

HTML, CSS & JavaScript

Created Date : 12/09/2022
Created By : GOVE ACADEMY

Version History



S. No	Revision Date	Version	Modifications	Modified By

Table of Contents



1.Web Application Introduction

2.HTML

2.1.Tags

2.2 Attributes

2.3 Form Elements

3.CSS

3.1 Introduction

3.2. Container

3.3. Types

4. JavaScript

4.1 Introduction

4.2. Synchronous vs Asynchronous

4.3. variables

4.4. Rules for declaring variables

4.5. mutable and immutable in JS

4.6. Statements

4.7. Block

4.8. Function

4.9. Callback Function

4.10. Form Elements

4.11. Displaying values in JavaScript

4.12. Array

4.13. Redeclare variable

4.14. Object

4.15. Operators

4.15.1. Arithmetic operator

4.15.2. Assignment operator

4.15.3. Comparison operator

4.15.4. Logical operator

4.15.5. Conditional operator

4.16. Debugging

4.16.1. Exception Handling

4.16.2. Exception Handling Statements

4.16.3. Debugger

4.17. Type Conversion

4.18. Break & Continue Statement

4.19. JavaScript arrow function

4.20. Class

4.21. JavaScript Module

4.22. Regular expression

**4.23. String Method , Date Object, Math
Object**

4.24. Event Handler

4.25. JavaScript strict mode

4.26. JavaScript Best practices

4.27. JavaScript Scope

4.28. JSON

1. Web Application Introduction

- ❑ A Web application is an application program that is stored on a server and delivered over the Internet through a browser interface.
- ❑ To Develop a web Application, following web programming languages are used
 - HTML
 - CSS
 - JavaScript

HTML



- HTML stands for **H**ypertext **M**arkup **L**anguage.
- A Markup Language is a set of Markup Tags.
- Structured content is displayed to browsers using HTML.

CSS

- CSS stands for **C**ascading **S**tyle **S**heet
- CSS is used to describe the presentation of the document written in HTML.
- CSS3 is the latest version ,new functionalities were introduced in the CSS3

JavaScript

- JavaScript is a scripting language, used for client-side web development
- To create interactive user interface in a web page
- Manipulating web content Dynamically.

HYPER **T**EXT **M**ARKUP **L**ANGUAGE

2. HTML

- ❑ Html is a static web programming language, and its standards are defined by W3C- World wide web Consortium.
- ❑ Html have pre-defined elements which combined makes the webpage.
- ❑ It controls the web page and will give look and feel-good experience to the users.

HTML Structure

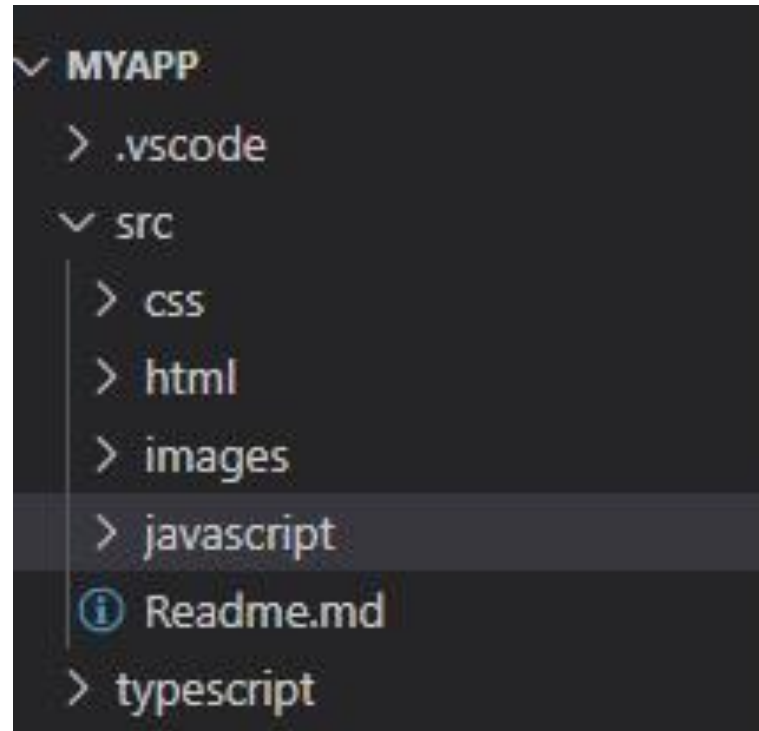
- ❑ HTML have some pre-defined structure to be followed.

```
src > html > html > <> sample.html > html > body
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Title</title>
5  </head>
6  <body>
7  ...Content
8  </body>
9  </html>
```

Display tags are
used inside the
body tag

Folder Structure

Basic folder structure where we had organized all our HTML , CSS and JavaScript files.



