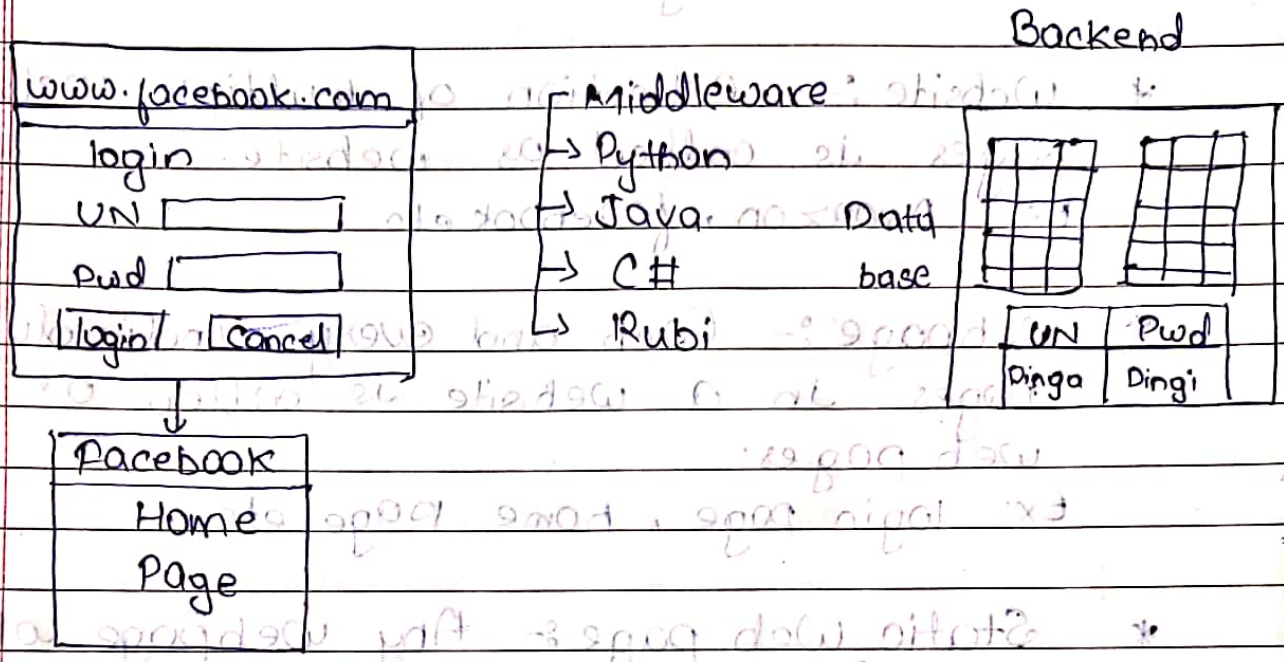
Date __/__/____
Page _____

Web Technology-



Front end → html
→ CSS
→ Javascript

* Software Defination :- Collection of multiple programs which develops to solve the realtime problems is called as software.

* Browser Defination :- Browser is a software which is used to fetch the information from world wide web according to user need.

Date __/__/__
Page _____

* Internet :- Internet is vast network which is used to connect multiple devices through out the world.

* Website :- Collection of multiple web pages is called as website.
Ex:- Amazon, facebook etc.

* Webpage :- Each and every individual pages in a website is called as web pages.
Ex:- login page, home page etc.

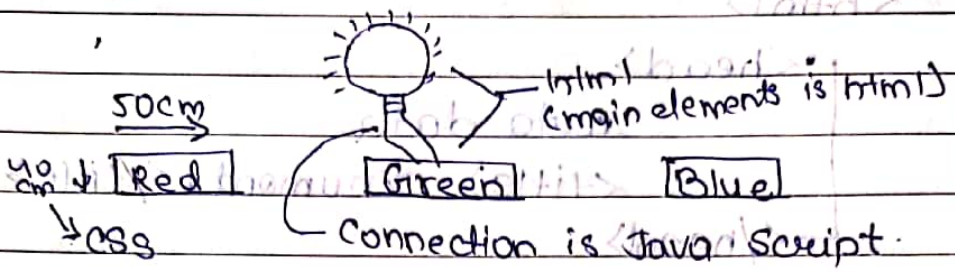
* Static Web page :- Any webpage which will display some content for different different user. Ex:- Help page, terms & conditions etc.

* Dynamic web page :- Any webpage which will displaying different content for different different users.
Ex:- facebook home page, WP home page.

* Server :- Server is combination of hardware and software which is used to store some data.

* Web Technology :- Web Technology refers to various tools and technique which is used to develop web page.

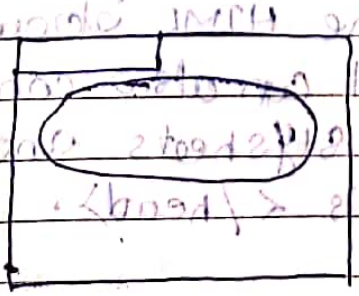
Date __/__/____
Page _____



- **HTML** :- HTML stands for Hyper text markup
  - Hyper text → Hyper link → we can Dev
  - Markup → an built tag → using
  - "<>" Angular braces
  - Language → Connection → b/w users & browsers

i) It is used to develop web pages
ii) It is a simple language where we have to use the tags and execute the code in one of the browser.
iii) Markup means that with HTML you declare what is presented to a viewer, not how it is presented.

- **Tag** :- Tag is a pre-defined word in HTML which is used to develop individual elements.



Date / /
Page

```
<html>
    <head>
        meta data
        <title> document 1 </title>
    </head>
    <body>
        <h1> ABCD </h1>
        <p> My name is Ankit </p>
    </body>
</html>
```

- <!doctype>:- Defines the HTML version used in the document. In this case it is HTML 5.

- <html> :- Opens the Page. No markup should come after the closing tag </html>. The language attribute (<html lang="en">) declares the primary language of the page, using the ISO language codes (en for english).

- <head> :- Opens the head section, which does not appear in the main browser window but mainly contains information about the HTML document, called metadata. It can also contain imports from external stylesheets and scripts. closing tag is </head>.

Date __/__/____
Page _____

- <meta> :- Gives the browser some metadata about the document. The charset attribute declares the character encoding. Modern HTML documents should always use UTF-8, even though it is not a requirement. In HTML, the <meta> tag does not require a closing tag.
- <title> :- The title of the page. Text written between this opening and closing tag (</title>) will be displayed on the tab of the page or in the title bar of the browser.
- <body> :- Opens the part of the document displayed to users, i.e. all the visible or audible content of a page. No content should be added after the closing tag </body>.
- <h1> :- A level 1 heading for the page.
- <p> :- Represents a common paragraph of text.

* Types of Tag :-

- Paired Tag :- Any tag which is used by opening and mandatory to close this type of tags called paired tags. Ex:- <html></html>

Date __/__/__
Page _____

* Unpaired Tag :- Any tag which can be used without closing this type of tag called unpaired tags.
Ex:-
, <hr>.

* Formatting Tag :-

① <mark> :- It is used to mark or highlight text in a document. [in yellow colour].

② <b> :- It is used to bold the text in a document.

③ <i> :- It is used to italic the text in a document.

④ <u> :- It is used to underline the text in a document.

⑤ <strike> or <s> :- It is used to make a cut line in a text like ~~999~~.

⑥ <small> :- It is use to make the text small in size.

⑦ <big> :- It is use to make the text big in size.

⑧ <sup> :- The <sup> tag defines superscript text. Superscript text appears half a character above the normal line and is sometimes rendered in a smaller font. Superscript text can be used for footnotes like WWW. Ex: My name is Ankit.

⑨ <sub> :- The <sub> tag defines subscript text. Subscript text appears half a character below the normal line, and is

Date __/__/____
Page _____

Sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H_2O.
Ex:- My Name is Ankit.

- **Heading Tag :-**

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

From h1 to h6 the size and boldness is decrease.

- Task :- ABCD $A^2 + B_2 + C^{2^2} + 3AB_2$

Body code :-

```
<p> ABCD A <sup>2</sup> + B <sub> 2 </sub> +
O <sup>2 <sup> 2 </sup> </sup> + 3AB <sub>
</sub> </p>
```

* **HTML Elements :-** HTML element is Nothing but start of the tag to end of the tag.

html element

ex:- `<h1 id="a1"> Hello </h1>`

tag — attribute — content — closing tag

Date / /
Page

* **HTML Attributes :-**

1) Attributes are used to provide the extra information of that particular tag.

2) Always attributes should be declared within open tag.

3) Each and Every tag can have attributes.

4) One tag can have multiple attributes.

5) Always attributes comes as pair.

```
<html>
  <body bgcolor = "green">
         ↓             ↓
   Attribute Name   Attribute Value
          \____________/
            Attribute
  </body>
</html>
```

* **HTML List :-**

HTML list is used to group the similar kinds of data.

• <ol start = "30"> :- starts from 30 in web page
• Reversed :- For Reverse order.

In list we have three types:-
1) Ordered list
2) Unordered list
3) Description list

1) Ordered list :- When we want to group ordered type of data then have to use ordered list. An ordered list can be created with the <ol> tag and each list item can be created with the <li> tag as in the example below:-

```
<ol>
<li> Item </li>
<li> Another Item </li>
<li> Yet Another Item </li>
</ol>
```

Output :-
1. Item
2. Another Item
3. Yet Another Item

2) Unordered list :- When we want to group unordered type of data we use unordered list. An unordered list can be created with the <ul> tag and each list item can be created with the <li> tag as shown by the example below :-

Date / / Page

```
<ul>
  <li> Item </li>
  <li> Another Item </li>
  <li> Yet Another Item </li>
</ul>
```

Output:-
- Item
- Another Item
- Yet Another Item

3) Description List :- When we want to describe a term we use description list. A description list (or definition list) can be created with the dl element. It consists of name-value groups, where the name is given in the dt element, and the value is given in the dd element.

i) <dl> </dl> → Description list
ii) <dt> </dt> → Description term
iii) <dd> </dd> → Description data

Ex:-

```
<dl>
  <dt> Name 1 </dt>
    <dd> Value for 1 </dd>
  <dt> Name 2 </dt>
    <dd> Value for 2 </dd>
</dl>
```

Date __/__/____
Page _____

Output :-

Name 1
Value for 1
Name 2
Value for 2

* Container tags / Grouping Elements :-

In HTML we have two container tags :-

i) // block element
ii) // inline element

i) :- • Div is block level element which is used to store some elements.

- It will occupy entire width of the screen.
- We can specify height and width for div.
- If we use multiple div the output will be displayed in the next line for each div.

ii) :- • Span is a inline element which is used to store elements.

- It will occupy content size.
- We cannot specify height and width for span.
- If we use multiple span the output will be displayed in the same line.

Date __/__/____
Page _____

Tables

- Tables are used to insert the data in table format.

- To Develop a table we have to use following tags :-

i) <table> :- Used to develop a table
ii) <tr> :- Table row is used to develop single row
iii) <td> :- Table data is used to insert data in row develop each cell.

Attributes :-

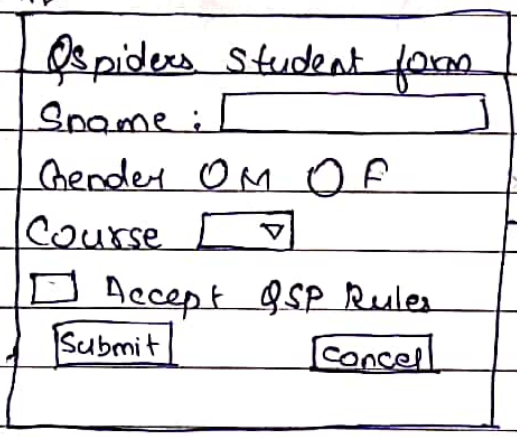
i) border :- Border is used to set the border.
ii) Cellspacing :- It is used to specify the spaces between two cells.
iii) Cellpadding :- It is used to specify the space between text and border of the cell.
iv) <th> :- It is used to develop table headings.
iv) Colspan :- It is used to merge 2 or more colums.
v) rowspan :- It is used to merge 2 or more rows.

Date __/__/____
Page _____

* Forms :-

- HTML forms is used to collect the user inputs.
- HTML form consist of the elements which develop textfields, checkbox, radio button, buttons etc.

```
<form>
```



→ Qspiders form

```
</form>
```

* Form tags and Form elements :-

1) <input>
2) <label>
3) <select>
4) <textarea>
5) <button>
6) <fieldset>
7) <legend>
8) <option>

Date / /
Page

* Form Input types :-

1. Datetime-local
2. Email
3. File
4. Hidden
5. Range
6. Month
7. Number
8. Password
9. Radio
10. Image
11. Checkbox
12. Reset
13. Search
14. Submit
15. Tel
16. Text
17. Time
18. Url
19. Week
20. Button
21. Date

* id Attribute :-

- id="" :- It is an attribute which is used to give a unique id for HTML elements.

