be a web page, image, video, or other piece.

1.6 2. web operations:

1.7.

1.8. The web works on client-server approach. Following steps explain how the web works:

1. User enters the URL (say, <http://www.tutorialspoint.com>) of the web page in the address bar of web browser.

2. Then browser requests the Domain name server for the IP address corresponding to www.tutorialspoint.com.

3. After receiving Ip address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and check its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, it interprets it and display the contents of web page in web browser's window.

1.9

1.10. 3. Site Structure :

1.11: Every web site was built in inherits structure and should have a consistent and simple organization called a site structure. A site is collection of HTML files, documents and images contained in a single master folder (the root folder). Within this root folder you can save your documents and subfolders organized in a manner that makes sense to you, as well as to others in your department that may need to edit the information. Therefore it is recommended that the structure of web site include:

- 1.12. 1. A root folder that contains the web site.
2. A web page entitled index.htm (or index.html) that resides within the root folder to represent the default homepage for the web site.

3. May be images folder that contains the graphics, illustrations, images and photographs used in the web pages.

4. Additional folders for organizing the content.

3.1 Hyperlinks:

Hyperlinks are the primary method used to navigate between pages and websites. Links can point to other web pages, web sites, graphics, files, sound, e-mail addresses, and other locations on the same web-page. When text is used as a hyperlink, it is usually underlined and appears as a different color. There are four types of hyperlink. Text hyperlink - uses a word or phrase to take visitors to another page, file or document. • Image hyperlink - uses an image to take visitors to another page, file or document.

- Bookmark hyperlink: Uses text or an image to take visitors to another part of a web page.
- E-mail hyperlink: Allows visitors to send an e-mail message to the displayed email address.

3.2. How Browsers Display web page:

When a web page is opened in a browser, the browser reads and interprets the HTML file and formats the webpage for display. If there are references to external files, such as images or multi-

media, these files are downloaded from the server and displayed on the browser window. It is important to note that HTML files are text files that only contain references to the external files - you do not "embed" these files into the web page.

4. Types of web site :

There are many types of websites, each catering to a particular type of content or use. Hence, few illustrative but not exhaustive cases are given below:

- 1) Blog (web log) : Site generally used to post online diaries which may include discussion forums.
2. Social Networking Site : where users could communicate with one another and share media, such as pictures, videos, music and blogs with other users. These may include games and web applications.
3. Wiki site : which users collaboratively edit (such as wikipedia and wikihow).
4. web portal : that provides a starting point or a gateway to other resources on the internet.
5. Search Engine site : a site that provides general information and is intended as a gateway or lookup for other sites like Google, Yahoo, Bing Search engines.

6. **Education Site:** where teachers, students, or administrators can post information about current events or involving their school.

5. **Website Design issues:**

There are many points one should keep in mind while designing a web site. These points make a site usable and easily accessible to the users, which are discussed below:

1. **Information Availability:** All the information that helps a visitor make informed decisions should be in the website. The general benchmark of a good web site is that it should be providing 80-90% of what a user/visitor is looking for.

2. **page layout:** How the content is displayed on the page - the page should be laid in such a way people should be able to find relevant content quickly else they lose interest and leave the site. page should be simple to enhance the usability of the page.

3. **Colors:** Standard colors should be used as colors can affect the usability of the site. For example, the standard colors used for links (blue for links; violet for visited links; and red for active links) should be maintained as one runs the risk of confusing the visitors.

- 4) web Accessibility :- web accessibility means accessibility to all including people with disabilities (like visual, audio, physical, speech) can interact or contribute to the web. As more and more accessible web sites and software are made available, people with disabilities should be able to use and contribute the web more effectively.
- 5) User-friendly Site :- web site should be creating based on what the audience requires.
- 6) Download Speed :- The download speed cannot determine by a web designer. Some considerations are optimizing the graphics (images not larger than 10 Kb), making smaller pages, and avoiding nested tables.

INTRODUCTION TO CSS

HTML is just an skeletal layout of a website. We need CSS to design a website, add styles to it and make it look beautiful.

↳ What is CSS

CSS stands for Cascading Style Sheets. CSS is optional but it converts an off looking HTML page into a beautiful and responsive website.

