HTML

Q→ What is Server ?

→ Server is a place where all the resources are present.

→ It accepts all upcoming request.

→ Compare to normal computer, it has higher configuration.

Ex :- Google, facebook, etc.

Q→ What is Protocol ?

- Protocol is set of rules defined for communication.
- Browsers understand only http/https.

Q→ What is HTTP ?

→ HTTP stands for Hyper Text Transfer Protocol.

→ It is a protocol used to access the data on the www.

→ The http protocol can be used to transfer the data in the form of plain text, hyper text, audio, video and so on.

→ HTTP is similar to FTP, as it also transfer the file from one host to another host, but HTTP is simpler than FTP, as HTTP uses only one connection.

Q→ What is HTTPS ?

→ HTTPS stands for Hyper text Transfer Protocol secure.

→ It is a secure extension or version of HTTP.

→ This protocol is mainly used for providing extra security to the data sent between website and the web brawser.

→ It is mainly used on the internet and used for secure communication.

→ Those websites which need login credentials should use HTTPS protocol for sending the data.

Page :
Date :

Q→ What is Web Browser ?
→ Web Browser is an application used to communicate with websites.
→ Web Browser understands only web languages
→ Different browser contain different compilers:-
Ex: Chrome, Mozilla firefox, etc.

Q→ What is Request ?
→ Request is a data exchange from browser (clients) to the websites (server)
→ Requests can be send in different ways :-
1) Typing URL
2) Clicking on Hyperlinks
3) Submitting Response.
→ Request can share user data.

Q→ What is Response ?
→ Response is a data exchange from website (server) to browser (client).
→ Response can be divided into two types
- Static Response : A static Response refers to a webpage that always displays the same content to every visitor.

- Dynamic Response : A dynamic response means the content of the page can change based on user-input, location, or other factors, essentially generating different content for each user depending on the situation.

Scanned with ACE Scanner

Q> What is Web Technology ?

Web technology refers to the way where two devices are communicating.

Q→What is URL ?

A URL (Uniform Resource locator) is a type of uniform resource identifier and is address of a resource on the 'world wide web' and protocol used to access it.

It is used to indicate the location of a web resource to access the web pages.

Q→What is WWW?

The world wide web is another way to describe the internet, which is a network of computers which are connected and that shares the information and communication around the world.

Q→ What are Webpages ?
A document which can be display in a web browser or structure by any markup language is nothing but the webpages.

Q→ What is website ?
A collection of webpages which are ~~required~~ grouped together and usually connected together in various ways.

Q→ What is Web-Server ?
A computer that hosts a website on the internet.

Q→ What is Search Engine ?
A web service that helps you to find other webpages, such as Google, Bing, Yahoo, etc...

Q→ What is Internet ?
Internet is a global network that connects billions of computers ~~access~~ across the world with each other. and to the world wide web.

Q→ What is Intranet ?
An Intranet is a private network of computers designed for certain group of people and owned by a particular firm or organisation.

Q-> What is Extranet ?

- Extranet is a part of an organisation's intranet.

- It is a communication network that is based on Internet Protocol (IP).

- It provides controlled access to firms intranet to its trading partners, customers and other business.

Working Principle of Web

- HTML is used to create the structure of webpage.
- CSS (Cascading Style Sheet) is used to make website attractive by giving some designs.
- JavaScript is used to make websites dynamic and interactive.
  JS is also used to add functionality in webpages.
- React JS is the library of javascript to make 'single page application'.
- Angular is also a Single Page Application.
- Library is defined as the collection of predefined codes.
- Framework is a collection of libraries.

⇒ Static Website: A website in which webpages are delivered exactly as they are stored, with no real time content changes.

⇒ Dynamic Website: It generates content in real time, typically using databases and scripting languages to provide interactively and personalized experienced.

Q-1 Difference b/w Static and Dynamic Website.

	Static Website	Dynamic Website
Content	Fixed content, manually updated.	Content changes automatically
Technology	Built with HTML, CSS; No Database Required	Uses Server Side language (PHP) & Database.
Speed	Faster; No processing required	Slower ; Requires server side processing.
flexibility	less flexible, manual updates.	Highly flexible, easy to update.
Cost/ Maintenance	Cheaper, simple to maintain	Costlier, more maintenance
Use Case	Small sites, fewer updates	Larger, interactive sites.

- Multiple Page Application are the application which operates within the multiple webpages.
- Single Page Application are the application which operates within the Single webpage.

Q→ Difference b/w SPAs and MPAs

	SPAs	MPAs
Performance	Faster loading Time	Slower loading Time
Debugging	More Difficult	Well supported by debugging tools.
Development	Fast	Slower, more complex
Maintenance	Fast and easy	Slower
Security	Simplified	More Challenging
SEO	Limited	Easier and more effective.
Cost	More Expensive	Less expensive
Scalability	Not Scalable	Scalable.

History of HTML

Tim Berners-Lee, a physicist at the CERN research institute in Switzerland invented HTML in 1991. The first version consisted of 18 HTML tags. Now, there are currently about 140 HTML tags, although not all of them are supported by modern browsers.

1991 - Tim Berners Lee invented HTML 1.0.

1993 - HTML 1.0 was released

1995 - HTML 2.0 was published

1997 - HTML 3.0 was invented

1999 - HTML 4.0 comes out. It was very successful.

2014 - HTML 5.0 was released and used worldwide.

* Structure of HTML :-

```
<!DOCTYPE Html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
              initial-scale=1.0">
        <title> Document </title>
</head>
<body>



</body>
</html>
```

* Introduction of HTML

- HTML stands for Hyper Text Markup Language.

- HTML is the standard markup language for creating web pages.