2.1. Tags

HTML tags defines the format if the web browser and displays the content. HTML tags contain three main parts: opening tag, content and closing tag

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

- All HTML tags must be enclosed within < > these brackets.
- Every tag in HTML perform different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

2.1.1. Heading Tag

- HTML headings are defined with <h1> to <h6> Tags
- Heading tags is used based on the priority

Sample Code

```
<!DOCTYPE html>
<html>
<head>
  <title>Headings</title>
</head>
<body>
  <!--Display Content must be inside the body tag-->
  <h1>Gove Enterprises</h1>
  <h2>Gove Enterprises</h2>
  <h3>Gove Enterprises</h3>
  <h4>Gove Enterprises</h4>
  <h5>Gove Enterprises</h5>
  <h6>Gove Enterprises</h6>
</body>
</html>
```

Output:

Gove Enterprises

Gove Enterprises

Gove Enterprises

Gove Enterprises

Gove Enterprises

Gove Enterprises

2.1.2. <p>

- HTML paragraph are defined with <p> Tag
- Every new paragraph is start with <p> tag and ends with </p>

Symbol Code

```
<!DOCTYPE html>
<html>
<head>
  <title>Headings</title>
</head>
<body>
  <!--Display paragraph must be inside the body tag-->
  <p>Think Positive ; Be Positive; Talk Positive</p>
</body>
</html>
```

Output

Think Positive ; Be Positive; Talk Positive

2.1.3.

- HTML Images are defined with the tag.
- Image tag is used to insert image in web site

```
html > check > image.html > ...  
<!DOCTYPE html>  
<html>  
<head>  
  <title>Headings</title>  
</head>  
<body>  
  <h1>Gove Enterprises</h1>  
  <!--Display image must be inside the body tag-->  
    
</body>  
</html>
```

Output:

Gove Enterprises



Images can be imported into html tags in three different ways

Relational path.

Relational path is the place where the file is in the current directory.

```

```

Absolute path.

An absolute path refers to the complete details needed to locate a file or folder, starting from the root element and ending with the other subdirectories .

```

```

Web path.

By using web path, we can import pictures from online servers and can be used in img tag

```

```

Web path vs Relative path vs Absolute path

Below is the sample program how the images are imported using this three methods.

```
<!DOCTYPE html>
<head>
  <!-- define Title for HTML Page -->
  <title>HTML Training Material</title>
</head>

<body>
  <div>
    <!-- Web Path -->
    <h3>Web Path</h3>
    
    </div>
  <div>
    <!-- Relative Path -->
    <h3>Relative Path</h3>
    
    </div>
  <div>
    <!-- Absolute Path -->
    <h3>Absolute Path</h3>
    
    </div>
</body>
</html>
```

Web Path



Relative Path



Absolute Path



Incorrect format using image src

File Format


When the path/File format defined is not valid, the text which given in the img tag alt attribute would display.

Sample code

```
<!DOCTYPE html>
<html>
<head>
  <title>Headings</title>
</head>
<body>
  <h1>Gove Enterprises</h1>
  <!--Display image must be inside the body tag-->
  
</body>
</html>
```

Output:

Gove Enterprises

image

2.1.4. <a>

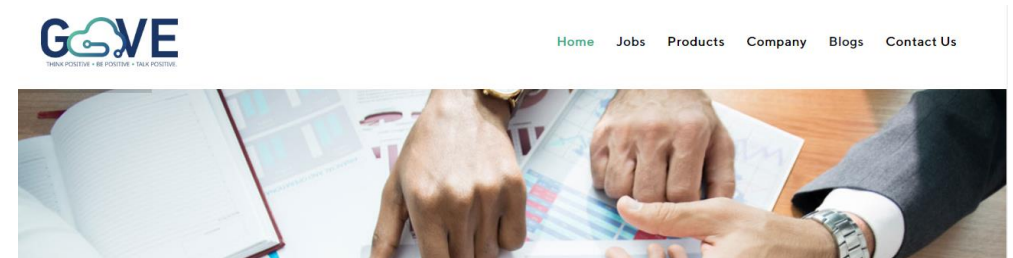
- <a> tag is used for hyperlink
- which is used to link from one page to another.

Sample Code

```
<!--Hyperlink tag<a></a>-->  
<a href="https://gove.co/">Gove Enterprise</a>
```

Output:

[Gove Enterprise](https://gove.co/)



2.1.5.

- A Line Break is used to start a new line without starting a new paragraph.

Sample Code

```
<!--break Tag-->
<p>
  Gove is a simplified next generation cloud business platform for modern innovative
  manufacturers,<br> with the primary focus on user experience and performance management for complex
  workflow</p>
```

Output:

Gove is a simplified next generation cloud business platform for modern innovative manufacturers,
with the primary focus on user experience and performance management for complex workflow

2.1.6. <hr>

- ❑ <hr> tag is the Horizontal rule.
- ❑ Display the horizontal line that is used to separate the content in a HTML Webpage.

Sample Code

```
<!--hr Tag-->  
<p>  
Gove is a simplified next generation cloud business platform for modern innovative  
manufacturers,<br> with the primary focus on user experience and performance management for complex  
workflow</p>  
<hr>
```

Output

Gove is a simplified next generation cloud business platform for modern innovative manufacturers,
with the primary focus on user experience and performance management for complex workflow

2.1.7.

 Tag is used for formatting the output as Bold.

Sample Code

```
<!--Bold Tag-->
<p>
  Gove is a simplified next generation <b>cloud business platform </b>for modern innovative
  manufacturers,<br> with the primary focus on user experience and performance management for complex
  workflow</p>
```

Output

Gove is a simplified next generation **cloud business platform** for modern innovative manufacturers,
with the primary focus on user experience and performance management for complex workflow

2.1.8. <i>

Italic Tag <i> display the content in italic format.