↳ Our first line of CSS

Create a CSS file in our directory and add it to our HTML. Add the following line to our CSS body

```
body {  
    background-color: red;  
}
```

This will make our page background as red.

CHAPTER :- 1

CREATING OUR FIRST CSS WEBSITE

We can create our first CSS website in this section.

↳ THREE WAYS TO ADD CSS TO HTML:

There are three ways to add CSS to HTML:

1. `<Style > tag` → Adding `<style > ... </style >` to HTML
2. `Inline CSS` → Adding CSS using style attribute
3. `External CSS` → Adding a stylesheet (.css) to HTML using `<link >` tag.

↳ CSS SELECTORS:

A CSS Selector is used to select an HTML element(s) for styling.

```

body {
  color : red ; → Declaration (property: value)
  background : pink ; → Declaration
}
  
```

↳ ELEMENT SELECTOR:

It is used to select an element based off the tag name. For example:

```
h2 {  
    color: blue;  
}
```

↳ Id SELECTOR:

It is used to select an element with a given id. For example:

```
# first {  
    color: white;  
    background: black;  
}
```

→ # is used to target by id.

↳ Class Selector:

It is used to select an element with a given class.

For example :

```
red {  
background: red;  
}
```

↳ * can be used as a universal selector to select all the elements.

```
* {  
margin: 0;  
padding: 0;  
}
```

↳ An inline style will override external and internal styles.

↳ COMMENTS IN CSS :

Comments in CSS is text which is not parse and is thus ignored.

CHAPTER :- 2

COLORS AND BACKGROUNDS

CSS rules are simple key - value pairs with a selector. We can write CSS rules to change color and set background.

↳ THE COLOR PROPERTY:

The CSS color property can be used to set the text color inside an element.

```
p {  
  color : red ; → Text color will be changed  
  } to red.
```

Similarly, we can set color for different elements.

↳ TYPES OF COLOR VALUES:

Following are the most commonly used color values in CSS

1. RGB → Specify color using Red, Green, Blue values
2. HEX Code → Specify color using Hex code.
 e.g: # FF 7180
3. HSL → Specify the color using HSL values
 e.g: hsl (8, 90%, 63%)
↓
Hue, Saturation, Lightness.

The value of the color or background color is provided as any one of these values.

↳ THE BACKGROUND - COLOR PROPERTY:

The CSS background-color property specifies the background color of a container.
 For example:

```

brown {
  background-color: brown;
}
  
```

can be other types of colors as well.

↳ THE BACKGROUND - IMAGE PROPERTY:

Used to set an image as the background.

```
body {  
  background-image: url("harry.jpg")  
}
```

The image is by default repeated in X and Y directions.

↳ THE BACKGROUND - REPEAT PROPERTY:

Can be any of

- Repeat - X → repeat in horizontal directions
- Repeat - Y → repeat in vertical directions
- No repeat → image not repeated.

See more possible values at MDN docs

↳ THE BACKGROUND - SIZE PROPERTY:

Can be following

- Cover → fits and no empty space remains.
- Contain → fits and image is fully visible.
- Auto → Display in original size.

- `{{Width}}` → Set width and height will be set automatically
- `{{Width}} {{Height}}` → Set width and height.

NOTE: Always check the MDN docs to direct a given CSS property. Remember, practice will make us perfect.

↳ THE BACKGROUND - POSITION PROPERTY :

Sets the starting position of a background image.

```
div 1 {
    background-position : left top ;
}
```

↳ THE BACKGROUND - ATTACHMENT PROPERTY :

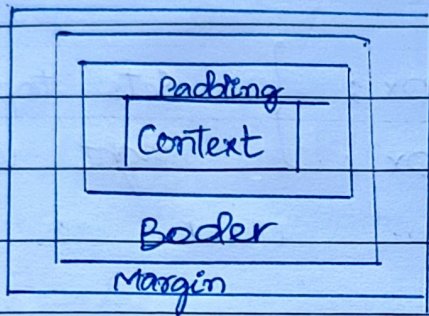
Defines a scrollable/non-scrollable characters of a background image.

```
div 2 {
    background-attachment : fixed ;
}
```

CHAPTER : 3

CSS Box Model

The CSS box model looks at all the HTML elements as boxes



↳ Setting width and height :

We can set width and height in CSS as follows:

```
# box {
  height : 70 px ;
  width  : 70 px ;
}
```

Note:- The total width/height is calculated as follows:

Total height = height + top/bottom padding + top/bottom border + top/bottom margin.