- HTML describes the structure of webpage.

- HTML elements tell the browser how to display the content.

- HTML uses .html extension.

- In HTML, a 'Boilerplate' refers to a standard template or set of code that provides a basic structure for a web page.

The World Wide Web Consortium (W3C) is a non-profit organization that develops technical guidelines and standards for the web.

- Declaration Statement declares the file belong to which version of HTML.

- DOCTYPE HTML → It will specify the browser, this document file belongs to version HTML5.

- DTD (Document Type Definition) → It is the declaration statement in HTML1 - HTML4.

* **Tags** : Anything which is enclosed between angular brackets (<>) is known as 'tag'.

There are two types of HTML tags:

1) **Pair Tags** : have an opening tag and a closing tag with content in between, such as <p> Content </p>. etc.

2) **Unpair Tag** : also known as "Self closing' or 'Void tags", do not enclose content and end with a slash, such as <img/>, <br/>, <input/> etc.

* **Element** : A tag with some content in it.

- <html></html> element is the root element in HTML.
- <head></head> The head element will hold the data of html document.
- <body></body> The structure shown on the webpage is written in body tag.
- <meta> meta tag will specify the metadata.

  -Metadata means the data about the data.

- Charset → It is used to specify the character set.

  Two types of Charset - (a) UTF-8 (for 8-bit)
  (b) UTF-16 (for 16-bit)

  UTF - Unicode Transformation Format.

Task : Find the font size of heading tags in 'rem' format.

<h1> </h1>	Bold	2.125 rem	34 px
<h2> </h2>	Bold	1.875 rem	30 px
<h3> </h3>	Bold	~~1.875~~ 1.5 rem	24 px
<h4> </h4>	Bold	1.25 rem	20 px
<h5> </h5>	Bold	1.125 rem	18 px
<h6> </h6>	Bold	1 rem	16 px

* Shortcuts

- h$ * 6 (h1 to h6) → Print all the headings.
  h$ * n (hn to hn)
- Shift + alt + ↓ / ↑ → copy the code or line where cursor is present.
- Ctrl + alt + ↓ / ↑ → Multi-line Cursor / Select multiple lines in element.
- tr * n <table> → The selected row will created n times.
- tr * n.classname → to give some classname to all row at once.
- tr * n.classname $ → It gives classname1, classname2, ----, classnameN etc.

HTML	XML
- Hyper Text Maskup language	- Extensible Maskup language.
- HTML is used to create structure of webpages.	- XML is used to transfer the data.
- HTML tags are predefined tags.	- XML tags are user-defined tags.
- In HTML, closing tags are not necesarly.	- In XML, closing tags are necessary.
- HTML used to displays the data.	- XML is used to store data
- HTML doesn't contain data it just displays it.	- XML carries data to and from the database.
- HTML is not case-sensitive	- XML is case sensitive.
- Extension : '.html' and '.htm'	- Extension : '.xml'.

* **Attribute :** Attribute gives extra information about the html element.

There are two types of HTML Attributes :

1. Global Attributes
2. Element Specific Attributes

1. **Global Attribute** : Global attributes are attributes that can be used with any HTML element. It is describe globally inside HTML code.

The core attribute of Global Attributes are : Id, class, Style, title

* **Id** : • id attribute is used to identify the html element uniquely.

• id should be unique for every html element.

* **class** : • we can target multiple elements using 'class'.

• we can use same class in multiple elements.

* **Style** : • style attribute is used to provide inline CSS.

* **title** : • The title attribute is used to specify extra info" about the element.

• When the mouse/cursor moves over the element then it shows the information.

2. **Element Specific Attribute** :

Element specific attribute are attributes which is described to find specific element inside the html code.

* Formatting Tags :

• To change the format of html elements the tags used are known as formatting Tags

•Formatting is a process that allows us to format text to increase its visual appeal.

Types of Formatting Tags :

- <b> Defines bold text.
- <strong> Defines important text
- <em> Defines emphasized text (italic).
- <i> Defines a part of text inside the html element is to displayed in italic.
- <sub> Defines subscripted text
- <sup> Defines superscripted text
- <ins> Defines inserted text (underline).
- <u> It is used to give underline the text inside the html element.
- <del> Defines deleted text
- <strike> It gives strike to text inside the html but it is depricated.

- <s>   used to give strike to the text.

- <pre>   whatever format you write it will appear in that format only.

* **Semantic Tags in HTML:**

Semantic tag will describle to the purpose to the browser as well as developer.

Types of Semantic Tags:

- <article>
- <aside>
- < details>
- < summary>
- < figure >
- < figcaption>
- < main>
- <header>
- <footer>
- <mark>
- <time>
- <nav>
- <section>

1. **<section>** - For large Container

Represents a standalone section of content that has its own theme or purpose. It often contains related content grouped together.

2. **<article>** - for large Container

Represents independent, self contained content. It can be distributed independently, like a blog, post, news article or forum post.

3. **<main>** - For Medium Container

Represents the main content of the document, encluding footer, header and navigational items.

4) <aside> - Also a container
It defines some content aside from the content
it is placed in (like a sidebar).

5) <details> - Information about something
creates a widget for showing or hiding additional
information.

6) <summary> - Dropdown button of details
provides a summary or title for the content of
<details> element. It creates a dropdown button.

7. <figure> - for media related content
used to group media content (images, diagram, etc.)
along with its caption.

8) <figcaption> - Caption of <figure>
provides a caption or description for the content
within the <figure> element.

9) <header> - Provides container at the top of webpage.
Represents the introductory content of a webpage.
It typically contains logo, navigations or headings.