Sample Code

```
<!--Italic Tag-->  
<p>  
  Gove is a simplified next generation cloud business platform for modern innovative  
  manufacturers,<br> with the <i>primary focus on user experience </i>and performance management for complex  
  workflow</p>
```

Output

Gove is a simplified next generation cloud business platform for modern innovative manufacturers,
with the *primary focus on user experience* and performance management for complex workflow

2.1.9.

- ❑ tag defines the Important text in HTML

Strong tag will basically highlight the text

Sample Code

```
<p> Gove <strong>Enterprise</strong> Tirunelveli</p>
```

Output

Gove **Enterprise** Tirunelveli

2.1.10.

- tag is used to display the content in italic format

Sample Code

```
<p> Think    <em>Positive;</em> Be Positive; Talk Positive</p>
```

Output

Think *Positive*; Be Positive; Talk Positive

2.1.11. <mark>

Mark Tag is used to define the content Marked/highlighted

Sample Code

```
<!--mark tag-->
<p>
  Gove is a simplified <mark>next generation cloud business platform </mark>for modern innovative
  manufacturers,<br> with the <i>primary focus on user experience </i>and performance management for complex
  workflow</p>
```

Output

Gove is a simplified **next generation cloud business platform** for modern innovative manufacturers,
with the *primary focus on user experience* and performance management for complex workflow

2.1.12. <small>

<small> tag defines the smaller text of the HTML Content

Sample Code

```
<!--small tag-->  
<p>Gove helps ambitious <small></small>entrepreneurs all</small> over the world.</p>
```

Output

Gove helps ambitious entrepreneurs all over the world.

2.1.13.

Delete tag defines the deletion on the HTML content, which results in content being struck out.

Sample Code

```
<!--Delete and Insert tag-->  
<p> Honesty is the <del>good</del><ins>best</ins> policy</p>
```

Output

Honesty is the ~~good~~best policy

2.1.14. <ins>

<ins> tag is used to insert a content into a document

Sample Code

```
<!--Insert Tag-->  
<p> Gove is a simplified next <ins>generation cloud business </ins> platform for modern innovative </p>
```

Output

Gove is a simplified next generation cloud business platform for modern innovative

Recap:

- HTML is the structured content is displayed to browsers using HTML.
- CSS is used to describe the presentation of the document written in HTML.
- JavaScript is used to create interactive user interface in a web page
- HTML tags defines the format
- Heading <h1> to <h6> Tags
- <p>
-
- <a>
-

-
- <ins>
- <I>
-
-
- <mark>
- <small>
-

2.1.15. <sub>

Subscript text appears half of a content below the normal line.

Sample Code

```
<p> H<sub>2</sub>O is the chemical symbol for water;</p>
```

Output

H₂O is the chemical symbol for water;

2.1.16. <sup>

Superscript text appears half of a content above the normal line.

Sample Code

```
<p>Mathematical Formula : ( a + b ) <sup>2</sup> = a<sup>2</sup> + b <sup>2</sup> + 2ab</p>
```

Output

Mathematical Formula : (a + b) ² = a² + b ² + 2ab

2.1.17.

The span tag ` ` is used to apply CSS to a specific area, like using inline css

Sample Code

```
<!--Super Tag and -->
<p> <span style = "color: red;">Gove Enterprises</span> : Think Positive; Be Positive ; Talk Positive;</p>
```

Output

Gove Enterprises : Think Positive; Be Positive ; Talk Positive;

2.1.18. <table>

```
<table border="1" style="text-align: center;">
  <tr>
    <th>Days</th>
    <th>1</th>
    <th>2</th>
    <th>3</th>
    <th>4</th>
    <th>5</th>
    <th>6</th>
    <th>7</th>
    <th>8</th>
  </tr>
  <tr>
    <th>Monday</th>
    <td>Eng</td>
    <td colspan="2">Math</td>
    <td>Chemistry</td>
    <td rowspan="3">Break</td>
    <td>Physics</td>
    <td colspan="2">Physics lab</td>
  </tr>
  <tr>
    <th>Tuesday</th>
    <td>Maths</td>
    <td>Physics</td>
    <td>English</td>
    <td>Chemistry</td>
    <td colspan="2">Computer Science</td>
    <td>Chemistry lab</td>
  </tr>
  <tr>
    <th>Wednesday</th>
    <td>Physics</td>
    <td>maths</td>
    <td colspan="2">Chemistry lab</td>
    <td colspan="2">Physics lab</td>
    <td>Computer lab</td>
  </tr>
</table>
```

Sample Code

- Table tag is used to define the table .
- Create table using <tr>- table row, <th> - table head, <td> - table data
- <table> table tag is starting tag for creating a table structure
- <tr>- tr means table row. - Every table is based on the row and columns
- <th>- th means table header –this tags will bold the text

Days	1	2	3	4	5	6	7	8
Monday	Eng	Math		Chemistry	Break	Physics	Physics lab	
Tuesday	Maths	Physics	English	Chemistry		Computer Science	Chemistry lab	
Wednesday	Physics	maths	Chemistry lab			Physics lab	Computer lab	

Output

2.1.19.