↳ Setting margin and padding:

We can set margin and padding as follows:

```

    • box {
      margin : 3px ;
      padding : 4px ;
    }
  
```

} Sets top, bottom, left, right values.

```

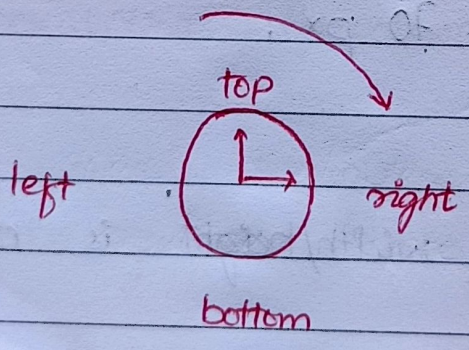
    • box main {
      margin : 7px 0px 2px 11px ;
    }
  
```

top right bottom left
 ↓ ↓ ↓ ↓

```

    • box last {
      margin : 7px 3px ;
    }
  
```

top and bottom left and right
 ↓ ↓



clock-wise

We can also set individual margins/paddings like this:

```
margin - top : 70 px
margin - bottom : 3 px
margin - left : 8 px
margin - right : 9 px
```

Same goes with padding.

↳ Setting Borders:

We can set the border as follows:

```
.bx {
  border - width : 2 px ;
  border - style : solid ;
  border - color : red ;
}
```

or just set border :
2px solid red ;
(short hand).

↳ Border radius :

We can set the border radius to create rounded borders

```
.div 2 {
  border - radius : 7 px ;
}
```

CHAPTER 1: 4

FONTS AND DISPLAY.

↳ The display property:

The CSS display property is used to determine whether an element is treated as a block/inline element and the layout used for its children.
flexbox/grid/etc.

• Display: inline

Takes only space required by the element. No linebreaks before and after. Setting width/height or margin/padding not allowed.

• Display: block

Takes full space available in width and leaves a newline before and after the element.

• Display: inline-block

Similar to inline but setting height, width, margin and padding is allowed. Element can sit next to each other.

• Display: None v/s Visibility: hidden

With display: none; the element is removed from the document flow. Its space is not blocked.

With visibility: hidden; the element is hidden but its space is reserved.

↳ Text - Align Property

Used to set the horizontal alignment of a text.

```

• div {
  text-align: center;
}

```

• Text - Decoration Property:

Used to decorate the text.

Can be underline, line-through, overline, none.

• Text - transform Property :

Used to specify uppercase and lowercase letters in a text.

```
p. uppercase {  
    text-transform : uppercase;  
}
```

↳ Line - height property :

Used to specify the space b/w lines

```
• Small {  
    line-height : 0.7;  
}
```

↳ FONT :

Font plays a very important role in the look and feel of a website.

↳ Font-family :

Font family specifies the font of a text.
Can hold multiple values as a
"fallback" system.

p {

font-family : "Times New Roman" , monospace ;
}



Always do this to ensure the correct
font of our choice is rendered.

CHAPTER : 5 SIZE, POSITION AND LISTS

There are more units for describing size other than 'px'. There are rem, cm, mm, vh, percentages, etc.

↳ What is wrong with pixels?

Pixels (px) are relative to the viewing device. For a device with size 1920 x 1080, 1 px is 1 unit out of 1080/1920.

↳ Relative lengths

These units are relative to the other length property. Following are some of the most commonly used relative lengths

em → unit relative to the parent font size
em means 'my parent element's font size'

rem → unit relative to the root font size
(<HTML > tag)

vw → unit relative to 1% viewport width.

vh → unit relative to 1% viewport height.

% → unit relative to the parent element.

↳ MIN/MAX - HEIGHT/WIDTH PROPERTY:

CSS has a min-height, max-height, min-width and max-width property.