10) <footer> - Provides a container at the bottow
It contains authorship info", copyright info",
contact info", sitemap, back to top links, related document,
etc

11). <nav> - Used for navigation bar
It is meant for sections of the website dedicated
to navigation like menus, home, or a list of internal
links.

12) <time> - Represent specific time/date/duration It allows for machine-readable time data and is helpful for calenders or schedulings.

13) <mark> - Used to highlight content.

- <div> - For Small container (Non Semeantic Tag) It serves as ~~an~~ generic container for grouping together other HTML element.

* LIST in HTML:

HTML list allow web developers to group a set of related items in a list.

There are 4 types of lists:

1. Ordered List
2. Unordered List
3. Nested List
4. Description / Definition List

1. Ordered List <ol>

These lists are used when the order of the items is important. Each item in an ordered list is typically marked with numbers or letters.

Attributes of <ol> :

1. type = "1", "a", "A", "i", "I".
2. start = accept only number.
3. reversed - can reverse sequence.

<li></li> - contains list items.

2. **Unordered list <ul>**

These lists are used for items that do not need to be in any specific order. The list items are typically marked with bullets.

Attribute of <ul> : type = "circle"; "disc"; "square".

- We cannot use Start and Reversed attribute in <ul>.

3. **Nested List :** List inside a list.

4. **Descripted / Description list <dl>**

These lists are used to contain terms and their corresponding descriptions.

Two tags used in <dl>

1. <dt> Definition Term
2. <dd> Description data

* Types of Elements in HTML :

1. **Block Level Element**
   - Block level Element will take complete viewport width.
   - We can modify the size of block level elements.
   - All the Block level elements will strat from new line.

e.g. : <div>, <section>, <main>, <article>, <header>, <footer>, <nav> <aside>, <table>, <ol>, <li>, <ul>, <dt>, <dd>. etc.

2. **Inline Elements:**

- Inline Element will take only content width.
- We can't modify the size of Inline Element
- Inline Elements will appear in the same line.

e.g : <span>, <a>, <b>, <sup>, <time>. etc.

3. **Inline - Block Element**

- Inline Block Element will take the content width.
- We can modify the size of Inline Block Element.
- If we use multiple Inline-Block Element, they will appear in same line.

e.g : All media tags.

✻ **Image Tag <img>**

The <img> tag is used to embed an image in an HTML page.

The <img> tag has two required attributes :

- **src** - specifies the path to the image.
- **alt** - Specifies an alternate text for the image, if the image for some reason cannot be displayed.

"./" → will show the src or source path current directory

"../" → will exit the current directory

* Anchor Tag <a>

The <a> defines a hyperlink, which is used to link from one page to another.

<a> element has an important attribute (href=" ").

(href=" ") - hyper-reference, which indicates the link's destination.

<a href="./" > </a> (Inline Level Element)

* CDN (Content Delivery Network)

- A network which is used to deliver the content.
- The group of multiple servers is known as CDN.

* Target Attribute

The target attribute specifies where to open the linked document.

Attribute Values :

_blank : Opens linked document in a new window or tab.

_self : Opens linked document in same frame as it was clicked

_parent : Opens linked document in parent frame.

_top : Opens the linked document in the full body of the window.

framename : Opens the linked document in the named iframe.

* Principles of HTML :

DRY - Do Not Repeat Yourself

SOC - Separation of Concerns.

* Table tag in HTML :

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

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in table.
<tr>	Defines a row in table
<td>	Defines a cell in table
<thead>	Groups the header content
<tbody>	Groups the body content

Page :
Date :

Attribute inside "table" tag :

- Colspan : It is used to merge two or more columns in a table.

- Rowspan : It is used to merge two or more rows in a table.

- Cell padding : Cell padding is the space between the cell edges and the cell content.
  By default, padding is set to '0'.

- Cell spacing : Cell spacing is the space between each cell.
  By default the space is set to 2 px.

- rowspan and colspan can be used only with <th> and <td>.

- cell spacing and cell padding is used with <table> tag.

*HTML Media Tags

<audio> - An inline element is used to embed sound files into a webpage.

<video> - used to embed video files into a webpage.

<source> - Used to attach multimedia files like audio, video etc.

<embed> - Used for embedding external applications, generally multimedia content like audio or video into HTML document.

<audio> and <video> have two attribute
loop : It is a boolean attribute
controls - "

<iframe> - It is used to display another document in HTML document.

- Media Tag is a group of tags that helps with creating multimedia experiences on a webpage.

- Media is used to specify that target URL

- Media tags have attributes like :
src, control, loop, autoplay, iframe

There are two types of path :

1. Relative Path : Relative path takes the reference from current directory. ('./', '../')

2. Absolute Path : Absolute path will always take reference from root directory i.e. 'C-drive'.

* FORM in HTML :

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

Syntax : <form action=" " method=" "> </form>

The default behaviour of form tag is refreshing.

Page
Date

Attributes of form

1. **Method Attribute** : Method is used to get the data and send the data (post) to the server.
The default value for method is "get".

Method attributes have some types -
- dialog
- get
- post

2. **Action Attribute** : Action attribute is used to store the data in a particular file.

* 22 types of input in form :

- button
- checkbox
- color
- date
- datetime-local
- email
- file
- hidden
- image
- month
- number
- password
- radio
- reset
- search
- submit
- tel
- time
- url
- week
- text
- range

The default value of the type = " " attribute is text.