- ❑ and
- ❑ Ul – unorderedlist tag is used to list the items with bullet points
- ❑ Li - list the item using items tag

Sample Code

```
<!--Gove Enterprise Team-->
<h3>Gove Teams :</h3>
<ul>
  <li>Training Team</li>
  <li>product Team</li>
  <li>Development Team</li>
  <li>Testing Team</li>
  <li> Operation Team </li>
  <li>Finance Team</li>
</ul>
```

Output

Gove Teams :

- Training Team
- product Team
- Development Team
- Testing Team
- Operation Team
- Finance Team

2.1.20.

- ❖
- ❖ order list tag is used to list the items with Numbering
- ❖ e.g. : we listed the css style using orderlist
Items listed in numbering

Sample Code

```
<h2>Types of style</h2>  
<p>CSS stands for Cascading Style Sheets</p>  
<ol class="css types">  
  <li>Inline CSS</li>  
  <li>Internal CSS</li>  
  <li>--External CSS</li>  
</ol>
```

Output

Types of style

CSS stands for Cascading Style Sheets

1. Inline CSS
2. Internal CSS
3. --External CSS

2.1.21. <link>

- ❖ <link></link>
- ❖ Link tag is used in head position to import external files
- ❖ Imported sample.css file using href attributes

```
<head>  
  <!-- Import CSS -->  
  <link rel="stylesheet" href=" ../css/sample.css" >  
  
  <!-- Internal Script -->  
  <script type="text/javascript">
```

2.1.22. <script>

Script tag is used to write JavaScript inside an html page which contains Scripting statements

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript String Methods</h2>
<p>Replace "gove" with "GOVE" in the paragraph below:</p>
<button onclick="myFunction()">Try it</button>
<p id="demo">Welcme to GOVE</p>
<script>
function myFunction() {
  let text = document.getElementById("demo").innerHTML;
  document.getElementById("demo").innerHTML =
    text.replace(/gove/i, "gove");
}
</script>
</body>
</html>
```



JavaScript String Methods

Replace "gove" with "GOVE" in the paragraph below:

Try it

Welcme to GOVE

Recap:

HTML Tags

- `<sub>`
- `<sup>`
- ``
- `<tr>`
- `<th>`
- `<td>`
- ``
- ``
- `<link>`
- `<script>`

2.2. HTML Attributes

Attributes provide additional information about elements. Attributes usually come in name/value pairs

- ❑ **href** - hypertext reference is used to specifies the url using to <a> <link>
- ❑ **src** - source attributes is used to mentioned the source to get the data and access
- ❑ **alt** – Alt stands for alternate and is used in image tags to show alternative text for an image.
- ❑ **Colspan** - To combine the columns for table access, colspan is used in the table tag.
- ❑ **Rowspan** -To combine the rows for table access, rowspan is used in the table tag.
- ❑ **Height** - The height attribute is used to specify the height of the element.
- ❑ **Width** - The width attributes defines the width of the element.
- ❑ **Style** - The style attribute is specifying an inline style of an element .
- ❑ **Id** – is used to specify unique id for an HTML element
- ❑ **Class** - the class attribute is used to apply CSS for an element.

Id and class

- Id
 - Id is a unique identifier assigned to an HTML element
 - It used to refere to the element in CSS
 - Every element have a unique id
 - Id is used to identify the element

```
<style>

#w1{
  color: ■ green
}
w2{
  color: ■ purple
}
#w4{
  background-color: ■ blue;
}

</style>
```

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

```
<h1 id="w1">Hello World</h1>
<h2 id="w2">Hello World</h2>
<h3 id="w3">Hello World</h3>
<h4 id="w4">Hello World</h4>
<h5 id="w5">Hello World</h5>
<h6 id="w6">Hello World</h6>
```

- Class
 - Class is a non unique identifier assigned to HTML tag
 - It can be used to group elements together and apply the same style
 - Classes shared across the multiple elements

```
<div class="header">  
  <h1>Hello World</h1>  
  <h2>Hello World</h2>  
  <h3>Hello World</h3>  
  <h4>Hello World</h4>  
  <h5>Hello World</h5>  
  <h6>Hello World</h6>  
</div>
```



```
<style>  
  .header {  
    color: red;  
    background-color: aqua;  
  }  
</style>
```


2.3. HTML Form Elements



HTML Forms are required, when you want to collect some data from the end-users. For example, during user registration you would like to collect information such as name, email address, etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form.

- ❖ Text Input Controls
- ❖ Checkboxes Controls
- ❖ Radio Box Controls
- ❖ Select Box Controls
- ❖ File Select boxes
- ❖ Hidden Controls
- ❖ Clickable Buttons
- ❖ Submit and Reset Button

2.3.1 Text Input Controls:

There are three types of text input used on forms

❖ Single-line text input controls:

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.

❖ Password input controls:

This is also a single-line text input, but it masks the character as soon as a user enters it. They are also created using HTML **<input>** tag.

❖ Multi-line text input controls:

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

```
<form>
  Name : <input type="text" class="clsinputName" id="txtinputName" name="inputName" /><br />
  Password : <input type="password" class="clsinputpwd" id="txtinputpwd" name="inputPwd" /><br />
  Address : <textarea id="txtAddress" class="clstxtAddress"></textarea><br />
  <button id="btnSubmit" onclick="getelement()">Click</button>
</form>
```

Name :

Password :

Address :

2.3.2 Checkbox Control :

Checkboxes are used when more than one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to **checkbox**.