Date ___/___/_____
Page _____

=) Every tags or elements can have id.

=) Always id value should be unique

* Name = " " :-

- It is an attribute which is used to give name for the user entered data.
- It is also responsible for passing the data from the form.

* <select>

- It is a form tag which is used to develop dropdown.

* <option>

- It is a form tag which is used to develop options in a dropdown.

Ex:- <label for="course"> id = "course"> Course </label>
<select name="course" id="course">
<option value="btech"> B.Tech </option>
<option value="ba"> BA </option>
<option value="mca"> MCA </option>
<option value="msc"> M.sc </option>
<option value="bca"> BCA </option>
</select>

Date __/__/____
Page _____

* <textarea>
- It is a form tag which is used to text area fields.

* <button>
- It is a form element which is used to develop a button.

* <fieldset>
- It is a form tag which is used to develop a field around the form.

* <legend>
- It is a form tag used to specify the name of the form.

* Validation Attributes :- Validation attributes are used to perform the validation on form.

i) Required :- It is validation attribute on form to specify the mandatory fields which is denoted by *.

ii) No-validate :- It is validation attribute which should be used in form tag if we use no attribute all the validation attributes applied for the form will not perform any validation.

Date __/__/____
Page _____

iii) Disabled:- It is a validation attribute which is used to disabled an element.

iv) Read Only :- It is a validation attribute, if we use read only attribute in any element initial value will be the final.
Value :- It is used to give initial values to any field.

v) Placeholder :- It is a validation attribute which is used to give the place-holder for the field.

vi) Pattern :- It is a validation attribute where we can validate by our own pattern.
Pattern = "[a-z A-Z 0-9 @ # $] {4, 10}"
Min Max

vii) Min-length :- It is validation attribute which is used to set the minimum length.

viii) Max-length :- It is validation attribute which is used to set the maximum length.

ix) Autofocus :- It is validation attribute which is used to focus on perticular element by default.
It is used only one time inside a form.

x) Autocomplete :- It is validation attributes which is used to unable and disable the autocomplete. We can use the values 'on' and 'off' for autocomplete.

Date / /
Page

* How to use image in webpage :-

- We can develop images in our webpage by using img tag.
  <img src=" " alt=" ">

* src=" " :- It is an attribute to specify the source file.

* Marque :- It is used to slide some element, it is paired tag.
  <marque behaviour="scroll"></marque>

- behaviour=" " :- It is used to specify behaviour of marque

- direction=" " :- It is used to specify the direction of the marque.

* How to use audio, video in ~~vid~~ webpage

- <video> :- To develop the videos in web pages, we can use video tag.

- Attributes used in video tag :-

i) src=" ";
ii) height=" ";
iii) width=" ";
iv) controls
v) Autoplay
vi) Muted

Date __/__/____
Page _____

* Anchor Tag :-

- Anchor tag is used to develop hyperlinks.
- In real-time projects we use anchor tag to link from one web page to another web page.

* href attribute :-

- It stands for hyper reference.
- It is used to specify the hyper reference or url.

* target attribute :-

- It is used to specify how the linked web page should be displayed.
- Values :-

  _blank, _parent, _self, _top

* How to use audio and video in web page :-

- Video :-
  To develop videos in web page we use <video> tag

Date / /
Page

Attributes used in <video> :-

controls :- To show controls in video.

autoplay: video should start automatically

muted : video should be muted.

<summary>

Syntax:

```
<details>
  <summary> Read more </summary>
  <p> ABC
    -
    -
    -
  </p>
</details>
```

- Symmantic tag are available only in HTML5.

- To develop symentic tag we use <summary> tag.

* <iframe>
- It is used to develop a window inside another window. Usually iframe are used to develop ads.

Date __/__/____
Page _____

CSS

- Css stands for Cascading Style sheets.
- Css describe how html elements should be displayed in the browser.
- External css file should saved in .css file format.

* We can use css in three ways:-

i) Inline CSS
ii) Internal CSS
iii) External CSS

i) Inline CSS :- We use CSS within the tag by using style attribute it is nothing but inline css.
- The priority of inline CSS is one (first priority).

ii) Internal CSS :- We use CSS internally in the html file by using style tag within the head.
- The priority of internal CSS is two (second priority).

iii) External CSS :- We write CSS code in external file with .css file format, by linking that file to the html file by using link tag within the head.
- The priority of external CSS is three (3rd priority).

Date / /
Page

* External CSS or Internal CSS syntax :-

```
Selector {
        property : value;
        }
```

ex :-

```
h1 {
color : green;
background-color : blue;
}
```

Selector — h1
Property — color, background-color
value — green, blue

* Selectors :- In CSS before giving any style to the web page we have to select the element first by using selectors.

or

Selector are used to select the element

- Simple Selector :-

a) Universal selector (*) :- Universal selector denoted by symbol (*) it selects all the elements. * { ... }

b) Element selector (tag name) :- Element selector will be denoted by tag name, It can select multiple element with the matching tag name. eg :- h1 { ... }

Date __/__/____
Page _____

c) Id selector(#) :- It selector is denoted by the symbol (#) it will select an element based on id value.
eg:- #a2 { ---- }

d) Class selector (.) :- Class selector is denoted by the symbol (.) it select multiple element ele based on class value.
eg:- .c1 { ---- }

2) Grouping Selector :- In grouping selector we can use multiple tag name for one style.
eg:- h1, span, div {
color: red;
}

3) CSS Combinator :- A combinator is something that explains the relationship between the selectors.

a) descendant selector (space) :- The descendant selector matches all elements that are descendents of a specified element.
ex:- div p
{
color: red;
}

Date / /
Page

b) Child Selector (>) :- The child selector selects all the elements that are the children of a specified element

eg:-

```
div > p {
    color: green;
}
```

The Selects all the <p> elements that are children of a element.

c) Adjacent Sibling Selector (+) :- The adjacent sibling selector is used to select an element that is directly after another specific element.

- Sibling elements must have the same parent element, and "adjacent" means "immediately following".

ex:-

```
div + p {
    color: blue;
}
```

d) General sibling selector (~) :- The general sibling selector selects all elements that are next siblings of a specified elements.

ex:-

```
div ~ p {
    color: brown;
}
```

Date __/__/____
Page _____

* Pseudo Class selector :-

- Pseudo class selector are used to style visited, unvisited & active links as well as it can handle mouse hovering actions.
- PSeudo class selector are denoted by the symbol (:).
- Unvisited link (link)

=> It is used to style an unvisited link.

```
ex:- a:link {
        color: red;
     }
```

- Visited link (visited)

=> It is used to style a visited link.

```
ex:- a:visited {
        color: blue;
     }
```

- Active link (active)

=> It is used to style a active link. (Active is nothing but when the user click on it.)

```
ex:- a: active {
        Color: green;
     }
```

Date / /
Page

- Mouse Hovering (hover)

=) It is used to develop some style when user perform mouse hovering.

```
eg:- #a1 : hover {
        background-image: url(abc.jpg);
     }
```

* Pseudo Element Selector (::)
  scope Resolutory operator

=) Pseudo element selector are denoted by the symbol scope resolutory operator.

- ::after :- (p::after)

=) Insert content after every <p> element.

- ::before (p::before)

=) Insert content before every <p> element

- ::first-letter (h1::first-letter)

=) Selects the first letter of every <h1> element.

```
ex:- h1::first-letter {
        font-size: 100px;
        color: aqua;
     }
```

Date __/__/____
Page _____

:: first-line (p:: first-line)

) Selects the first line of every <p> element.

:: selection (p:: selection)

) selects the portion of an element that is selected by a user.

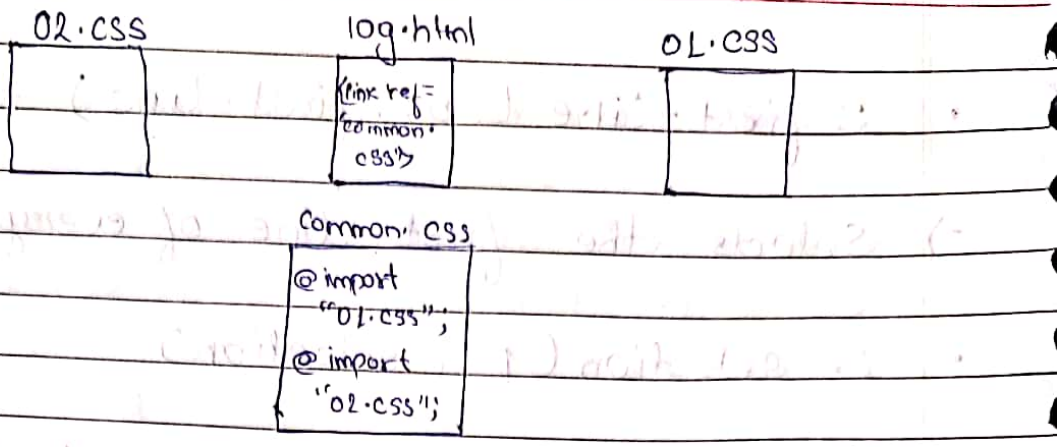
ex:- p:: selection {
background-color: red;
color: aqua;
}

Coloring properties:-

border: 1 px Solid red;
background-color: # aJed64;
color: rgb(210, 205, 66);
rgba: (255, 65, 110, 0.5);
↓
alpha ⇒ brightness (0.1 to 1.0)
Also can use color picker.

How to use multiple CSS file for one HTML file?
We can use @import and link multiple CSS file into one common CSS file which is linked up with HTML file.

Date / /
Page



* Background Properties:-

- Background-color: color;

=) It is used to give color to the background.

- Background-image: url(src);

=) It is used to give image as the background.

=) We have to use url to specify the image path.

- Background-repeat: no-repeat;

=) It is used to handle background repeating.

- Background-position: center/left/top/bottom;

=) It is used to set the position of the background.

Date __/__/____
Page _____

- Background Attachment : fixed / scroll;

=) It is used to handle the attachment of the background.

* Border Properties :-

- Border-style : ridge / dotted / dashed / solid / double ....

=> It is used to specify the style of the border.

=) Values :- groove / grade / solid / double and more.

- Border-width :

=) It is used to specify the thickness or width of the border.

- Border-color :-

=) It is used to specify the color of the border.

- Border-left :- It is a border property which is used to style the left border.

- Border-top :- It is a border property which is used to style the top border.

Date __/__/____
Page _____

- Border-right:- It is a border property which is used to style the right border.
- Border-bottom:- It is a border property which is used to style the bottom border.

* Border-style: solid dashed dotted groove;
  (Top, Right, Bottom, left)

* Border-Radius

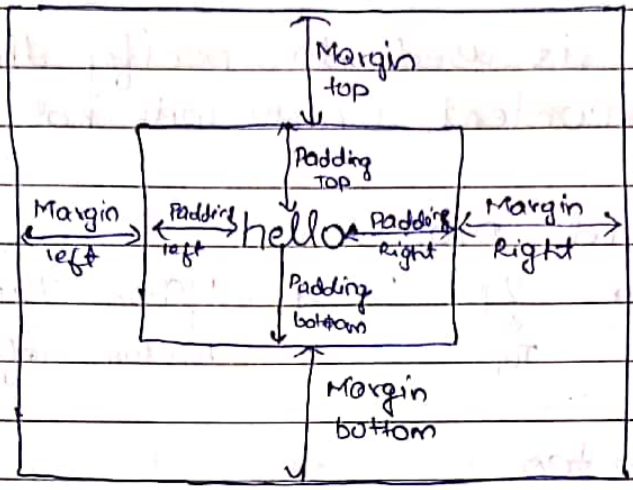
=> It is a border property which is used to specify curve in the corner of the border.

=> We can give the values in % or pixels.

=> border-top-left radius
border-top-right radius
border bottom-left radius
border bottom-right radius

Date __/__/____
Page _____

* Box - Model :-



* Margin Properties :-

- Margin is used to set the margin from the computer screen.

- Short hand property
  margin : 0px auto;
  ↓ ↓
  Vertical horizontal
  height width

- margin-top :- To set the margin from top.
  margin-right :- To set the margin from right.
  margin-left :- To set the margin from left.
  margin-bottom :- To set the margin from bottom

Date __/__/____
Page _____

* Padding Properties :-

- Padding is used to specify the space around content. (Auto will not work)

- Shorthand property :-
  Padding : 2px 30px 49px 70px;
  (2px → Top, 30px → Right, 49px → bottom, 70px → left)

- Padding - top
  padding - right
  padding - bottom
  padding - left

* Box - Sizing Property :-

- It maintain the actual height and width of the container.

- Developers will use box sizing properties as the first css code in every CSS file.

- Eg :- * {
  Universal selector box-sizing : border-box;
  margin : 0%; → Indicates initial margin of the user interface
  padding : 0%; → Indicates initial padding of the user interface.
  }

Date __/__/____
Page _____

* Display Properties :-

- Display properties is used to specify how the content should be displayed in the output.

i) display : inline ;

- It is a display properties which converts element into inline properties.