Validation Attributes in form :

1. Required : It specifies that an input field must be filled out before submitting the form.

2. Disabled : It specifies that an input field should be disabled

3. Pattern : Specifies a regular expression that the input field value is checked against, when the form is submitted

4. Size : Specifies the visible width, in characters of an input field.

5. Step : Specifies the legal number intervals for an input field.

6. Maxlength : Specifies maximum number of characters allowed in an input field.

7. Min length : Specifies minimum number of characters allowed in an input field.

8. Min and Max :
9. Specifies the maximum and minimum values for an input field.

10. Readonly : Specifies that an input field is read-only.

1. Selected : Specifies a pre-selected option

2. Placeholder : Specifies a short-hint that describe expected-value.

CSS

Q→ What is CSS ?

- Cascading Style Sheet
- CSS describes how HTML element are to be displayed on screen.
- CSS is style sheet language we use to style an HTML document.
- External stylesheets are stored in CSS files.

* History of CSS :

- In 1994, Hakan Wium Lie, he was the person who introduce the idea of CSS to W3C.

- CSS was developed by World Wide Web Consortium (W3C), with significant contributions from "Hakon Wium Lie and Bert Bos.

CSS1 (First Version) in 1996
CSS2 (Second Version) in 1998
CSS3 (Third Version) in 2011

* Types of CSS :

1. Inline CSS
2. Internal or Embedded CSS
3. External CSS.

CSS Syntax : { selector, property, value
p {color : blue; } }

4 Ways to Add CSS

1. **Inline CSS** : achieve with the help of "style" attribute inside opening tag of element on which CSS has to be applied.

e.g :
`<p style="color:red"> Hello World! </p>`

2. **Internal or Embedded CSS** : applying CSS styling with the help of `<style></style>` tag, defines inside "head" tag.

e.g. :
```
<style>
        p {color: red;
          }
</style>
```

3. **External CSS** : achieved by linking an external file contain styling of webpage with the "link" tag having relation stylesheet.
   - the filename extension should be ".css".
   - External CSS is used to apply CSS on multiple pages.

Priority Order of CSS
- inline CSS has higher priority

Page :
Date :

* **Selectors** : Selectors are used to target an HTML element.

Types of Selector :-

1. Simple Selector
2. Combinator Selector
3. Attribute Selector
4. Pseudo class Selector
5. Pseudo Elements Selector

1. **Simple Selector** → • id selector
(select elements based on name, id, class)
- class selector
- universal selector
- element selector
- group selector.

i) **Id Selector** :-
- To target an element uniquely we use Id Selector.
- Unique Id attribute within the page is selected.
- core attribute selector.
- Selected by symbol '#' followed by id name.

Syntax :

```
#id_name {
        styling properties
      }
```

Note : An id name cannot start with a number.

(ii) **Class Selector :**

- The class selector elements, it selects HTML elements with a specific class attribute.
- To select element with specific class, period (.) character, followed by class name.

Syntax :

```
.center {
    CSS property ;
}
```

```
.class name {
    CSS property
}
```

Note : A class name cannot start with a number.

(iii) **Universal Selector (*):**

The universal selector selects all HTML elements on the page.

Syntax :

```
* {
   CSS property
}
```

(iv) **Element Selector :**

- The element selector selects all the ~~HTML~~ elements with the specified element name.
- Call by type of tag
- we can directly target the HTML element without prefix.

Syntax :

```
div {
   CSS property
}
```

```
element name {
    CSS property
}
```

(v) Grouping Selector :

• The group selector selects all the HTML elements with the same style definitions.

• Separate each selector with a comma.

```
e.g. →   h1, h2, p {
                 CSS Property;
             }
```

2. Combinator Selector

• A combinator defines the relationship b/w two selectors.

• A CSS selector can be simple or complex consisting of more than one selector connected using combinators.

* Descendant Combinator (space)
* Child " (>)
* Adjacent Sibling " (+)
* General " " (~)

(i) Descendant Selector (space)

• The descendant selector selects all the elements that are descendants of a specified element.

• These elements can be any level deep within the specified element.

• (parent, parent's parent, parent's parent's parent)

```
• Syntax :  selector1  selector2 {
                 Property Declaration;
            }
```

(ii) Child Selector (>)

- The child selector selects elements that are direct children of a specified element.
- Child selector is stricter than descendant selector, as it selects only the direct children.

- Syntax : selector1 > selector2 { properties;
}

(iii) Adjacent Sibling Selector (+)

- Adjacent sibling selector selects an element that is directly after another specific element.
- Sibling elements must have the same parent.
- This selects only the next sibling.

- Syntax : former_element + target_element
{ properties;
}

(iv) General Sibling Selector (~)

- The General Sibling Selector selects elements that follows a specified element and shares the same parent.
- This can be useful for selecting groups of elements with the same parent.

- Syntax: former_element ~ target_element
{ property;
}

3. Attribute Selector :

- The attribute selector is used to select elements with a specified attribute and value.

- Syntax : Selector[attribute

* Types of Attribute Selector

(i) [attr]
(ii) [attr = value]
(iii) [attr ~= value]
(iv) [attr |= value]
(v) [attr ^= value]
(vi) [attr $ ="value"]
(vii) [attr * ="value"]

4. Psuedo-classes :

- A pseudo-classes is used to define a special state of an element.

- For example, it can be used to :
  * Style an element when a user moves the mouse over it.
  * Style visited and unvisited link differently.
  * Style an element when it gets focus
  * Style valid/invalid/required/optional form elements.

- Syntax : selector: pseudo-class{

```
                         property: value;
                       }
```

✱ Dynamic pseudo-classes
- Link
- visited
- active
- focus
- hover

✱ UI element pseudo-classes
- enabled
- disabled
- checked

✱ Structural pseudo-classes
- first-child
- last-child
- Nth-child
- First-of-type
- Last-of-type

5. Pseudo-elements

- A CSS pseudo-elements is used to style specified parts of an elements.