If the content is smaller than the minimum height, minimum height will be applied.

Similar is the case with other related properties.

↳ The position property:

Used to manipulate the location of an element. Following are the possible values:

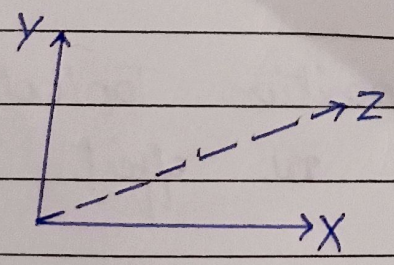
- Static: The default position. Top/bottom/left/right/z-index has no effect.

- **Relative**: The top/bottom/left/right/z-index will now work. Otherwise the element is in the flow of document like static.
- **Absolute**: The element is removed from the flow and is relatively positioned to its final non-static ancestor. Top/bottom etc works.
- **Fixed**: Just like absolute except the element is positioned relative to the browser window.
- **Sticky**: The element is positioned based on user's scroll position.

↳ Z-INDEX PROPERTY:

The Z-index property specifies the stack order of an element.

It defines which layer will be above which in case of overlapping elements.



→ Z is the third dimension.

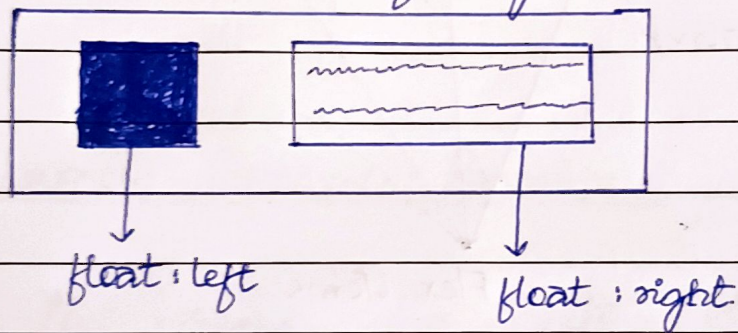
CHAPTER :- 6

FLEXBOX

Before we look into the CSS flexbox, we will look into float and clear properties.

↳ THE FLOAT PROPERTY :

Float property is simple. It just flows the element towards left/right.



↳ THE CLEAR PROPERTY :

Used to clear the float. It specifies what elements can float beside a given element.

↳ THE CSS FLEXBOX :

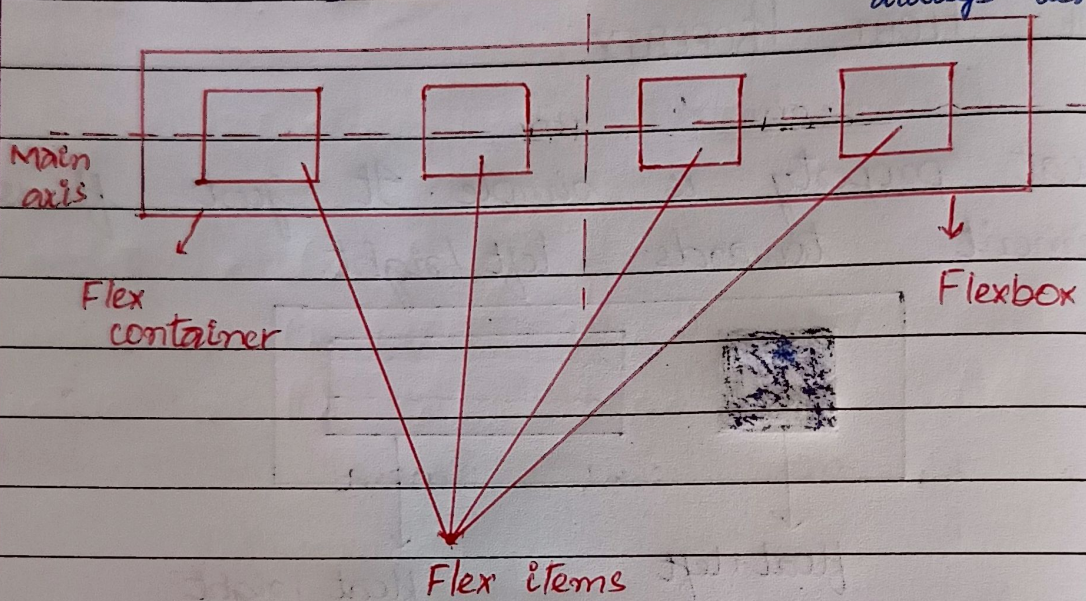
Aims at providing a better way to layout, align and distribute space

among items in a container.