- ex :- #main > div {
  display : inline ;
  height : 100px ;
  width : 100px ;
  }

ii) display : block ;

- It is used to convert inline elements into block level elements.

- ex :- #main > div {
  display : block ;
  }

iii) display : inline-block ;

- It is used to convert inline elements into semi block elements. It means contents will be

Date / /
Page

displaying in the same line but we can give height and width.

- Eg:- #main > span {
  display: inline-block;
  height: 500px;
  width: 400px;
  }

* Outline Properties:-

- Outline properties are used to specify the outline for a content or a container.
- Outline comes after border.

i) Outline-width:- It is used to specify the width of the outline.

ii) outline-color:- It is used to specify the color of the outline.

iii) Outline-style:- It is used to specify the style of the outline.

iv) Outline-offset:- It is used to specify the space between the border and the outline.

Date / /
Page

Note:-

We cannot specify the style for particular side.

* Text-decoration:- To decorate the text we will go for text decoration.

* i) text-align:- text-align is used to align the text for center, right, left.

ii) text-decoration:- It is used to specify the decoration of the text like overline, line-through, underline, none. It is also a shorthand property.

iii) text-decoration-color:- It is used to specify the color for decoration.

iv) text-decoration-style:- It is used to specify the style of the decoration like dotted, double, dashed, wavy.

v) text-transform:- It is used to transform a text to uppercase and lowercase.

Date __/__/____
Page _____

* Text Spacing :-

i) text-indent:-

- It is used to specify initial gap or indent before starting a text.
- Ex:-
```
h4 {
    text-indent: 50px;
}
```

ii) letter-spacing:-

- It is used to specify the space between each letters or characters.
- Ex:-
```
h4 {
    letter-spacing: 5px;
}
```

iii) word-spacing:-

- It is used to specify the space between each and every words.
- Ex:-
```
h4 {
    word-spacing: 10px;
}
```

Date __/__/____
Page _____

iv) Line-height :-

- It is used to specify the space between each line of text.

Ex:-
```
p{
    line-height: 20px;
}
```

v) White-space :-

- It is used to wrap the text for the exact width of the computer screen.
- Values we can use wrap & nowrap.

* Shadow Properties :-

i) Text-shadow :-

- It is used to give shadow for the text

Syntax :-
text-shadow: horizontal vertical blurness color;

Ex:-
```
h1 {
    text-shadow: 2px 2px 2px blue;
}
```

Date __/__/____
Page _____

ii) box-shadow:-

- It is used to give shadow for the box. other containers except text.
- Syntax:-
  box-shadow: horizontal vertical blur color;
- Ex:-
  ```
  div {
    box-shadow: 10px 10px 5px black;
  ```

* Adding fonts to our HTML code:-

- To use any fonts, logo, icons etc. we have to link 'font awesome' web page to our html ed code. In order to do that we have to.

1. Open the browser search for 'font awesome cdn'
2. Click on the first link (cdnjs)
3. We will get to some info in that use first red coloured and copy that link (</>)
4. Paste it in the head of the html page. (now we are ready to use icons and logo)

Date / /
Page

1. In the browser search for "font awesome.com" (official website).
2. Click on icons, scroll and select the icons and (some are paid some are unpaid).
3. Select your icon and copy the snippet.
4. Paste that code in your html code wherever you want.

* Position Properties :-

- Position property is used to specify particular position of the elements

i) position : static;

=> Position static is nothing but it makes the container rigid where we cannot specify the position.

ii) position : relative;

=> It will take the position with respect to its sibling. We can set the position of the element by using left, right, top, bottom.

Date __/__/____
Page _____

=> Usually relative will be given for parent container.

iii) position: fixed;

=> If the position of a container is fixed it will be fixed in the computer screen even through user scrolls the fixed container will not move.

=> But we can set initial position by using top, left, right, bottom.

iv) position : absolute;

=> Generally used to set the position inside a container.

=> Normally absolute is the child of relative

=> It doesn't create white background.

=> It is transparent.

=> Once we give background it will transparent.

=> By default it displays all the containers in initial point we have to specify the position.

Date / /
Page

iv) position: sticky;

=> It is used to make an element sticked. where if the page is scrolled the sticked element will be in the same position.

=> Usually used in nav bars.

* Font Properties :-

- Font properties are used to specify the style for texts, icons, emojis.

i) font-size : ;

=> It is used to specify the size of the font.

ii) font-family : ;

=> It is used to specify font-family

iii) font-style : ;

=> It is used to specify the style of the font.

iv) font-weight : ;

=> It is used to specify the boldness of the font.

Date __/__/____
Page _____

* z-index properties :-

- z-index properties are used to specify the depth of an element.
- values can be given in negative and positive numbers.

* Flex - properties :-

- The flex is use to set flexible length on flexible items.
- If we want to se use flex properties parent container should be displaying flex.

i) flex - direction :- It is use to specify direction of child containers.

Values :-
flex - direction - row - Default
column
row - reverse
column - reverse.

ii) flex - basis :- It is a flex - property which is use to increase or decrease the length of the container.
values :- In pixel.

Date __/__/____
Page _____

iii) flex-grow:- It is use to increase the length of child containers to fit the length of parent container.

iv) flex-shrink:- It is used to decrease the length of child containers only when the child containers are conjusted. shrink will be done to maintain the actual length of the child container.

* Transform Properties:-

- Transform property are used to transform the element.
- transform: translate (100px, 100px)

i) translate() :- It is used to translate the position of the container with respect to given values.

ii) rotate() :- It is used to rotate element with respect to degree of angle given.

iii) skew() :- When we want filled on any element we use transform skew.

Date __/__/____
Page _____

* Transition property :- It is used to specify the transition should be applied on what style.

ex:-
```
div {
    height : 500px;
    width : 500 px;
    transition : all;
    transition-duration : 10sec;
    transition-delay : 0.1s;
}
div:hover { height : 700px;
            width : 700px;
}
```

* Transition-duration :- It is use to specify the time duration of the transition.

ex:- transition-duration: 5s;

* transition-delay :- It is use to specifying delay time before performing.

ex:- transition-delay : 0.1s;

Date / /
Page

* Animation :-

- Changing the style of an element with respect to time is called as animation.

i) animation-name :- It is used to give a particular unique name for the animation.

ii) animation-duration :- It specifies how much time it should be taken to perform animation for 1 iteration.

iii) animation-iteration-count :- It is used to specify how many times the animation should be repeated.
Values :- 1, 2, infinite.

iv) animation-delay :- It is used to specify delay time before performing animation.

v) animation-~~duration~~direction- It is used to specify direction of performing animation.
Values :- alternate, alternate-reverse, normal, reverse.

vi) animation-timing-function :- It is used to control the speed of the animation at starting and ending.
Values :- ease-in, ease-out, ease-in-out, linear.

Date __/__/____
Page _____

vii) @keyframes :-

- It is used to specify the styles of animation.
- It use the animation name to perform animation.
- It can be used in two ways :-

@keyframes {

a) from {
    = ;
}
to {
    =
};
}

b) 0% {
    - }
20% {
    = }
40% {
    - }
60% {
    - }
80% {
    -
}
100% {
    - }

Date / /
Page

* Overflow Properties :-

- Overflow properties are used to handle the overflowing elements.

i) overflow: visible :- It is a default value of overflow property where it display the overflowing content as it is.

ii) overflow: hidden :- It is use to clip the overflowing content but it will not give any scrollbar.

iii) overflow: scroll :- If we use overflow scroll then it will give scroll bar even trough if the content is overflowing or not.

iv) overflow: auto :- It will give the scroll-bar only when content is overeflowing.

Date __/__/____
Page _____

(Smooth backgrounds)

* Gradient - Properties :-

- If we want to use gradient properties it must be used in background image.

i) linear - gradient :-

linear-gradient: to left, red, green, black;

direction, color

- We can use deg in direction and corners also. We can also use n number of colors

ii) Radial - gradient :-

radial-gradient: circle, blue, red, black;

- We can use % in any one color to increase it.

iii) conic - gradient :-

conic-gradient: red, blue, green;

- We can use % in any color to increase it.
- We will not get auto suggestion.

Date / /
Page

* Grid Properties :-

- It is a CSS property, to overcome the drawbacks of HTML table we use grid.
- Drawbacks of table :-

i) We cannot specify height and width for html table.

ii) It takes only content height & width.

- Grid - template - columns :-

⇒ It is used to specify the number of columns and size of the columns.

- Grid - template - rows :-

⇒ It is used to specify the number of rows and size of the rows.

Note :- • In grid, to make the alignments horizontally we use justify content.

- Since it is grid we use align content for vertical alignments.

Date __/__/____
Page _____

- Gap :- It is used to specify some gap between the child containers inside grid

Javascript

* History of Javascript :-

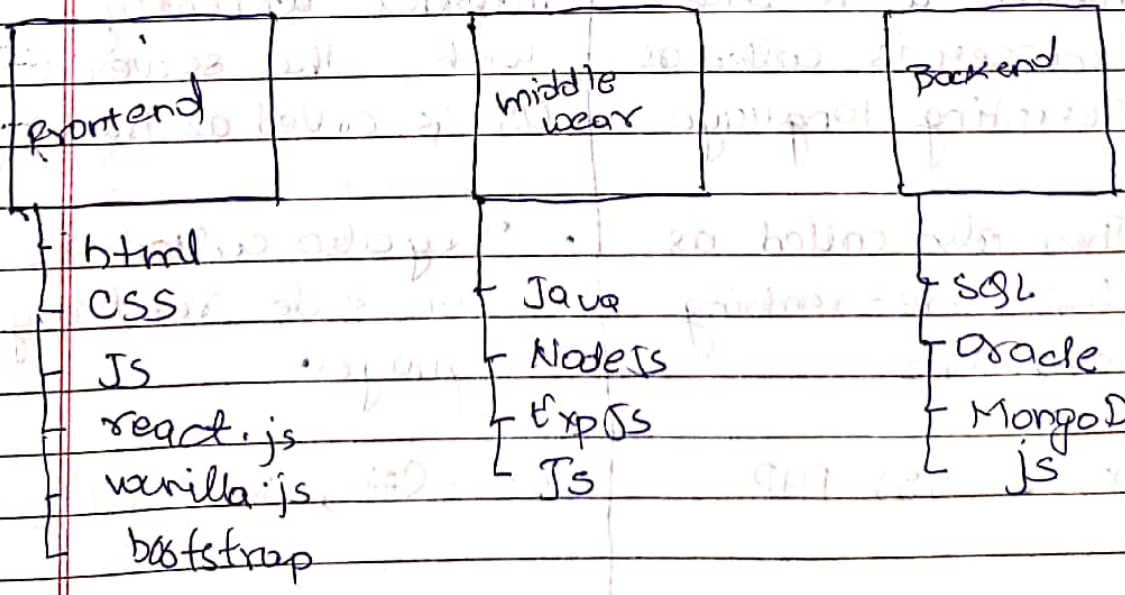
- Brendan Eick developed javascript in 1995.
- Eick was working in a company called netscape (browser).
- Initially the language was released by the name Javascript.
- Later changed it for Mocha / Live script.
- The company ecma (Europian computer manufacture association) has upolated livescript and released as ~~ECMA Script~~ Ecma script & Java script.
- The current version of Java script is ES6.

Date __/__/____
Page _____

JA

* Difference between JAVA & JavaScript.

JAVA	JAVAScript
• It is both compiled and interpreted.	• It is only interpreted.
• Asynchroized	• Synchronized
• Multi stack	• Single stack
• Strongly Typed	• Weakly Typed
• It will run in virtual machine.	• It runs in browser.



Date __/__/____
Page _____

Q:- What is Javascript?

Ans:- Javascript is a object-based scripting language which is use to give the dynamic functionality to the web-pages.

*	Object-Based	Object-Oriented
•	Here we have inbuilt objects.	• Here we don't have any inbuilt object.
•	Source code is not dependent on the object.	• Source code is dependent on object.