- For example, it can be used for,

  - Style the first letter or line, of an element
  - Insert content before or after an element
  - Style the marker of a list
  - Style the viewbox behind a dialog box.

- Syntax : selector :: pseudo-element {
  property : value ;
  }

- :: first-line
- :: first-letter
- :: before
- :: after
- :: marker
- :: selection

⟹ What is !important ?

The !important rule in CSS is used to add more importance to a property/value than normal.

Like we use !important for a property, then it will override ALL previous styling rules, for that specific property on that element.

Priority Order of Simple Selector

ID > CLASS > ELEMENT > UNIVERSAL

With the help of !important rule, we can override the priority order.

* Text Property :

- In web design, CSS text properties can be used to decorate text.
- Web designers use these properties to transform plain text into amazing looking or simply it increases the visual appearance of texts.

- Text Formatting

- **Text Color** : The color property is used to set the color of the text.

The color are specified by -

i) predefined color names - (red, blue, black etc.)

ii) RGB (Red, Green, Blue) - rgb (val1, val2, val3)
0-255 , 0-255, 0-255

iii) RGBA (RGB Alpha) - rgb (val, val, val, α)
0-1

iv) hsl (hue, saturation, lightness) - hsl (h, s, l)
0-360 ↓ 0-100%
0-100%

hue → 0-360°, 0 = red, 120 = green, 240 - blue
saturation (0-100%) → 0% = shade of gray
100% = full color
lightness (0-100%) → 0% = black
100% = white
50% = neither black nor white

Page

Date

v) hsla(h,s,l,alpha) = hsl(val,val,val, $\underset{0-1}{\alpha}$)

α(0.0-1.0) → 0.0 (fully transperency)
1.0 (not transparent at all)

vi) Hexadecimal Value : #ffffff, #000000

- Text-Align : This property is used to set the horizontal alignment of a text.

A text can be left or right centred, centered, justified.

justify : each line is stretched sothat every line has equal width, and the left and right margins are straight.

- Text-Transform : This property is used to specify uppercase and lowercase letters in a text.

Three values : uppercase
lowercase
capitalize

- Text-Shadow : This property adds shadow to text.

It can be use in 4 ways → (a) X-axis
(b) Y-axis
(c) spreadness
(d) color

- **Text-Decoration** : This property is used to add a decoration line to text.

text-decoration property is a shorthand property for:

- text-decoration-line (required)
- text-decoration-color (optional)
- text-decoration-style (optional)
- text-decoration-thickness (optinal)

- **Letter-spacing** : used to provide space between letter.
- **Word-spacing** : used to provide space between words.
- **Text-Indent** : This property specifies the indentation of the first line in a text-block.

text-indent : length | initial | inherit

Note : "-ve" values are allowed, first line will be indented to left if value is '-ve'.

* Text-Effect

- **Text-overflow** : clip / ellipsis
- **word-wrap** : break-word
- **word-break** : Keepall / breakall
- **writing-mode** : horizontal-tb / vertical-rl

* Background Property :

The CSS background properties are used to add background effects for elements.

- Background-image : url ("image.jpg")
- Background-repeat : repeat-x/repeat-y/no-repeat
- Background-size : auto|length|cover|contain
- Background-position : top/right/left/bottom/top-right etc.
- Background-attachment : scoll/fixed

* Height & Width Property :

The CSS height & width property are used to set the height & width of an element.

Property	Descriptions
· height	- sets the height of an element
· width	- sets the width of an element
· min-height	- specifies minimum height of an element
· min-width	- specifies minimum width

- max-height - specifies maximum height
- max-width - specifies maximum width.

* Font-Property

The font property is a shorthand property for:

- font-style
- font-variant
- font-weight
- font-size
- font-family

- The font-size and font-family values are required.
- If one of other values is missing, their default value are used.

Property	Description
• font-style	Specifies font-style, default value = normal
• font-variant -	Specifies font-variant, default value = normal
• font-weight -	Specifies font-weight, default value = normal
• font-size/line-height -	Specifies font-size, default value = normal
• font-family -	Specifies font-family, default value = depend Browser

*line-height - specifies space b/w lines.

✳ **Background-Color** : • The background-color property sets the background color of an element.

• The background color property is specifies as a single <color> value.

Syntax : {background-color : "value" ;}

✳ **Background** : The Background property in CSS is very commonly used and contains many variants.

The background property consists of following properties:

- background-color
- background-image
- background-position
- background-size
- background-repeat
- background-origin
- background-clip
- background-attachment

CSS Background is a shorthand property of all above.

✳ **Gradients**

- CSS gradients let you display smooth transitions between two or more specified colors.
- CSS defines three types of Gradient :

i) **Linear Gradient** (goes down, up, left, right, diagonally)

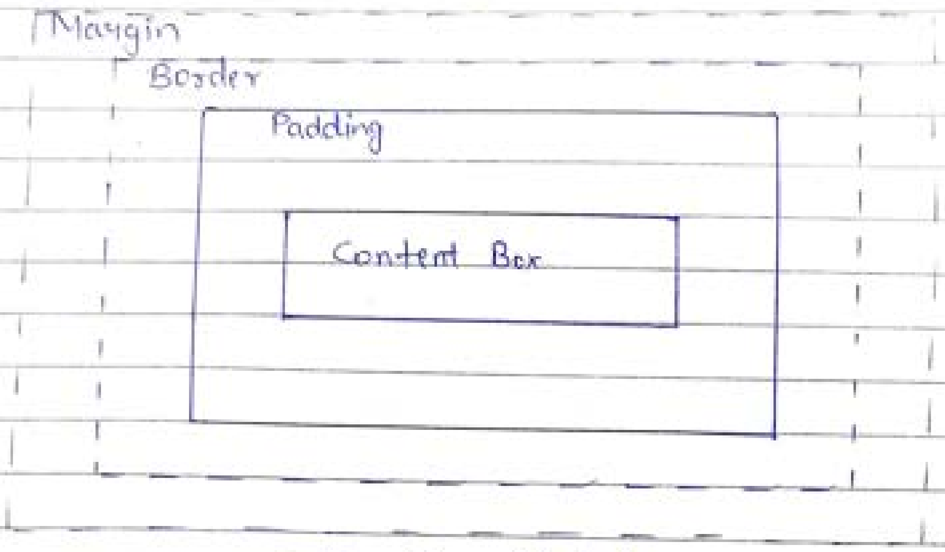
ii) **Radial Gradient** (defined by their center)