• container {

display: flex;

→ Initialize a flexbox
 Cross axis → Needs not to be always vertical.



↳ CSS Media Queries

Used to apply CSS only when a certain condition is true

Syntax:

```
@media only screen and (max-width: 800px) {
  body {
    background: red;
  }
}
```

CHAPTER :

TRANSFORMS, TRANSITIONS AND ANIMATIONS

Transforms are used to rotate, move, skew or scale elements. They are used to create a 3-D effect.

↳ THE TRANSFORM PROPERTY :

Used to apply a 2-D or 3-D transformation to an element.

↳ THE TRANSFORM-ORIGIN PROPERTY :

Allows to change the position of transformed elements.

2D transforms → can change x and y axis.

3D transforms → can change z axis as well.

↳ CSS 2D TRANSFORM METHODS :

We can use the following 2-D transforms in CSS:

- translate ()
- rotate ()

- scale x ()
- scale y ()
- skew ()
- matrix ()
- scale ()

↳ CSS 3D TRANSFORM METHODS:

- rotate x ()
- rotate y ()
- rotate z ()

↳ CSS TRANSITIONS:

Used to change property values smoothly, over a given duration.

↳ THE TRANSITION PROPERTY:

The transition property is used to add transition in CSS.

Following are the properties used for CSS transition.

- transition-property → The property we want to transition.
- transition-duration → Time for which we want transition to apply.
- transition-timing-function → How we want the property to transition.
- transition-delay → Specifies the delay for the transition.

All these properties can be set using a single shorthand property.

transition: width 35 ease-in 25;

① property ② duration ③ timing-function ④ delay

↳ TRANSITIONING MULTIPLE PROPERTIES:

We can transition multiple properties as follows:

transition: opacity 1s, ease-out 1s, transform 2s ease-in;

↓

Yes we can skip transition delay here.

↳ CSS ANIMATIONS:

Used to animate CSS properties with more control.

We can use @keyframes rule to change the animation from a given style to a new style.

```
@keyframes Larry {  
  from {width: 20px;} → can change multiple  
  to   {width: 31px;} properties.  
}
```

↳ Properties to add Animations:

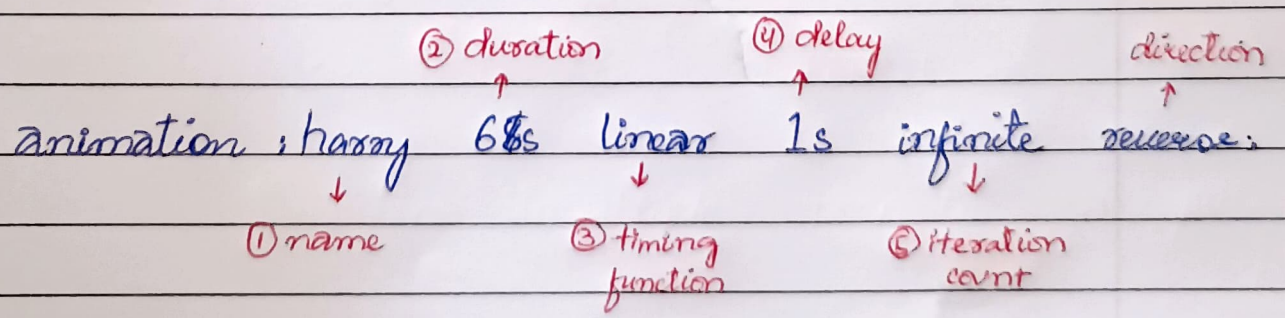
Following are the properties used to set animations in CSS:

- animation-name → name of the animation
- animation-duration → how long does the animation run?