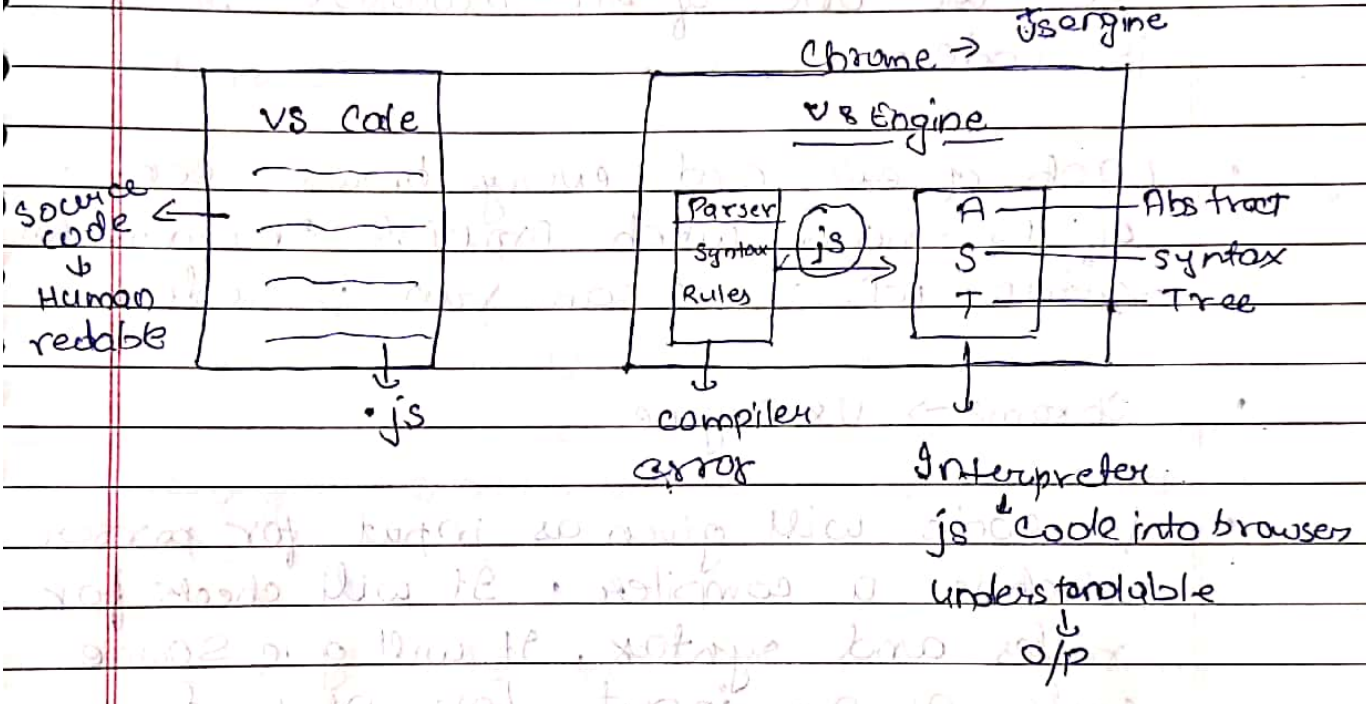
*	Scripting language	Programming language
•	Any computer language which is used to interact with the browser is called as scripting language	• Any computer language which is used to interact directly with the server or OS is called as Progr. lang.
•	They also called as client-side-rendering languages	• They also called as server-side-rendering languages.
•	Ex:- JS, VBS, PHP	• Ex:- C#, Java, C++

Date: / /
Page

Q:- What is dynamic functionality?

Ans:- On any of the web page if the style and content is changing based on dynamic inputs (click, mouseover, submit, scroll).

* JS Architecture :-



Chrome → V8 engine
Firefox → Spider monkey
IE → Chakra
Safari → Js core
1) Opera → Carakan
2) Adobe Flash :- Tamarin
3) Oracle JDK :- Nashorn
4) UC Browser :- Blink
5) Brave :- Blink
6) Mozilla - Rhino

Node js ⇒ Rayon Dal

Date __/__/____
Page _____

- We will write js code in one of the editor it is also called as source code.
- JS code should be saved in .js file extension.
- We use one of the browser to execute js code.
- Each of eve and every browser consist of js engine which provide environmental setup where we can run js code.
- Chrome → V8 engine
- js code will given as input for parser which is a compiler. It will check for rules and syntax, It will give same code as an input for Abstract syntax Tree which is interpreter.
- Interpreter will convert js code into browser understandable or machine understandable code or format.
- By using machine understandable code browser will give the output.

Note :- If we want to run js code outside the

Date / /
Page

browser we have to use node.js which is developed Bryon Dal (Indian Developer).

* We can use JS code in 3 ways :-

i) Inline js
ii) Internal js
iii) External js

i) Inline js :- The process of using js code in the console is called as inline js.

ii) Internal js :- We can use js code within the head or within the body by using script tag is called internal js.

iii) External js :- We can use js code in external file with .js file format and linking it in the HTML code by using script tag.

*	Async	Defer
•	If we use async attribute in script tag, first html elements will executed then the js code will executed.	• If we use defer attribute in script tag, both html elements and js code will execute simultaneously.

Date __/__/____
Page _____

* Tokens :- Tokens are the smallest individual part of the program.

* Identifiers :- Identifiers are the name which is given in js code.

* Keywords :- Keywords are the pre-defined words which has its own meaning.

- In js we have 60+ keywords.
- Keywords are :-

abstract	arguments	await*	boolean
break	byte	case	catch
char	class*	const	continue
debugger	default	delete	do
double	else	enum*	eval
export*	extends*	false	final
finally	float	for	function
goto	if	implements	import*
in	instance of	int	interface
let*	long	native	new
null	package	private	protected
public	return	short	static
super*	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	yield

Date __/__/____
Page _____

* Literals (classified into)

Premitive type
↓
immutable in nature
↓
Object dereferencing
↓
- BigInt
- Number
- String
- Undefined (UD)
- Boolean
- NULL

Non-primitive type
↓
mutable in nature
↓
Object referencing
↓
- functions / methods
  ↳ 9 Types
- Object
  - Time
  - Math
  - Date
- Array
  - filter()
  - map()
  - reduce()

[for, for of, for in, for each, while, do-while]

- Class

* Variables :- • Variables is a named memory location which is used to store some value or data. It can change multiple times during execution.

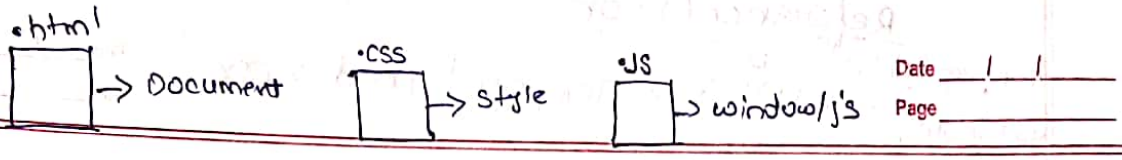
• In JS we have 3 types of variables :-

Date / /
Page

i) var
ii) let
iii) const

	var	let	const
Variable declaration	✓	✓	✗
	var a;	let b;	const c;
Variable initialization	✓	✓	✗
	a = 10	b = 10.50;	c = 24;
Variable decl. & init.	✓	✓	✓
	var a = 20	let b = 24;	const = 25;
var re-initialization	✓	✓	✗
	a = 40	b = 29	c = 10.25
var re-decl.	✓	✗	✗
	var a = 100	let b = "Hi"	const g = b;

Note:-
- All the var variables will store in window object. Hence, var is global scope.
- Both let and const are local scope since they will not get stored in window object.
- window is a inbuilt object which is present inside to BOM.



Date __/__/____
Page _____

* Output methods of Javascript :-

→ console.log(); // dev output statement
- Used to print the output in console.

→ document.write();
- Used to print output as document

→ document.writeln();
- It will give 1 space after every output.

→ alert();
- It is used to develop alert popup.

→ confirm();
- Confirmation popup.

→ prompt();
- It will develop popup which is used to take user input.

* Operators :-

- Operators are the symbols which is used to perform some operations on operands.

eg :- 10 (+) 10 — Operands

Operator

Reference Error

console.log(y); ↔ let y=10; → TDZ → Temporal Dead zone

Date / /

Page

i) Arithmatic operator :- +, -, /, %, *, ++, --

ii) Relational operator :- >, <, <=, >=, ==, !=

iii) logical operator :- && , ||

iv) TypeOf operator :- console.log(typeof var);

v) Conditional operator :-

Syntax :- (Query) string

condition ? "value1" : "value2"

10 > 2 ? "correct" : "wrong";

* :- Hoisting :-

- Utilizing a variable before declaring is called as Hoisting.
- Hoisting is possible for var only.
- Hoisting is not possible for let and const.

Note :- • Since var is global variable hoisting is possible.

- If we try to hoist let & const we will get reference error. Reference

- => will convert string into number if there is only number value.

Date __/__/____

+ => it will concatinate if one of the value is string else addition

Page _____

NaN => Not a Number

error is nothing but the time gap between variable calling function and variable declaration.

* Typecasting :-

- Converting from one type of data to another type of data is called as Typecasting.
- In Js we have two types of Typecasting :-

i) Implicity Typecasting
ii) Explicit Typecasting

i) Implicit Typecasting :- Typecasting done by Javascript Engine is called Implicit Typecasting.

```
Ex:- var a = prompt ("enter age"); //type is string
     console.log ('hello10' - 10); //NaN
     console.log (10 - true); // 9
     console.log (10 + True); //11
```

ii) Explicit Type casting :- Developer forcefully converting one type of data to another type of data is called as Explicit Type casting.

Date __/__/____
Page _____

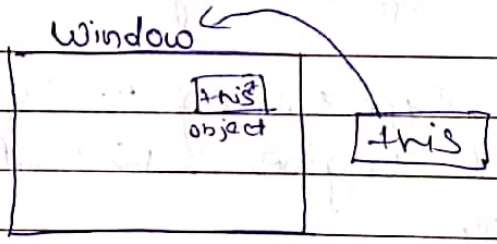
Ex:-

```
var age = prompt("Enter your age")
var o1 = Number(age)
        or
var age = Number(prompt("Enter age"));

let k = String(156)
console.log(typeof k);
```

* This Keyword :-

- This Keyword is used to point for window object or current object.




```
console.log(window);
console.log(this);
```

* Conditional Statements :-

- Conditional statements are used to check the logical ~~codes~~ conditions in the JS code.

Date / /
Page

i) if statement.

- Syntax :-
```
if (condition)
{

}
```
Example :-
```
if (age < 18)
{
    log (age);
}
```

ii) if - else statement.

- Syntax :-
```
if (condition)
{

}
else
{

}
```
EX :-
```
if ( age < 18)
{
    log (age);
}
else
{
    log (age);
}
```

iii) Nested if - else statement.

- Syntax :-
```
if (condition)
{
    if (condition)
    {

    }
}
else
{

}
```
```
if ( age <= 18)
{
    if ( age = 18)
        log (age)
    else
        log (age)
}
else
{
    log (age)
}
```

== → compare value //true var a = 10, var b = "10"

=== → compare value with datatype //false

Date __/__/____
Page _____

4) Else if

- Syntax:

```
if (condition)
{

}
else if (condition)
{

}
else
{

}
```

```
if (age <= 18)
{
log (age)
}
else if (age <= 18)
{
log.(age)
}
else (age > 18)
{
log (age)
}
```

* Write a JS code whether the given number is more than 10 or not.

=) ~~let a = prompt ("Enter a num~~

```
let number = Number (prompt ("Enter a number"));
if (number > 10)
{
    console.log ("Number is greater than10")
}
else
{
console.log ("Number is lessthan 10);
}
```

Date / /
Page

Q:- Write a JS code to check wheather the given number is odd or even.

Sol:-
```
let num = Number(prompt("Enter "));
if (num % 2 == 0)
{
  console.log("Even");
else
  {
    console.log("Odd");
```

* Interpolation:-

- Interpolation is the way of using variables in printing statement by using back ticks.
- Both <pre> tag & back ticks works same that is it will print the exact content how we have written in the code.
- Ex:-
```
var car = 'BMW'
let engine = 1800;
const clr = 'black';
console.log(`My car name is ${car}
             My car engine is ${engine}cc
             My car color is ${clr}`);
```

Date __/__/____
Page _____

* Loops :-

- Whenever we want to perform similar kinds of operation we use loops.

- In JS we have 6 types of loops :-
  i) for loop
  ii) while loop
  iii) do - while loop
  iv) for - each loop
  v) for - of loop
  vi) for - in loop

i) for loop

- Syntax :- for(initialization; condition; inc/dec)
  {
  -
  }

- Whenever we have the range it means starting point & ending point we should go for for loop.

- Ex :- for(var i = 1; i <= 10 ; i++)
  {
  console.log(i)
  }

Date 1/1
Page

ii) While loop

- When we don't have the range we go for while loop.
- Syntax:- while (condition)

```
var a = 10
```

- Ex:-

```
while(a>=1){
    console.log(a);
    a--;
}
```

GEC
(Global Execution Context)

- GEC stands for Global Execution Context.
- It is a memory space which is created to execute javascript code.

1. Whenever the JS code enters the JS engine GEC will be created.

2. GEC will follow 2 phases.

Date / /
Page

i) Memory will be allocated
ii) it start the execution from top to bottom & left to right.
Initially all the variables value will be undefined.