iii) **Conical Gradient** (rotated around a center point)

Scanned with ACE Scanner

* CSS Box-Model

The CSS Box-model is essentially a box that wraps around every HTML element.
It consists of content, padding, border, margin



The Box Model

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent. It is the inner space of box.
- **Border** - A border that goes around the padding and content.
- **Margin** - Clears an area outside the border. The margin is transparent. It is the outer space of box.

The box-model allows us to add a border around elements, and to define space between elements.

padding : Apx Bpx Cpx Dpx ;
top right bottom left

" : Apx Bpx Cpx ;
top left-right bottom

" : Apx Bpx ;
top-bottom right-left

* We cannot pass negative (-ve) value in padding

⇒ Border Shorthand

border : size style color ;

⇒ Margin shorthand same as padding, but in margin we can pass (-ve) negative value.

* CSS positions

The CSS Position property specifies the type of positioning method used for an element.

There are 5 CSS Positions :

- Static Position
- Relative Position
- Absolute Position
- Fixed Position
- Sticky Position

padding : Apx Bpx Cpx Dpx ;
top right bottom left

" : Apx Bpx Cpx ;
top left-right bottom

" : Apx Bpx ;
top-bottom right-left

* We cannot pass negative (-ve) value in padding

⇒ **Border Shorthand**

border : size style color ;

⇒ Margin shorthand same as padding, but in margin we can pass (-ve) negative value.

* CSS positions

The CSS Position property specifies the type of positioning method used for an element.

There are 5 CSS Positions :

- Static Position
- Relative Position
- Absolute Position
- Fixed Position
- Sticky Position

1. Static Position

- The default and initial position of an HTML element is a Static Position

- Static positioned element are not affected by top, bottom, right, left properties.

2. Relative Position

A position which is taking reference from its normal position and changing its positions by top, bottom etc. properties.

3. Absolute Positions

- Position Absolute will take reference from its nearest parent, which is having position: relative.
- If the nearest parent isn't position: relative, then it will check for a level-up parent to be positioned relative, and if not, then ultimately it will take the reference from the body.

4. Fixed Position :

- It will fixed the element to the viewport.

Page :
Date :

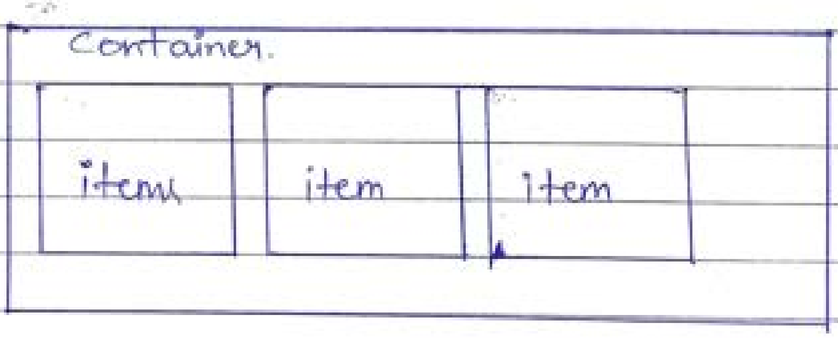
5. Stick Position

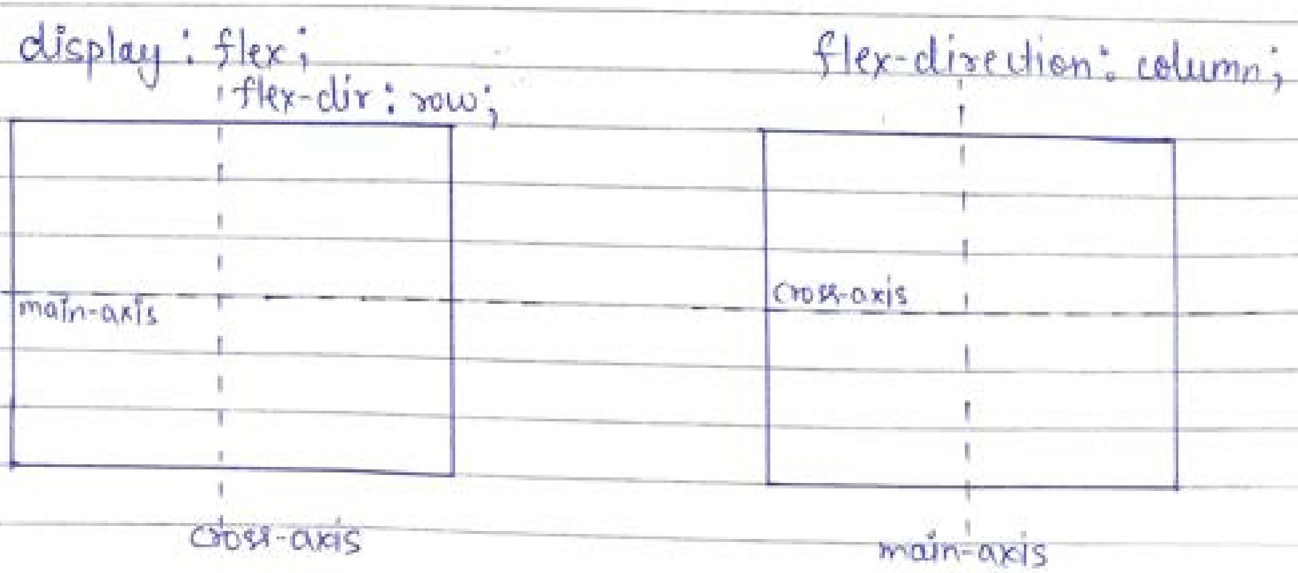
- It is scrollable upto certain point. Once, it reach to that point, it will get position fixed.