```
<div>
  <form>
    <br />
    <span>Checkbox Controls</span><br />
    <input type="checkbox" name="mongo" value="on"> Mongo DB<br />
    <input type="checkbox" name="express"> Express<br />
    <input type="checkbox" name="react"> React JS<br />
    <input type="checkbox" name="node"> Node JS<br />
  </form>
</div>
```

Checkbox Controls

- ☒ Mongo DB
- ☐ Express
- ☒ React JS
- ☒ Node JS

2.3.3 Radio Button Control :

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to **radio**.

```
<div>
  <form>
    <br />
    <span>RadioBox Control</span><br />
    <input type="radio" name="group" checked="checked"> True
    <input type="radio" name="group"> False<br />
  </form>
</div>
```

RadioBox Control

☐ True ☒ False

2.3.4 Select Box Control :

A select box, also called drop down box which provides option to list down various options in the form of dropdown list, from where a user can select an option.

```
<div>
  <form>
    <br />
    <span>SelectBox Control</span><br />
    <select name="dropdown">
      <option value="--Select--" selected>--Select--</option>
      <option value="Mongo DB">Mongo DB</option>
      <option value="Express">Express</option>
      <option value="React JS">React JS</option>
      <option value="Node JS">Node JS</option>
    </select>
  </form>
</div>
```

SelectBox Control

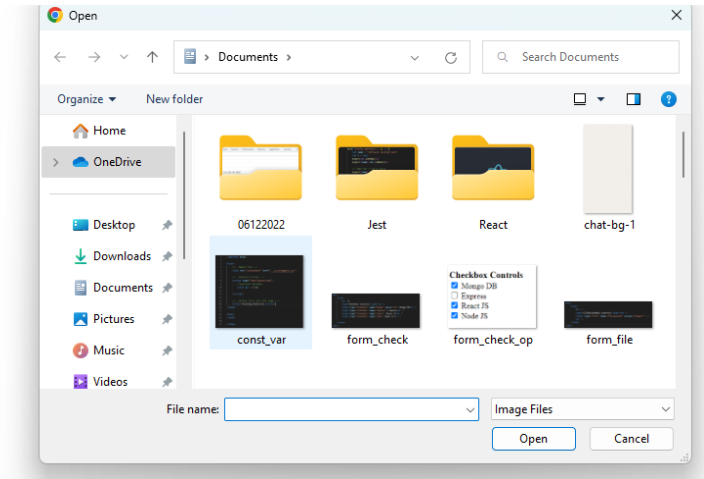
--Select--
 --Select--
 Mongo DB
 Express
 React JS
 Node JS

2.3.5 File Upload Box :

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the `<input>` element but type attribute is set to **file**.

```
<div>
  <form>
    <br />
    <span>FileUploadBox Control</span><br />
    <input type="file" name="fileupload" accept="image/*" />
    <br />
  </form>
</div>
```

FileUploadBox Control
Choose File No file chosen



2.3.6 Button Controls :

There are various ways in HTML to create clickable buttons. You can also create a clickable button using `<input>` tag by setting its type attribute to **button**.

```
<div>
  <form>
    <br />
    <span>Button Controls</span><br />
    <input type="submit" name="submit" value="Submit" /><br />
    <input type="reset" name="reset" value="Reset" /><br />
    <input type="button" name="ok" value="OK" /><br />
    <input type="image" name="imagebutton" src="../img/gove_logo.jpg" alt="GOVE" height="50" width="50" /><br />
  </form>
</div>
```

Button Controls

Submit
Reset
OK



2.3.7 Hidden Form Controls :

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page.

```
<div>  
  <input type="hidden" name="pagename" value="10" />  
</div>
```

Recap:

- Attributes provide additional information about elements
- href
- src
- alt
- Colspan
- Rowspan
- Height
- Width
- Style
- Id
- Class
- HTML form elements are used to create forms using HTML form controls.
- Text Input Controls – Single line, Password, Multiline.
- Checkboxes are used when more than one option is required
- Radio buttons are used when many options are required.
- A select box, also called drop down box.
- To upload a file to your web site - File Upload Box
- Button Controls - Create a clickable button

Think !



1. HTML stands for _____
2. CSS stands for Cascading Style Sheet & describe the _____ of the document.
3. Which language is named as scripting language ?
4. Elements are _____ tags.
5. What three parts make up a tag?
6. How do you represent a tag?
7. How to define the Heading tag ?
8. Uses of image tag?
9. If the file is in the current directory.
10. Complete details of the path , starting from the root element and ending with the other subdirectories is called ?

Option1: Relative path

Option2: absolute path

11. _____ path is used to import pictures from online servers.
12. When path/File format defined is not valid, the text which given in the img tag _____ attribute would display.

13. Syntax for inserting a link.
14. How to define the line break?
15. What is <hr> tag?
16. _____ is used for formatting the output as Bold.
17. <i> display the content in _____ format.
18. Which tag is used for Important text in HTML?
Option1: Strong
Option2: Bold
19. What tag defines the strike out on the HTML content?
20. The span tag is used to apply _____ to a specific area.
21. What all are the tags used to define a table?
22. Additional information about HTML elements are called _____
23. List some attributes in HTML?

CSS- Cascading Style Sheet

3. CSS

3.1. CSS Introduction

- CSS is used for the presentation of web pages that includes color, font and layout.
- It allows one to change itself for the presentation on different types of devices, such as larger screens and smaller screens
- If HTML were the engine components of a car, CSS would be the body style and the paint job.
- A website can run without CSS, but it certainly isn't pretty. CSS makes the front-end of a website shine and it creates a great user experience.
- Without CSS, websites would be less pleasing to the eye and likely much harder to navigate.

Problem faced before creating CSS

- Font, background color and style must be written on each line as if the web page is very big it is hard to write tags in each line.
- CSS style are saved in external CSS files, so it is easy to change the entire website by changing just the one file.
- There are many useful tags in CSS when compared to HTML

3.2. CSS container

- ❑ Container is used to define common Styles for a set of content in Web-Page.
- ❑ It Provides
 - Margins
 - Paddings
 - Alignment
 - Fonts
 - Colors
- ❑ For the specific html to get applied for the specific container we define a class name for each of the container

3.3. Types of CSS

- **Inline** - An inline CSS is used to apply a unique style to a single HTML element.