3. GEC is classified into two parts:
   i) Variable state
   ii) Function/exe. state

4. It will scan for two times.
   i) 1st scan, it will scan for variables.
   ii) 2nd scan, it will scan for functions

.js

```
var a=10;
let b=
log(a);
log(b);
```

Variable state	function/exe st.
window this ↳ var a=10 let b=UD	log(a) log(b)

.js

```
const t=True
var s="Spij"
```

GEC

variable state	function/exe st.
this	

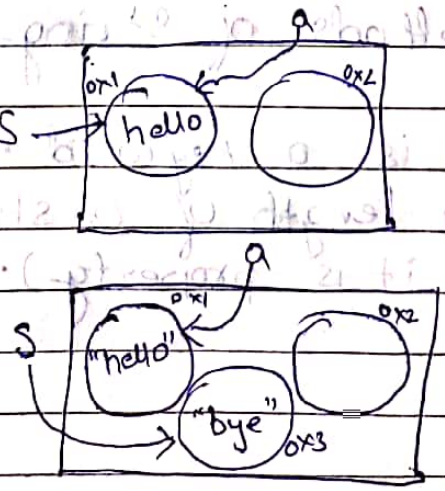
Date / /
Page

* Literals

Primitive lit
↓
Immutable in nature
↓
Object dereferencing
↓
Numbers
String
boolean
UD
Big Int
Null

Non-primitive lit
↓
Mutable in Nature
↓
Object referencing
↓
functions
Array
object
class

* Immutable in Nature :- Immutable is nothing but, whenever we try to reinitialise primitive literals instead of updating the same memory value it will create new memory space, but old memory space will not be erased.



Date / /
Page

```
var s : "Hello"
var a = s;
```

vardatatype varname value

```
log(s) - hello
log(a) - hello
s = "bye";
log(s); - bye
log(a); - hello
```

* String :-

- The group of characters is called as string.

- We can declare strings in 3 ways :-

i) "Hello"
ii) 'Hello'
iii) `Hello`

```
log("Hello") // Hello
log("'Hello'") // "Hello"
log('"Hello"') // "Hello"
```

- Inbuilt methods of string :-

i) length :- It is a keyword. It is used to fetch the length of a string. (It is not a function it is property).

Date / /
Page

toUpperCase() :- It is used to convert a string into uppercase.

toLowerCase() :- It is used to convert a string into lowercase.

toLocaleLowerCase() :- It is used to convert a string into ~~uppercase~~ lowercase for local languages like turkish.

toLocaleUpperCase() :- It is used to convert a string into uppercase for local languages like turkish.

replace() :- It is used to replace a character with another specified character.

charAt() :- It is a function which is used a character from a string based on index.

slice() :- It is used to fetch a part of the string based on index.

```
s="hello"; log(s.slice(0,3);
```

split() :- It is used to convert string into an array.

s.split() → length=1

s.split('') → length of every character

Date __/__/____
Page _____

* Null :-

- Null is a primitive datatype which is immutable in nature.
- In JS null is used when we don't want to initialize the any value. If we use NULL memory space will not be created.

Note:- When we try to print type of NULL it gives the output object which is non-primitive data.

* BigInt :-

- It is a primitive data which is immutable in nature.
- When we want to declare long number we can use BigInt.
- We can declare BigInt in 3 ways :-

```
var a = 1234567n;
var b = BigInt(1256789);
var c = BigInt('12567890');
```

Ex:- If we declare as number then it will show output in mathematical expression that's why we have to use Bigint.

Date __/__/____
Page _____

* Boolean :-

- It is a primitive type of data which is immutable in nature.
- Boolean consist of only two values:-
  i) True
  ii) False
- If the condition is satisfied we will get the output true.
- If the condition is not satisfied we will get the output false.

```
log(10==10) //true;
```

* Undefined :-

- It is a primitive type of data which is immutable in nature.
- If any variable is not initialized with any value we will get the output as undefined.

```
let b;
log(b) //undefined;
```

Date __/__/____
Page _____

Number :-

- Number is a primitive literal which is immutable in nature.
- All Both whole numbers are and decimal numbers belongs to number datatype.

Functions

- Function is a block of statement which will get executed whenever it is called or invoke.
- Functions are non-primitive data which is mutable in nature.
- We have 9 types of functions in JS :-
  - i) Anonymous function
  - ii) Function Expression
  - iii) First Class function
  - iv) Named function
  - v) Arrow function
  - vi) Higher order function
  - vii) IIFE (immediate invoking function)
  - viii) Nested function
  - ix) Callback function

Date / /
Page

* Anonymous function :-

- The function which doesn't have name is called as anonymous function.
- Ex:-
```
function()
{
    log("hello");
}
```
- Since anonymous function doesn't have any name we can store it in a variable.
- Ex:-
```
var a = function()
{
    log("hello");
}
```

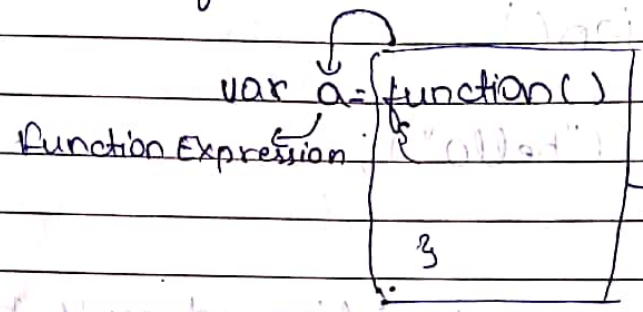
* Function Expression :-

- A variable which is initialised with a function is called as function expression.
- Ex:-
```
var a = function()
{
    log("Hello");
}
a();
```

Date __/__/____
Page _____

* First Class Function :-

- A function which is passed as the value for a variable is called as first class function.



* Named Function :-

- The function which has name is called as named function.

- Ex:
```
function a1()
{
    log("Hello");
}

a1();
```

* IIFE (Immediate Invoking Function)

- A function which is invoking immediatly in the next line by using paranthesis () is called as Immediate Invoking function.

Date __/__/____
Page _____

- Ex:- (function()
{
log("Hi");
})
()

* Arrow function :- // Lazy developers function :-

- To reduce the syntax of normal functions we go for arrow function.
- Syntax to declare arrow function :-

Fat arrow
↑
() => Syn Statements;

- Ex:-

a = () => {
log("hi");
log("bye");
}

- Rules of Arrow function :-

i) If we want to use multiple statements then braces are mandatory.

ii) If we want to use return type braces are mandatory.

iii) This keyword will not work in arrow function.

Date __/__/____
Page _____

- Example for function with parameters :-

```
x = 10; y = 20;
function mo(a, b)
{
    var res = x+a+b;
    log("Result is ${res}");
}

mo(x, y);
```

- Example for function with return type :-

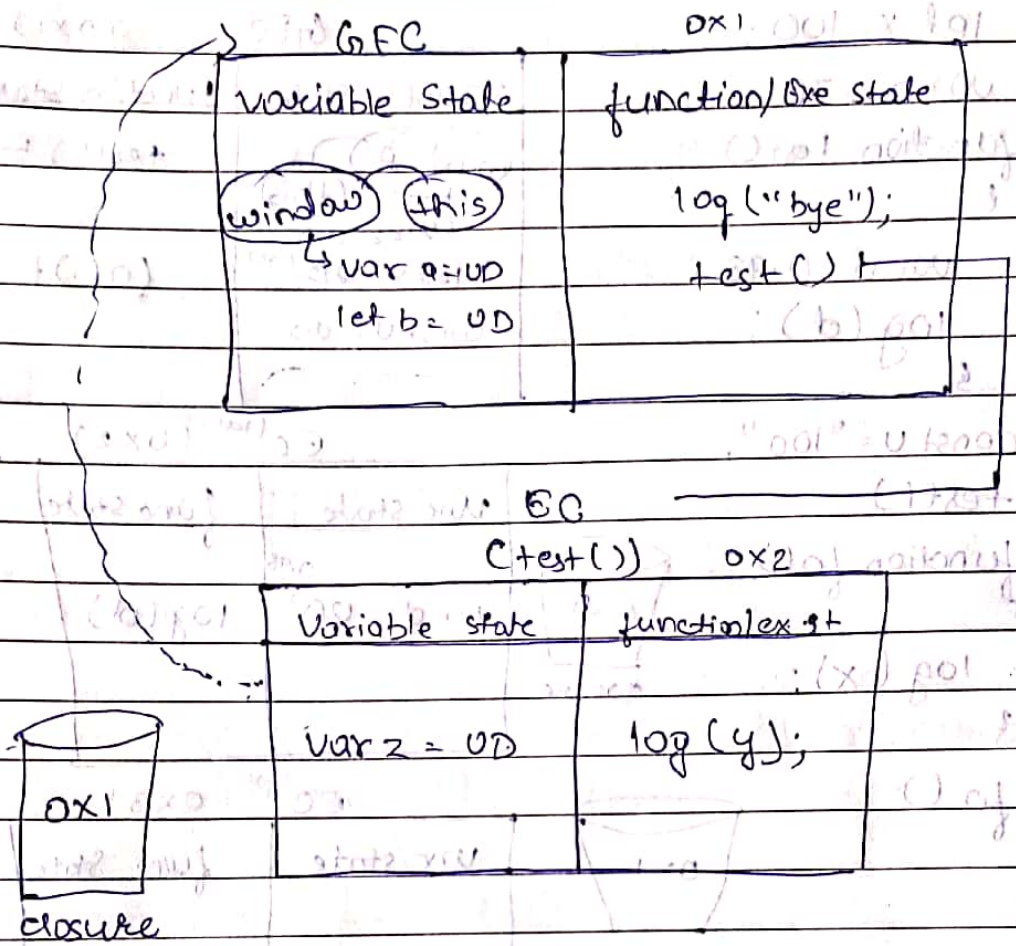
```
function mo(a, b)
{
    var res = a+b;
    return res;
}

var o = mo(40, 40);
log(o);
```

-

```
js
var a = 10
let y = 20
function test()
{
    var z = "hello"
    log(y);
}
log("bye");
test();
```

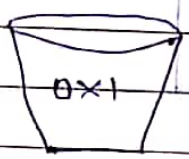
Date / /
Page



- Closure:- Closure is a memory space which consist of address of the previous execution context.

Date __/__/____
Page _____

```
let x = 100;
var z = 70;
function test()
{
  var d = 0.98
  log(d);
}
const u = "100";
test()
function fo()
{
  log(x);
}
fo()
```



GEC (0x1)

Variable State	Function State
window, this var z = UD let x = ~~UD~~ 100 const U = UD	test() → fo() →

EC(test) (0x2)

var state	func state
var d = ~~UD~~ 0.98	log(d)

EC(fo) (0x3)

var state	func state
	log(x)

Date __/__/____
Page _____

i) factorial

=> var number = prompt("Enter the number for Factorial")
var fact = 1
for(let i = 1; i <= number; i++)
{
fact = fact * i;
}
log(fact);

ii) fibbonacciSeries

=> var total = prompt("Enter the total numbers")
var fibbonacci = [total]
fibbonacci[0] = 0;
fibbonacci[1] = 1;
for(i=2; i<=total; i++)
{
fibbonacci[i] = fibbonacci[i-1] + fibbonacci[i-2];
log(fibbonacci[i]);
}

iii) Prime Number

=> var input = prompt("Enter the number to check");
if(input == 2 || input == 3 || input == 5)
log("Prime Number");
if((input%2==0)||(input%3==0)||(input%5==0))
log("No prime Number");
else
log("Prime Number");

Reverse a string

iv)
```
String = prompt ("Enter the string");
.l= string.length
var newstr = "";
for (var i= l-1; i>=0; i=i-)
{
    newstr = newstr + string[i];
}
log (newstr);
```

v) Palindrome

=>
```
var string = prompt ("Enter the Number");
var l= string.length
var flag = 0;
for( let i=0; i<l/2; i++)
{
    if ( string[i]==string[l-1-i])
    {
        flag++;
    else {
        flag=0;
    }
}
if ((flag==l/2){
    log ('Palindrome');
} else
    log ('Not a Palindrome');
}
```

Date __/__/____
Page _____

* Higher Order Function :-

- A function which takes one more function as the argument is called as Higher Order function.

Higher Order Function

```
function test (a) -  Parameter
{
    a()
}

test ( () => log("hello"); - Argument
```

Callback function.

* Callback Function :-

- A function which is passed as the argument for higher order function is called as Callback Function.