✱ CSS Flexbox

Flexbox is a layout method used for arranging items in rows or columns.
It is 1-directional layout
Flexbox consist of :

- a flex container - the parent element
- Flex items - the items/element inside parent element





- Display : flex / inline-flex

The flex container becomes flexible by setting the display property to flex.

* CSS propesties we use for flex-container are:

- flex-direction : this property specifies the display-direction of flex items in the container

values : row / column / row-reverse / column-reverse

- flex-wrap : this property specifies whether the flex-items should wrap or not, if there is not enough room for them on one flex line.

values : nowrap / wrap / wrap-reverse.

- flex-flow : shorthand property for flex-direction and flex-wrap.

- justify-content : this property is used to align items on the main-axis

values : center / ~~left / ri~~ flex-start / flex-end / space-between / space-around / space-evenly

- align-items : this property is used to align flex-items on the cross-axis.

values : center / flex-start / flex-end stretch / baseline / normal

- **align-content** : this property is used to align the flex-lines.

values : center | stretch | flex-start | flex-end
space-around | space-evenly | space-between

*** CSS properties we use for flex-items :**

- **order** : specifies the order of the flex items inside the flex container.

The order value must be number, default is 0.

- **flex-grow** : specifies how much a flex items will grow relative to the rest items

Value must be a number, default value is 0.

- **flex-shrink** : specifies how much a flex item will shrink relative to the rest of flex-items.

Value must be a number, default value is 1.

- **flex-basis** : specifies the initial length of flex-item.

- **flex** : shorthand property for the flex-grow, flex-shrink, and flex-basis

e.g. flex : 0 0 200px ;
growable(0) , shrinkable (0), initial-length (200px)

- align-self : specifies the alignment for the selected item inside the flexible container.

align-self override the default alignment set by the ~~align~~ container's align-items property.

⚘ Grid Layout

Grid layout Module offers a grid-based layout system, with rows and columns
It is a 2-directional layout.

Grid Container : A grid layout consist of parent element, (the grid container), with one or more child elements.

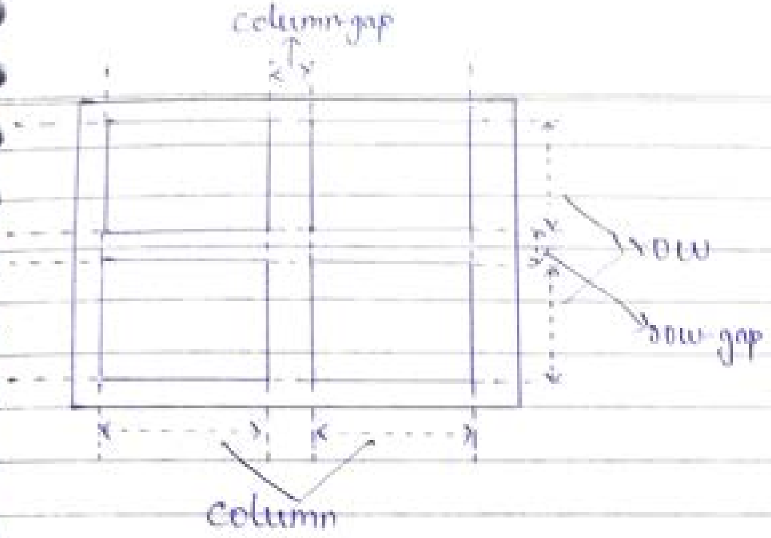
Grid-items : All direct children of grid container automatically become grid-items.

- Display : grid / inline-grid

An HTML element becomes a grid container when its display property is set to "grid" or "inline-grid".

- Grid-Column : The vertical lines of grid-items
- Grid-row : The horizontal lines of grid items.
- Grid-gap : The spaces between each column/row is grid-gap.

Scanned with ACE Scanner



- column-gap : specifies gap b/w columns in a grid.
- row-gap : specifies gap b/w rows in a grid.
- gap : Shorthand property for row-gap and column-gap.

The rows and columns of a grid is defined with the "grid-template-rows" and the "grid-template-columns" properties.

- grid-template-columns : defines number of columns in grid layout, and it can define the width of each column.

grid-template-column : auto auto auto ;

three values means three columns, value is space-separated-list, where each value defines the width of respective column.

Note: If grid items are more than number of column, then grid will automatically add a new row to put in.

Scanned with ACE Scanner

- grid-template-rows : defines the height of ~~[illegible]~~ each row.

grid-template-rows : 80px 200px;

value is space-separated, where each value defines height of respective ~~columns~~ rows.

- justify-content : used to align the whole grid inside the container.

values : start / end / center / space-around space-between / space-evenly

- align-content : used to vertically align the whole grid inside the container.

values : same as justify-content.

- place-content : shorthand property for "justify-content" and "align-content".