```
<li style="font-size: 18px;font-weight: bold;">Inline Style</li>
```

- **Internal** - An internal CSS is used to define a style for a single HTML page.

```
<style type="text/css">
  .internalLI {
    font-size: 18px;
    font-weight: bold;
  }
</style>
```

- **External** - The external style sheet is generally used when you want to make changes on multiple pages

```
span {
  font-size: 18px;
  font-weight: bold;
}

.externalLI {
  font-size: 18px;
  font-weight: bold;
}
```

Below mentioned is a sample code for how these css types can be used.

Sample Code

```
<!DOCTYPE html>

<head>
  <!-- define Title for HTML Page -->
  <link rel="stylesheet" href="../css/sample.css">
  <!-- Internal CSS -->
  <style type="text/css">
    .internalLI {
      font-size: 18px;
      font-weight: bold;
    }
  </style>
  <title>HTML Training Material</title>
</head>

<body>
  <p>
    Style can be used in 3 different ways:
  </p>
  <!-- Unordered List -->
  <ul>
    <li style="font-size: 18px;font-weight: bold;">Inline Style</li>
    <li class="internalLI">Internal Style</li>
    <li class="externalLI">External Style</li>
  </ul>
</body>

</html>
```



output



Recap:

- CSS is used for the presentation of web pages that includes color, font and layout.
- Font, background color and style must be written on each line
- CSS container - For the specific container, define a class name.
- Types – Inline, Internal and External.

Think !

1. CSS Stands for _____
2. _____ of web pages that includes color, font and layout.
3. _____ is used to define common Styles for a set of content in Web-Page.
4. We define a _____ for each container for the specific html to be applied for that specific container.
5. What are the types of CSS?
6. CSS is used to define a style for a single HTML page.
7. Functions of the CSS background?
8. Which property is used for background in CSS?
9. _____ property specifies whether the background image should scroll or be fixed
10. what is used to create space around elements, outside of any defined borders.
11. List out margin Property?

12. _____ is used to create space around an element's content, inside of any defined borders.

13. what property is used to specify all the padding properties in one property.

14. _____ property is used to set the color of the text.

15. How to align the last line of a text, Choose one from the Below:

Option1: text-align-last:

Option2: text-align-last:

Option3: text-align-last:

16. Which property is used to specify uppercase and lowercase letters in a text?

17. _____ property is used to specify the space between the characters in a text.

18. choose the one to specify the space between lines:

Option1: line-height:

Option2: Line-height:

Option3: line-height:

19. What property adds shadow to text?
20. _____ property to specify the font of a text.
21. List out the properties of font style.
22. The default text size in browsers is,

Option1:16px

Option1:.20px

Option1:10px

23. display: _____ allows to set a width and height on the element.
24. display: _____ add a line-break after the element, so the element can sit next to other elements.

Exercise 1:

Assume that you are owner of the Departmental Store, You need a webpage to attract the customers, In order that Create a Static Webpage for your Departmental Store using HTML and CSS.

Following things are to be considered while creating the Static webpage:

1. Apply HTML Tags such as: <h1>...<h6>, <p>, , <a>, ,
, <hr>, <table>, , , and <link> tag to import the CSS File.
2. Create a CSS File and apply it over the HTML page,
 - Body background
 - Image Border
 - Paragraph Alignment
 - Font style & Size
3. Create a HTML Form to collect the customer data for getting Feedbacks and Delivery purpose.
 - Name - Input Box
 - contact - Input Box
 - Gender - Radio Box
 - Favorite items - Select Box
 - Delivery Address - Text Area
 - Suggestions - Text Area
 - Submit Button.

JavaScript



4 JavaScript



4.1. Introduction

- ❖ JavaScript is a dynamic programming language.
- ❖ JavaScript makes the web page functional by performing the actions. It gives life to a static web page by making it functional.
- ❖ JavaScript is Case Sensitive programming language.
- ❖ JavaScript can be defined as client-side programming language.
- ❖ Node.js is a runtime environment for JavaScript. node.js is introduced java script is also used as server-side programming rather than client-side programming.

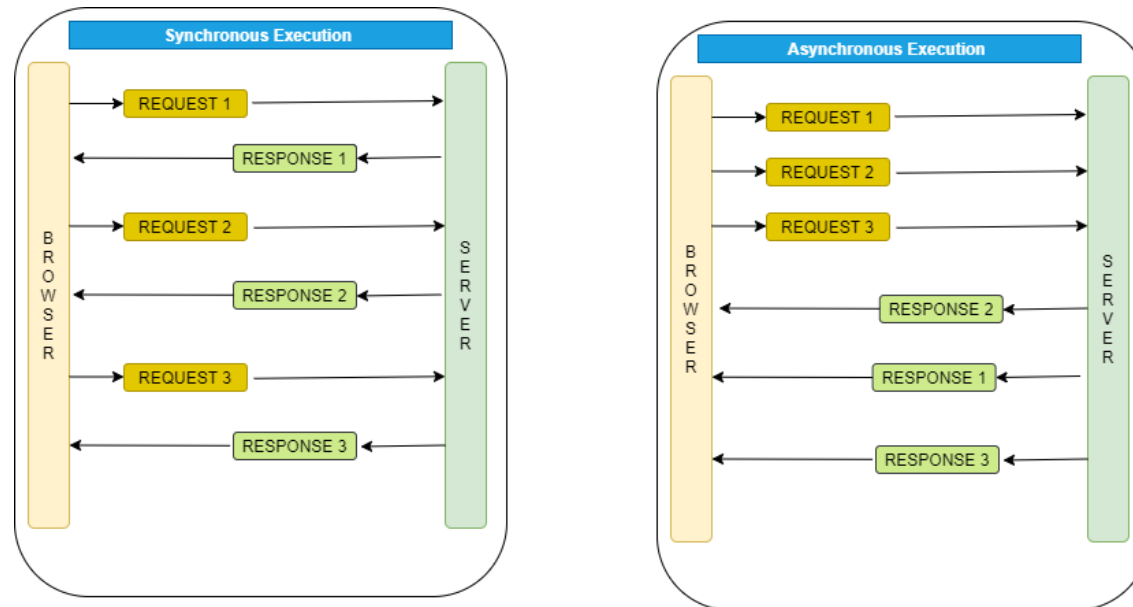
Node.js

- ❖ Node.js is an open-source run time environment for java script built on Google Chrome's JavaScript V8 Engine.
- ❖ Node.js allows us to run JavaScript on the server.

4.2. Synchronous vs Asynchronous

Synchronous (or sync) execute the code blocks in a sequence manner . In sync programming, the program is executed line by line, one line at a time. Each time a function is called, the program execution waits until that function returns before continuing to the next line of code.

Asynchronous (or async) execute the code blocks non sequential way. In async programming the program doesn't wait for the task to complete and can move on to the next task.



4.3. Variables



Variables are container for storing value and it can be declared in different ways

Var:

Using var keyword to declare, it will declare inside the scope not globally