Higher Order Function

```
function test (a)
{
    a()
}

test (() => log ('Hello');
```

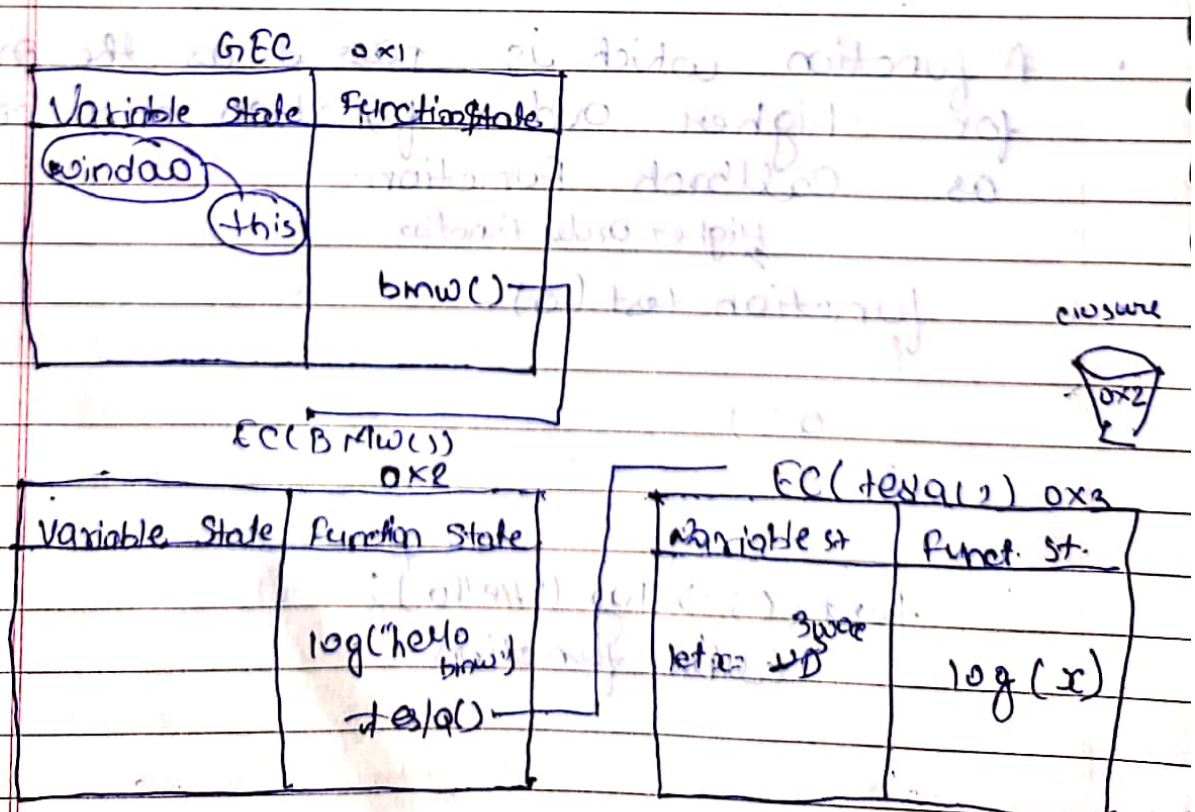
Callback function

Date __/__/____
Page _____

* Nested function :-

- A function which is declared inside another function is called as Nested function.

```
function bmw()
{
    log("Hello bmw");
    function testa()
    {
        let x = "3000cc";
        log(x);
    }
    return testa;
}
bmw()()
```

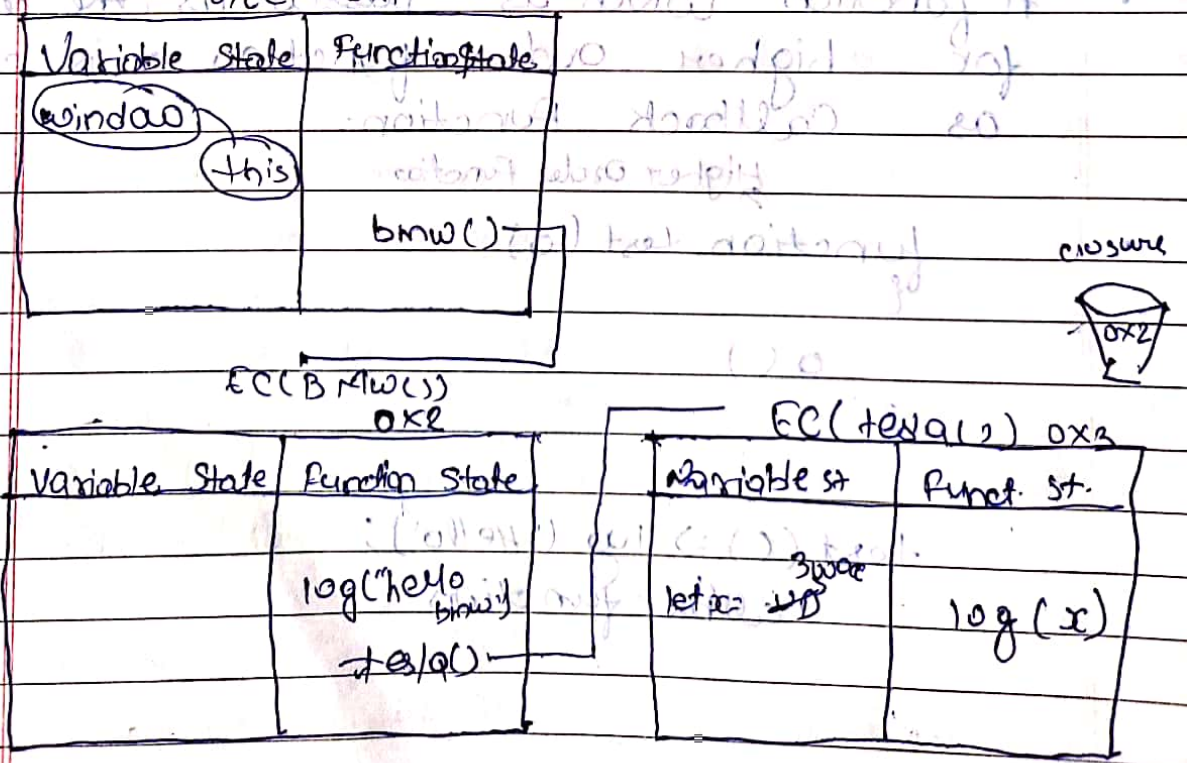


Date / /
Page

* Nested function :-

- A function which is declared inside another function is called as Nested function.

```
function bmw()
{
    log("Hello bmw");
    function tesla()
    {
        let x = "3000cc";
        log(x);
    }
    return tesla;
}
bmw()()
```



Date __/__/____
Page _____

```
let x = 100
function a()
{
    log(x); // lexical scope
    var y = 8;
    function b()
    {
        log(x); // lexical scope
    }
    return b;
}
a()()
```

⇒ Lexical Scope :- The ability of javascript engine to utilize a variable in the previous execution context when it is not available in current execution context.

Date __/__/____
Page _____

* Array :-

- Array is a non-primitive datatype which is used to store homogenous as well as hetrogenous type of data.
- We can declare array in two ways :-

i) By using new keyword

ii) ex :-
```
let arr= new Array()
arr[0]= 10;
arr[1]= 10.50
arr[2]= 'hello'
```

ii) By using literal method

ex :-
```
var abb= [10, 'hello', true, 10.50]
         0     1       2     3
```

* Inbuilt methods of Array :-

i) Length :- It is used to fetch the length of an array.

Ex :- log(arr.length);

ii) push() :- It is used to insert an element at the end of an array.

ex :-
```
arr= [10, 20]
arr.push(30)
```

Date __/__/____
Page _____

iii) pop() :- It is used to remove an element at the end of an array.

ex:- arr = [10, 20, 30]
arr.pop();

iv) unshift() :- It is used to insert an element at the 0th index.

ex:- arr = [20, 30]
arr.unshift(10);

v) shift() :- It is used to remove an element at the 0th index.

ex:- arr = [10, 20, 30]
arr.shift();

vi) slice() :- It is used to fetch the part of an array.

ex:- arr = [10, 20, 30, 40];
res = arr.slice(1, 4)

vii) reverse() :- It is used to reverse the arrays value.

ex:- arr = [10, 20, 30]
res = arr.reverse(); // 30, 20, 10

viii) join() :- It is used to convert the array values into string.

Ex:- arr = [10, 20, 30, 40]
res = arr.join()
log(res)

Date __/__/____
Page _____

* How to reverse a string in one line.

=> let arr = "Hello"
res = arr.split('').reverse().join('');
log(res);

ix) sort(): It is used to sort the array elements in ascending order.

* For-in loop:- It is a loop which is used to fetch the index of from the specifying array.

Syntax: for(key in objectF)
{
log( );
}

Ex:- arr = [10, 20, 30, 40, 50]
for(var ele in arr)
index
{
log(ele)
index
}

* For-of loop:- It is a loop which is used to fetch the array elements.

Syntax: for(var type var-name of array)
{
log(-)
}

Date / /
Page

Ex:- var arr = [10, 60, 40, 30, 20];
for (var ele of arr)
{
log (ele);
}

* For-each() loop:- It is a higher order function which is used to fetch array elements and index.

Syntax :-

arr_name.forEach(callback function())

Ex:-

var arr = [10, 60, 30, 40, 50]
arr.forEach((ele, ind) => log(`${ele} ${ind}`));

* Filter() :- It is a higher order function which is used to filter the array elements. (array values).

Ex:-

index	100	200	300	400	500

var price = [100, 200, 300, 400, 500]
var filtrec = price.filter((a) => {
return a < 300;
})

log (filtrec);

Date __/__/____
Page _____

* map() :- It is a higher order function which is used to map the array values. (mapping means adding some values in array elements or subtract some values in array elements).

filtered_data [100, 200]

Ex:- var map_v1 = filtered_data.map(
(a) => {
return a * 1.18;
})

log(map_v1);

* reduce() :- It is a higher order function which is used to get total result of array values.

Ex:- map_v1 = [118, 236]

acc	value
0	+ 118
118	+ 236
354	

var total = map_v1.reduce((acc, value) => {
return acc + value;
}, 0)

log(total);

syntax:- reduce(function, 1/0)
↳ thumb rule
:- * 0 - +

Date __/__/____
Page _____

* Object :-

- Object is a non-primitive datatype which is mutable in nature.
- In js object is a stand alone entity with properties and types.
- Object is a real-time entities ~~which is~~ having properties and types.
- We can develop js object in two ways:-

i) Literal method
ii) Using new keyword

i) Literal method :-

Syntax :-

identifier
↑
var type obj_name = {
    Key1 : value1,
    Key2 : value2,
    Key3 : value3
}

- How to access object :-

log(obj)

- Ex:- let obj = {
    un : 'ABC',
    pwd : 123,
    address : 'xyz'
}

Date __/__/____
Page _____

- How to access particular data or record of object :-

```
log(obj.ub);
```

- If we want to insert new data for existing record in object :-

```
existing obj_name.new_dataname = data;
```

ex:-

```
obj.designation = "SDET"
```

- How to reinitialize already existing record :-

```
obj_name.key_name = updated data
obj.pwd = 1234
```

ii) By Using New keyword :-

Syntax :-

```
var_type obj_name = new Object()
```

Example :-

```
var obj2 = new Object()
obj2.ub = 'SRS'
obj2.pwd = 123
obj2.address = 'xyz'
```

Date __/__/____
Page _____

* Nested Object :-

- A object inside another object is called as nested object
- Ex:-
```
let obj = {
    name: 'ABCD',
    mob: 1234567890,
    photo: 'http://123.456'
    address: {
        state: 'WB',
        city: 'Kolkata'
    }
}
```
- We can use a function as value for key inside an object.

Ex:-
```
let obj = {
    name: 'ABCD'
    mob: 1234
    address: () => {
        log('hello');
    }
}

log(obj);
obj.add();
```

Date / /
Page

* Inbuilt methods of object :-

1. Object.keys() :- It is used to fetch only keys from an object.

ex :- var key = object.key(obj);

obj.keys()

2. Object.values() :- It is used to fetch only values from an object.
ex :- var val = object.values(obj);

3. Object.entries() :- It is used to fetch both key and values in the form of array.

ex :- var b = object.entries(obj);

4. Object.seal() :- It is used to seal an object if an object is sealed then we cannot add new data into that object, but we can update the existing data.

ex :- Object.seal(obj).

obj.phone = 12345 // reinstializing is possible

obj.pwd = 0174 // add new data is not possible.

Date __/__/____
Page _____

5. Object.freeze() :- It is used to freeze an object (If an object is freezed we cannot insert new data and also we cannot update the existing data.)

ex :- object.freeze(obj)
obj.phone = 63125 } Not possible.
obj.pwd = 0179

6. Object.isSealed() :- It is used to check weather the object is sealed or not.

eg :- object.isSealed(obj).

7. Object.isFrozen() :- It is used to check the object is frozen or not.

eg :- object.isfrozen(obj).

* Object Destructing :-

- This process of converting object records into variables is called as object destructing.
- Syntax :- var {type{key1, key2, ...}} = obj_name
- Ex :- var {office, branch} = obj;
  log(offices);

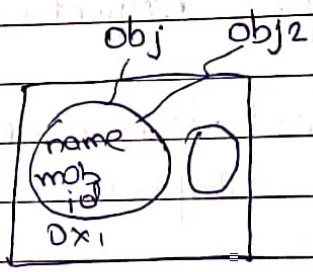
Date __/__/____
Page _____

* Mutable in Nature :-

- If we update/ reinitialize, the value of non-primitive data, then the value will be updated in same memory location.