- animation-timing-function → Determines speed curve of the animation.
- animation-delay → Delay for the start of an animation.
- animation-iteration-count → No. of times an animation should run.
- animation-direction → Specifies the direction of the animation.

↳ THE ANIMATION SHORTHAND:

All the animation properties from 1-6 can be applied like this:



↳ Using %age value states with animation.

We can use %age values to indicate what should happen when a certain percent of animation is completed.

```
@keyframes hurray {  
  0% {  
    width: 20px;  
  }  
  50% {  
    width: 80px;  
  }  
  100% {  
    width: 200px;  
  }  
}
```

⇒ can add as many intermediate properties as possible.

Java Script

It's a text based programming language long used for both on the client side and on the server side that allows us to make web pages interactive where HTML and CSS are languages that give structure and style to web pages interactive elements that engage a user. The common examples of JAVA Script that we might use everyday are a news recap video embedded on the New York Times or refreshing our twitter feeds. Incorporating JAVA Script improves the user experience of the web page by converting it from a static page into a dynamic page.

Q What is JAVA Script used for

It's mainly used for web based apps and web browsers but it's also used beyond the web in software servers and embedded hardware controls. Here are some basic things JAVA Script are used for:

1) Adding interactive behavior to web pages: JAVA Script allows users to interact with web pages. There are almost no limits to the things we can do with JAVA Script on a web page. The few examples are as given below:

- Show or hide more information with the click of a button
- Change the color of a button when mouse hovers over it
- Slide through carousel of images of the home page of an image

- playing a timer or count down on a web site.
- playing audio and video in a web page
- displaying animations

Software needed to run java script

We don't need any special software for java script because all main web browsers support it

How to use Java script

```
<html>
<head>
<title> welcome to java script
</title>
<script>
```

// write your java script code here

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

```
<body>
```

```
</script>
```

// write your java script code here

```
</script>
```

```
</body>
```

```
</html>
```

→ we can write <script> part either in the head tag or in body tag or both or both.

→ document.write is used to print the text

b = 20;
c = a + b

document.write("the sum is " + c);

```
</script>
</body>
</html>
```

→ Comparison of 2 no's

```
<html>
<head>
<title> Comparison </title>
</head>
<body>
<script>
var x;
    || y;

x = 100;
y = 20;
```

if (x > y)

document.write("the greater one is +x);

else

document.write("the greater one is +y);

```
</script>
</body>
</html>
```

→ Even odd

```
<html>
<head>
<title> even odd <title>
</head>
<body>
<script>
var num;
num = 9;
if (num % 2 == 0)
```

```
document.write (num + 'is even')  
else  
document.write (num + 'is odd')
```

```
</script >  
</body >  
</html >
```

=> loops

```
<html >  
<head >  
<title > loop in Java Script  
</title >  
</head >  
<body >  
  
<script >  
var i  
i = 1;  
while (i <= 10)
```

}

```
document.write ('welcome to  
Java Script <b>');  
++
```

}

```
</script >  
</body >  
</html >
```

Series (1 to 10)

```
<html >  
<head >  
<title series </title >  
</head >  
<body >  
  
<script >
```

```
var i;  
i = 1;  
while (i <= 10)
```

2

```
</script>  
</body>  
</html>
```

table generation

```
<html>  
<head>  
<title table <title >  
</head>  
<body>  
<script>  
var i;  
i = 1;  
while (i <= 10)
```

```
document.write ("2 x " + i  
+ " = " + 2 * i + "<br>");
```

it +

2

```
</script>  
</body>  
</html>
```

Pyramid of stars

```
<html>  
head  
title pyramid </title >  
</head >  
<body >  
<body >  
script  
var k;  
k = 1;  
while (k <= 5)
```

```
    k = 1;
```

while ($l < = e$)

```
{
  documents write ( + c );
  c + + ;
}
```

```
{
  document write ( " < br > br " );
  c + + ;
}
```

```
< / script >
< / body >
< / html >
```

1				
1	2	3		
1	2	3		
1	2	3	4	
1	2	3	4	5

Superintendent
AFPL (PVT.) ITI
Rambagh, Srinagar
Reg. No: PU0100010

checked & verified
Muneer Khan on 10/10/2022