```
var value1 = "Hello";  
var value2 = 4;  
var value3 = 5;
```

Let:

Using let keyword to declare, The declared values can be used globally within the scope.

```
let value1 = "Hello";  
let value2 = 4;  
let value3 = 5;
```

Const:

Using const keyword to declare, declaration must be initialized.

```
const value1 = "Hello";  
const value2 = 4;  
const value3 = 5;
```


Recap:

- JavaScript gives life to a static web page by making it functional.
- Node.js allows us to run JavaScript on the server.
- In sync programming, the program is executed line by line, one line at a time.
- In async programming the program doesn't wait for the task to complete and can move on to the next task.
- Variables are container for storing value
 - Var
 - Const
 - let

4.4. Rules for declaring a variable

1. Variable name should start with an alphabet letter
2. It can start with \$ or _ symbol
3. Should not start with special characters except \$ and _
4. Should not include space between variable name

4.5. Mutable and Immutable in java script

Mutable is a type of variable that can be changed and only objects and arrays are mutable.

A mutable object is an object whose state can be modified after it is created.

Immutable are the objects whose state cannot be changed once the object is created.

4.6. Statement

- ❑ Statements are the basic units of the program
- ❑ Statements are set of instruction given to the computer to perform actions

```
var value1 = "Hello";  
var value2 = 4;  
var value3 = 5;
```

4.7. Block

Basically, the collection of statements are blocks. Blocks are also set of instruction given to the computer.

```
3  <script type="text/javascript">  
4      window.onload=function calculation()  
5      {  
6          var value1 = 2;  
7          var value1 = 4;  
8          alert(value1)  
9      }  
10  
11  </script>
```

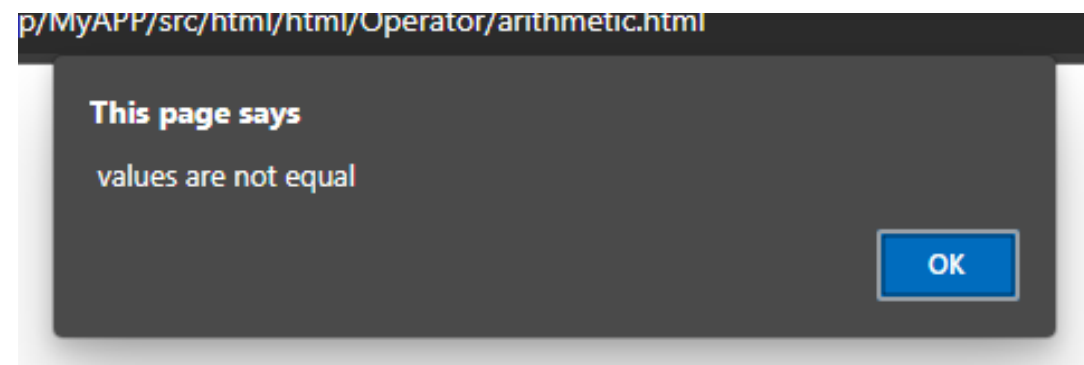
4.8. Function

- ❑ A JavaScript function is a block of code designed to perform a particular task.
- ❑ Functions are reusable pieces of code it can be called where we need to use it.
- ❑ Syntax for JavaScript function