- eg:- var obj = {
  name : 'ABC';
  mob : 12345;
  id : 012;
  }

var obj2 = obj
log(obj)
log(obj2)



- Math()

=> It is an inbuilt object which consist of some functions which is used to perform mathematical operations.

1) Math.round() :- It is used to round off a decimal number to the nearest whole number / non-decimal number.

eg:- log(math.round(10.6));

Date __/__/____
Page _____

2) Math.floor() :- It is used to round off a number (decimal no.) for the previous value.

ex :- log(math.floor(10.8)) //10

3) Math.ceil() :- It is used to round off a decimal no. for the next value.

ex :- math.ceil(100.12) //101

4) math.sqrt() :- It is used to return square root value of a number.

ex :- log(math.sqrt(16)) //4

5) math.cbrt() :- It is used to return cube root value of a number.

ex :- log(math.cbrt(27)) //3

6. math.random :- It is used to generate random decimal numbers.

ex :- log(math.random())

Date / /
Page

* Time :-

-> It is a inbuilt object which is used to make our function asynchronus.

1. setTimeout() - setTimout(fun, time in ms)

-> It is a higher order function, it will get executed after the specified time.

ex :-
```
setTimeout(() => {
    log("bye");
}, 5000);

setTimeout(() => {
    log("hello");
}, 10000);

function test() {
    log("hello test");
}
```

output :-
- hello test
- bye
- hello

Date __/__/____
Page _____

2. SetInterval() - setInterval(fun, time in ms)
=) It is a higher order function, which will get executed till infinity for every specified time.

ex:- setInterval(() => {
log('hello');
}, 2000)

* Date :-

- It is an inbuilt object which is used to fetch current date, time.
- If we want to use functions of date object, then we have to invoke date constructor.

Ex:- var x = new Date() //invoking

- Inbuit functions of date object :-

1. getDay() :- It is used to fetch the current Day index.

ex:- var a = getday()

2. getmonth() :- It is used to fetch the current month index.

ex :- var b = getMonth();

Date __/__/____
Page _____

3. getFullyear() :- It is used to fetch the current year.

4. getDate() :- It is used to fetch the current date.

ex:- var x = new Date();
log(x.getDate())

5. getHours() :- It is used to fetch the current hours.

6. getMinutes() :- It is used to fetch current minutes.

ex:- log(x.getminutes());

7. getseconds() :- It is used to fetch current seconds.

ex:- log (x. getseconds());

8. getmiliseconds() :- It is used to fetch current ms.

ex:- log (x.getmiliseconds());

Date / /
Page

Revision

Loops

⇒ Whenever we want to perform similar operations multiple times we use loops.

⇒ In Javascript we have 6 types of loops :-

i) for loop
ii) while loop
iii) do. while loop
iv) for each loop
v) for in loop
vi) for of loop

i) for loop

⇒ When we have the range we go for for loop.

⇒ Syntax :-

```
for(initialization; condition; inc/dec)
{
  //statements
}
```

⇒ ex :-

```
for(var i=0; i<10; i++)
{
    console.log(i);
}
```

Date / /
Page

ii) While-loop

→ When we don't have the range we go for while loop.

→ Syntax :-

```
while (condition)
{
  statements;
  inc/dec;
}
```

→ ex :-

```
var i=0;
while (i<10)
{
  console.log(i);
  i++;
}
```

```
var i=10
while (i>=10)
{
  log(i);
  i--;
}
```

iii) do-while loop

→ It is exit control loop where we will use when there is no-range.

→ syntax :-

```
do{
  //statements;
} while (condition)
```

Date __/__/____
Page _____

→ ex:-

```
do
{
  console.log('hello');
} while (10>20)
```

or

→

```
var a=0
do
{
document.write(n('hey');
a++;
} while (a<=5)
```

iv) for-each loop :-

→ It is a higher order function which will execute the callback function for every elements of specified array.

→

```
var arr = [100,200,300,400,500]
arr.forEach((a,b) =>
{
  consde.log(a+"      "+b);
})
```

Q:- Write a JS code to print the total value of all the elements in the array.

Sol:-

```
var arr = [100, 200, 300, 400];
var res = 0;
arr.forEach((v, i) =>
{
    res = res + v;
})
console.log(res);
```

v) for-in loop

→ It is a exit control loop which is to traverse or itirate index of array values.

→ Syntax:-

```
for(var name in arr name)
{
  //statements
}
```

→ ex:-

```
for var arr = [100, 200, 300, 400]
    for(e in arr)
    {
        console.log(e)
    }
```

Date / /
Page

vii) for-of loop

→ It is a exit control loop which is used to iterate array values or elements.

→ Syntax:-

```
for(var_name of arr_name)
{
// Statements;
}
```

→ ex:-

```
var arr = [100, 200, 300, 400]
for( e of arr)
{
    log(e)
}
```

* filter():-

→ It is an higher order function which is used to filter array elements.

→ Ex:-

```
var c = [100, 200, 300, 400, 500, 600];
log(c);
var f_c = c.filter((a)=>{
        return a<=300 });
log(f_c);
```

Date __/__/____
Page _____

* map():-

⇒ It is an higher order function which is used to map the array elements.

→ Map is nothing but adding some specific value or removing some specific value from array elements.

→ Ex:-

```
var gst_c = f_c(x) ⇒
        {
            return x * 1.18
        }
log(gst_c)
```

* reduce():-

→ It is an higher order function which will take 2 arguments i.e callback function and accumulator (0 or 1).
(add, multiplication)

Syntax:- reduce(callback, 0/1);

Date __/__/____
Page _____

→ var total_price = gst_c.reduce((a,v)=>{
return a+v;
},0);
log(total_price);

O/P → 708

* Object :-

→ Object is a non-premetive datatype which is mutable in nature.

→ In JS object is a stand alone entity with properties and types.

→ In JS object we store the data in key and value format.

→ We can declare JS object in 2 ways :-
i) literal way
ii) By invoking the constructor using new keyword

i) literal way :-

→ Syntax :- var-type obj-name = {
key : "value",
key : "value",
key : "value"
}

Date __/__/____
Page _____

Ex:-

```
var S =
{
    name : "ABC",
    age : 18,
    mob : 9234612
}
log(S) // Prints all obj values
log(S.name) // Prints name only
```

→ Write a JS code to declare student object.

Sol:-

```
var student =
{
    name : "PQR",
    age : 20,
    id : 01,
    address : "Kolkata",
    Mob_no : 123456789
}
log(student)
```

→ To reinitialize a record in object we use below syntax :-

```
Obj_name.key = 'new_value';
```

ex:-

```
S.age = 20;
log(S) // prints updated obj
```

Date __/__/____
Page _____

→ To insert a new record into an existing object we use the syntax

·Syntax:

```
obj_name.new_key = new_value
```

ex:

```
s.id = 5
log(s) //prints the updated value along
        with new record.
```

ii) By invoking the constructor using new keyword:

→

```
let obj = new Object();
obj.name = 'ABC'
obj.id = 1234
obj.address = 'xyz'
log(Obj);
```

→ We can use an array as the value for an Object key.

ex:-

```
var bike={
        name:"Rx 100",
        milage:'35 km',
        color:['red','white','black'],
        price:65000
        }
log(bike);
```

Date __/__/____
Page _____

→ We can use a function as a value for an object key

```
var bike = {
    name: 'Rx 100',
    milage: 35,
    color: ['red', 'black', 'white'],
    price: function () {
        log('hellooo');
        log(this);
    }
}
bike.price();
```

→ We can use an object as a value for one more object key (Nested Object).

```
var fruit = {
    mango: 80,
    pineapple: 30,
    watermelon: 50,
    cherry: {
        price: 200,
        color: 'red',
        size: "small"
    }
}
log(fruit) // print the whole object along with nested object
```

Date __/__/____
Page _____

* Object destructuring :-

→ The process of converting object records into variables.

Syntax :-

var {type} {key1, key2, ...} = obj_name;

Ex :-

```
var fruit = {
    name: 'mango',
    id: 12,
    price: 40
}
let {name} = fruit;
log(name);
```

* Inbuilt functions of Object :-

i) Object.keys()

→ It is used to fetch all the keys of an object in the form of array.

```
var res1 = Object.keys(fruit);
```

Date __/__/____
Page _____

ii) Object.values()

→ It is used to fetch all the values of an object in the form of array.

```
var res2 = Object.values(fruit);
```

iii) Object.entries()

→ It is used to fetch all the keys as well as values of an object in the form of 2-D array or Nested array.

```
var res3 = Object.entries(fruit);
```

iv) Object.seal()

→ It is used to seal the object where we cannot insert the new record but we can update the existing record.

```
Object.seal(fruit);
```

v) Object.freeze()

→ It is used to freeze the object where we cannot insert new records as well as we cannot update the existing records.

Date __/__/____
Page _____

Object.freeze(fruit);

vi) Object.isSealed()

→ It is used to check whether object is sealed or not.

vii) Object.isFrozen()

→ It is used to check whether object is ~~frez~~ freezed or not.

* Inbuilt Objects of JS :-

→ We have multiple various kinds of inbuilt object of JS. Some of them are:-

i) Math
ii) Time
iii) Date
iv) Window

i) Math :-

a) Math.round()

→ It is used to round off a decimal number to the nearest non-decimal number.

Ex:- Math.round(99.76) // 100
Math.round(99.36) // 99

Date __/__/____
Page _____

b) Math.ceil()

⇒ It is used to round off a decimal value to the next number irrespective of decimal value.

c) Math.floor()

⇒ It is used to round off a decimal number to the same non-decimal number irrespective of decimal value.

d) Math.random()

⇒ It is used to develop random decimal numbers.

Ex: Math.random();
// print output between 0 & 1.

e) Math.sqrt()

⇒ It is used to find the square root of numbers.

f) Math.cbrt()

→ It is used to find cube root of a number.

Date / /
Page

ii) Time :-

⇒ Time object has 2 inbuilt functions :-
a) SetTimeout()
b) SetInterval()

a) SetTimeout() :-

⇒ It is a higher order function which will execute its callback function after specified time duration.

```
setTimeout(() => {
        log("Byee");
    }, 2000);
log("Hello");
```

O/P
Hello
Byee

b) SetInterval() :-

⇒ It is a higher order function which will execute its callback function for every specified time duration.

```
setInterval(() => {
        log("hi")
    }, 1000)
```

Date __/__/____
Page _____

→ Set Interval executes till infinity.

iii) Date :-

```
var d = new Date()
d.getFullYear()  2023
d.getMonth()  9
d.getDay()  1
d.getDate()  30
d.getHours()  18
d.getMinutes()  37
d.getSeconds()  40
d.getMiliseconds()  628
```

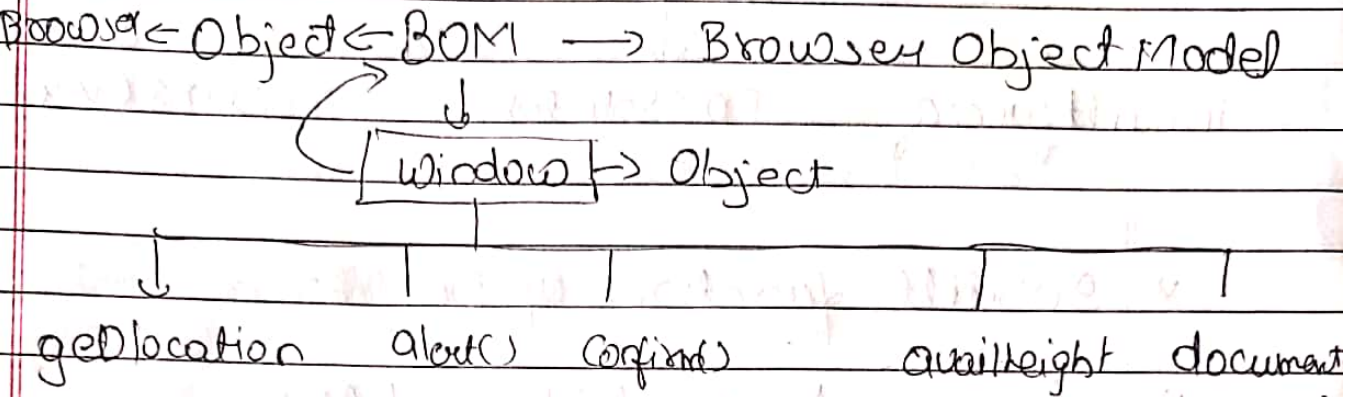
* Interpolation :-

→ It is a way of printing or utilizing variable using the syntax :-

```
${var}
```

Date __/__/____
Page _____

* BOM :-



→ BOM stands for Browser object Model.

→ It is an inbuilt object which is present in Browser.

* Window :-

→ Window is an inbuilt object which is present inside BOM.

* DOM :-

→ DOM stands for Document object Model.