✱ Grid items : A grid container contains grid-items.

- grid-column : this property defines on which column(s) to place an item.

It is a shorthand property for "grid-column-start" and "grid-column-end" properties.

✱ grid-column : 1/5;

✱ grid-column : 1 / span4;

Page:
Date:

* grid-row : defines on which row to place an item

  - shorthand property for "grid-row-start" and "grid-row-end".

  * grid-row : 1/4 ;
    grid-row : 1 / span 2 ;

* Grid-area : is a shorthand property for "g-r-st", "g-c-st", "g-r-end", and "g-c-end" properties.

  * grid-area : 1/2/3/5 ;

make an "item" start on row-line 1 and column-line 2 and end on row-line 3 and column-line 5.

  * grid-area : 1/2/ span 3 / span 2 ;

make an "item" start on row-line 1 and column-line 2 and span 2 rows and 3 columns.

⇒ grid area property can also be used to assign names to grid-items.

```
        .item1{
                grid-area : A ;
              }

        .grid-container{
                grid-template-areas:' A  A  A  A ';
                }
```

- Naming grid items can be referred to by the "grid-template-areas" property of the grid container.
- Each row is defined by apostrophes (' ').
- columns defined inside apostrophes, separated by space.

✱ CSS Animations :

- An animation lik an element gradually change from one style to another.
- To use CSS animation, we must first specify some key-frames for the animation.
- Keyframes hold what styles the element will have at certain times.

✱ @Keyframe Rule :

- When we specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

- To get an animation to work, we must bind the animation to the element.

⇒
```
div {
      width : 100px ;
      height : 100 px ;
      background-color : red ;
      animation-name : example ;
      animation-duration : 4s ;
     }

@keyframes example {
            from { background-color : red:}
             to  { background-color : green;}
```

Page :
Date :

Required animation properties for CSS ~~wo~~ Animation works:

① animation-name : name of element for animation.
② animation-duration: specify duration of animation.

*CSS Animation Properties:

*@Keyframes : specifies animation code

*animation-name : specifies the name of the @keyframes animation

*animation-duration : Specifies how long time an animation should take to complete one cycle.

*animation-timing-function : Specifies the speed curve of the animation.

values of animation-timing-functions:

+ease : animation with a slow-start, then fast, then end slowly (this is default)

+linear : animation with same speed from start to end.

+ ease-in: animation with slow-start

+ ease-out : animation with slow-end

+ ease-in-out : animation with a slow-start and end.

Scanned with ACE Scanner

Page :
Date :

* **animation-delay** : specifies a delay for the start of an animation.

-negative values also allowed, If negative value used, the animation will start as if it had already been playing for N seconds.

* **animation-iteration-count** : specifies the number of times an animation should run.

-infinite value make the animation continue for ever.

* **animation-direction** : specifies whether an animation should be played forwards, backwards or in alternate cycles.

-values of animation-direction :

- **normal** : animation played as normal (forwards) this is default
- **reverse** : animation played in reverse-direction (backward)
- **alternate** : animation played forward firste, then backwards.
- **alternate-reverse** : played backward first, then forward.

* **animation** : is a shorthand property for

- animation name
- animation duration
- animation-timing function
- animation- delay
- animation-iteration-count
- animation-direction

* animation : example 4s linear 2s infinite normal ;

* **Box-Sizing** : This property allows us to include the padding and border in an element's total width and height.

- **box-sizing : border-box ;**

* **Z-index** : • z-index property specifies the stack order of an element.

  • when elements are positioned, they can overlap other elements, z-index specifies which element should be placed in front of, or ~~ot~~behind the others.

• An element can have a positive or negative stack order.

• z-index only works on positioned elements. and flex-items

Note : If two positioned elements overlap each other without z-index specified, the element defined last in the HTML code will be shown on top.

CSS Box Shadow :

- Box-shadow property is used to apply one or more shadows to an element.
- In simplest use, we can only specify a horizontal and a vertical shadow.
- The default color of the shadow is the current text-color.

```
div {
    box-shadow : 10px 10px ;
   }
```

- box-shadow has some parameters :

(i) x-axis → shadow along horizontal
(ii) y-axis → shadow along vertical
(iii) blur → defines blur radius
(iv) spread → defines spread radius (+ve value inc. shadow size
-ve value dec. " ")
(v) color → color of shadow of element
(vi) inset → changes shadow from outer shadow (outset) to an inner shadow (inset).

An element can have multiple shadow.

CSS 2D Transforms

- CSS transforms allow us to move, rotate, scale and skew elements.

Page :
Date :

* CSS 2D Transform functions :

- translate () : moves an element from its current position (according to x-axis and y-axis)

- rotate () : rotates an element clockwise or counter-clockwise (according to given degree)
  - -ve degree rotate counter-clockwise.

- scale () : increases or decreases the size of an element (according to parameters width and height)

* transform : scale (2, 3) ;
  increase with 2 times, and height 3 times.

- scale X () : increase or decrease width of element.
- scale Y () : increase or decrease height of element.

- skew() : skews or twist an element along X and Y-axis by given angles

  trasform : skew (20deg, 30deg);

- skew X () : skew element along X-axis
- skew Y () : skew element along Y-axis

* CSS 3D Transform functions :

- rotateX() : rotates an element around its X-axis at a given degree.
- rotateY() : rotate an element ~~ato~~ around its Y-axis at a given degree
- rotate Z() : rotate an element around its Z-axis at a given degree.

* CSS Transitions :

- CSS Transitions allows you to change property values smoothly, over a given duration.

  - transitions-property
  - transition-duration
  - transition-timing-function
  - transition- delay

- transition : shorthand property for all of above.