→ It is an inbuilt object which is present in window object. DOM can handle entire html document.

Date __/__/____
Page _____

DOM

- inbuilt func of
- DOM Selectors
- DOM Events

* Inbuilt functions of DOM :-

i) document.createElement() :-

→ It is used to create instance of a specified HTML element.

Ex :- var a = document.createElement('div')
log(a); //

ii) innerText = ' '

→ It is a property which is used to insert some text into the HTML element.

a.innerText = "Hi";
log(a); // Hi

iii) appendChild()

→ It is used to insert an HTML element inside another HTML element as child.

Date __/__/____
Page _____

```
var div = document.createElement('div');
var h1 = document.createElement('h1');
h1.innerText = 'BYEE';
div.appendChild(h1);
document.body.appendChild(div);
```

iv) setAttribute ('', '')

→ It is used to set an attribute for an HTML element.

```
div.setAttribute('id', 'a1')
log(div); // <div id = a1> </div>
```

v) innerHTML

→ It is used to insert some HTML code inside an HTML element.

```
var div = document.createElement('div');
div.innerHTML = '<h1 id= "a1"> Hello </h1>';
log(div);
```

O/P

```
<div>
    <h1 id="a1"> Hello </h1>
</div>
```

Date __/__/____
Page _____

vi) Style

→ It is used to provide CSS stylings for in JS file for HTML elements in JS file.

```
var div = document.createElement('div')
div.style.border = '1px solid black'
div.style.height = '20px'
div.style.width = '20px'
document.body.appendChild(div)
```

* DOM Selectors :-

→ In JS before providing dynamic functionality for the HTML we have to select the element it is done by DOM selectors.

i) document.getElementById('id.value')

→ It is used to select an HTML element based on id.

→ Return type of getElementById() is object.

Date __/__/____
Page _____

ii) document.getElementsByTagName()

→ It is used to select multiple HTML elements based on Tag name.

→ Return type of getElementsByTagName() is HTML collection. (impure array)

→ To convert from HTML collection to pure array we use :-
Array.from(var_name)

iii) document.getElementsByClassName()

→ It is used to select multiple HTML elements based on class attribute value.

→ Return type- HTML collection

ex:- var a = document.getElementByClassName('a')
var res = Array.from(a);

iv) document.querySelector()

→ It is used to select an HTML element based on the argument passed.

→ In this selector we can pass id, class and tag name as the argument.

Date __/__/____
Page _____

→ When querySelector is matching with multiple elements it will return first matching element.

ex:- var res = document.querySelector('#pt')
log(res)

→ Return type is Object

v) document.querySelectorAll()

→ It is used to select multiple HTML element based on argument passed.

→ Return type is nodelist.

ex:- var res = document.querySelectorAll('.a');
log(res);
var res1 = Array.from(res)

Date / /
Page

* DOM Events :-

→ We have various types of DOM events: Whenever the user performs any actions in the UI, DOM events will occurs.

1) onClick Event
2) Mouseover
3) Mouseout
4) keydown
5) Key up
6) Onchange
7) onsubmit
8) dbl click

* onClick Event :-

→ We can handle the event in 3 ways :-

① onClick as an attribute

ex :-

```
<button onclick=test()>click me</button>
```

HTML

```
function test()
{
document.body.style.background color = 'red';
}
```

JS

Date __/__/____
Page _____

(2) On Click as Property

→

```
<button>
Click ME
</button>
```

html

```
var b= function()
{
  var n1= Math.round(Math.
    random()*255);
  var n2= Math.round(Math.
    random()*255);
  var n3= Math.round(Math.
    random()*255);
  document.body.style.background
  Color=`rgb(${n1},${n2},${n3});
}
var a= document.querySelector
        ('button')
  a.click=b
```

JS

(3) Using addEventListner()

```
var btn = document.getElementById
          ('btn');
  btn.addEventListner('click', () =>
{
  document.body.style.background
  Color = 'red'
});
```

→ addEventlistner is a higher order function.
addEventListner('event', callback)

Date __/__/____
Page _____

Q:- What is addEventListner ?

Ans:- It is a higher order function which is used to handle DOM events.

→ It takes 2 arguments event and callback function.

→ Whenever the user performs specified event callback function will be invoked.

* How to fetch the data from Textfield ?

html
```
<input type="text" id="bay">
<button> Submit </button>
```

js
```
var a = document.querySelector('#bay')
var b = document.querySelector('button')
b.addEventListener('click', ()=>
{
    var vl = a.value
    log(vl)
})
```

Date __/__/____
Page _____

* How to handle data in form?

→ Whenever we submit the form it will try to establish the connection to the database by default.

→ To handle this default behaviour of form we use preventDefault().

→ html

```
<form>
    <input type="text">
    <button>Click Me</button>
</form>
```

js

```
var a = document.querySelector('input')
var b = document.querySelector('button')
b.addEventListner('click', (e) =>
{
    e.preventDefault()
    log(a.value);
});
```

Date __/__/____
Page _____

* Mouseover :-

⇒ When the user performs mouse hovering actions in the UI mouseover will occure.

* Mouse out :-

→ When the user remove the mouse cursor from a particular container mouse out event will be occured.

ex :- html

```
<div id="pt">
    <div id="a"></div>
    <div id="b"></div>
</div>
```

js

```
var a = document.getElementById('a')
var b = document.getElementById('b')
a.addEventListener('mouseover', () => {
  b.style.backgroundImage = 'url(link)';
})
a.addEventListener('mouseout', () =>
{
    b.style.backgroundImage = '';
})
```

Date __/__/____
Page _____

* Keydown :-
Keypress
keyup :-

ex :-

html -

```
<input type: "text" id = "a" >
<input type = "text" id = "b" >

var a = document.getElementById('a')
var b = document.getElementById('b').

a.addEventListner('keydocon', () =>
{
    log("Key Dacned");
})
a.addEventListner('keypress', () =>
{
    log("Key Pressed");
})
a.addEventlisther('keyup', () =>
{
    log("key upped");
})
```

Date __/__/____
Page _____

```
a.addEventListener('keyup', () =>
{
    var vL = a.value
    b.value = `${vL}`
})
b.addEventListener('keyup', () =>
{
    var v1 = b.value
    a.value = `${v1}`
})
```

To uppercase :-

```
a.addEventListener('keyup', () =>
{
    var v1 = a.value
    var v2 = v1.toUpperCase()
    a.value = `${v2}`;
})
```

Date __/__/____
Page _____

* onSubmit :-

=> It is an event which will occur when a user submit the form.

=> We have to target the form tag for this event.

html :-

```
<form>
  <input type="text" id="a1"> <br>
  <input type="password" id="a2"> <br>
  <input type="submit" value="login">
</form>
```

js :-

```
var ft = document.querySelector('form')
ft.addEventListner('submit', (e) =>
  {
    e.preventDefault();
    log('form is submitted');
  })
```

Date __/__/____
Page _____

* onChange :-

→ When the user changes the control from one element to another element onchange event will occur.

→ We have to target the text field:

* Event Propogation :-

→ Whenever user performs any action or event in the UI event propogation will occur.

→ Event propogation have 3 stages :-

1) Event Capture
2) Event Target
3) Event bubbling:

Date __/__/____
Page ____

It should be some event in parent elm.

1) Event Capture :- It is a process where control travels from root element (html) to the specific element where event is occured. (Top to bottom).

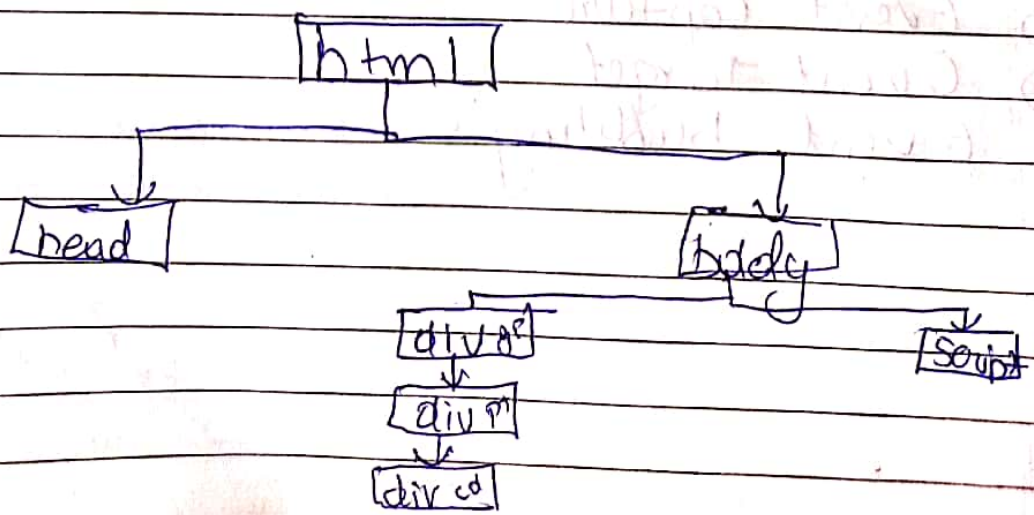
2) Event Target :- Executing or Triggering JS functionality is called Event Target.

3) Event Bubbling :- After event target controls travels back to the root element from bottom to top.

→ When bubbling happening if there is any event in the parents elements they will be triggered. To avoid this we use :-

stopPropogation() or
StopImmediatePropogation()

DOM Tree



Date __/__/____
Page _____

* Module :-

→ Module allows developer to use multiple javascript files where we can store the js code in organized way.

→ Modules are classified into two types
i) Common js module (react, angular, vue.js etc).
ii) ES6 module

i) Common js module :-

⇒ We use common js module when we want to attach or import the js libraries such as react, angular, vue.js etc.

ii) ES6 module :-

⇒ We use ES6 module when we want to attach or import a js file into another js file.

⇒ Rules of ES6 module :-

i) The file extension of js should be .mjs

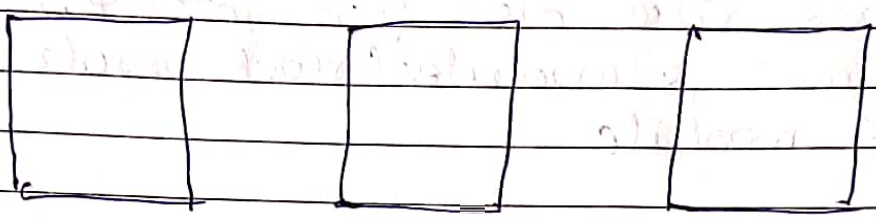
ii) We have to use type="module" in the script tag.

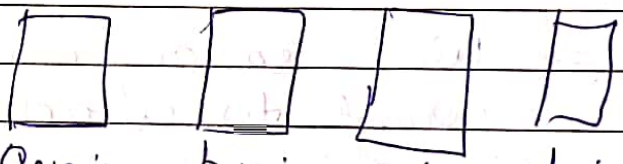
Date __/__/____
Page _____

iii) Properties should be exported

iv) exported properties should be imported

v) Framework should be executed atleast in a local server.





* JSON :-

=> JSON stands for Java script object Notation.

-> It is a file format to store transporting or exchanging data.

-> In JSON file we store the data in key and value pairs.

-> The file extension of JSON file should be .json.

Date / /
Page

→ JSON is language independents.

→ JSON is derived from JS object.

→ In JSON we can use the below datatypes :-

1) String
2) Number
3) Boolean
4) Null
5) Array
6) Object

→ Syntax :-

```
{
    "key" : "value",
    "key" : "value",
    "key" : "value",
    "key" : "value"
}
```

Ex:-

```
{
    "name" : "ABC",
    "id" : 1,
    "email" : "abc@123"
}
```

200 - 299 - Successful
400 - 499 - User side Error
500 - 599 - Server Side Error.

Date __/__/____
Page _____

* Promise :-

→ Promise is used to handle asynchronus operation.

→ Promise has 3 states :-
i) Pending State
ii) Resolve or successful state
iii) Reject or unsuccessful state

→ i) Pending State

→ When the user send the request to the Server, request is in process this situation is called pending state.

ii) Resolve or Successful State

→ When the user send the request to the server, server give the response as expected this situation is called resolve or successful state.

iii) Reject or unsuccessful state

→ When the user send a request to the server, server not responded due to some mistake (user side mistake or

json-server filename.json

Date __/__/____
Page _____

resource means piece of data.

Server side mistake) this situation is called Reject or unsuccessful state.

* then(()=>{})

→ It is a higher order function which points to the resource fetched by JSON. then() will execute when the promise meets resolve state.

* catch(()=>{})

→ It is a higher order function which points to the details about request fail. Catch() will execute when promise meets reject state.

* window.fetch("...")

→ It is a method belongs to window object which is used to resource using URI.

Syntax :-

```
window.fetch("http://localhost:3000/employees").
then((d)=>d.json()).then((a)=> console.
log(a)).catch((e)=>{console.log(e)})
```