```
function fun_name()  
{  
    // code to be executed  
}
```

```
function fun_name(parameter1, parameter2)  
{  
    // code to be executed  
}
```

```
onload=  
function assignment(){  
    var a = '2';  
    var b = 2 ;  
    if (a===b){  
        alert(" Values are equal ")  
    }  
    else{  
        alert(" values are not equal")  
    }  
}
```



Recap:

- Variable name should start with an alphabet letter, Should not start with special characters except \$ and _
- A mutable object is an object whose state can be modified.
- Immutables are the objects whose state cannot be changed.
- Statements are set of instruction given to the computer.
- Collection of statements are blocks.
- Functions are reusable pieces of code it can be called where we need to use it.

4.9. Callback Function

A callback function is a function passed into another function as an argument.

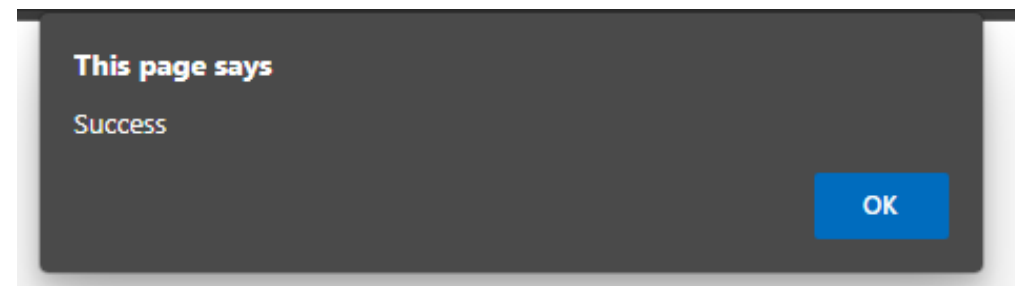
Sample Program

<pre>function A() { function B() }</pre>	<pre>function B() { //Code to be executed }</pre>
--	---

- function B will be executed when function A is being called.
- While executing function it will set a reference point in the memory so once the function B is executed it will return to the function A.

```
<script>
function test()
{
  sum();
  alert('Success')
}

function sum(){
  var a = 2;
  var b = 5;
  var sum = a+ b;
  return(sum);
}
</script>
```



4.10. Form Elements

To reference the <form> element, you can use DOM selecting methods such as

- ❖ getElementById
- ❖ getElementsByName
- ❖ getElementsByClassName
- ❖ getElementsByTagName

```
function getelement() {
    txtinput = document.getElementById('txtinput');
    txtInputName = document.getElementsByName('inputName');
    txtTagInput = document.getElementsByTagName('input');
    txtClassInput = document.getElementsByClassName('clsinput');

    console.log('Element By Id', txtinput);
    console.log('Element By Name', txtInputName);
    console.log('Element By Tague Name', txtTagInput);
    console.log('Element By Class Name', txtClassInput)
}
```

```
<div>
    <input type="text" class="clsinput" id="txtinput" name="inputName" />
    <button id="btnSubmit" onclick="getelement()">Click</button>
</div>
```

Element By Id <input type="text" class="clsinput" id="txtinput" name="inputName">

Element By Name ▶ NodeList [input#txtinput.clsinput]

Element By Tague Name ▶ HTMLCollection [input#txtinput.clsinput, txtinput: input#txtinput.clsinput, inputName: input#txtinput.clsinput]

Element By Class Name ▶ HTMLCollection [input#txtinput.clsinput, txtinput: input#txtinput.clsinput, inputName: input#txtinput.clsinput]

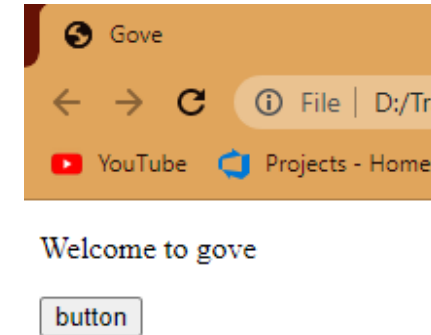
>

4.11. Displaying values in JavaScript

The values or output can be displayed in JavaScript in four ways.

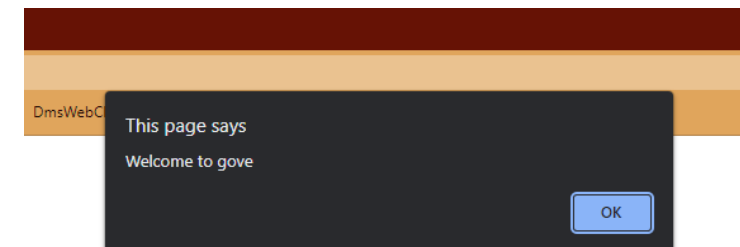
- ❑ InnerHTML (document.getElementById("").innerHTML="");)

```
<!DOCTYPE html>
<head>
<script type="text/javascript">
  function show() {
    document.getElementById('ptag').innerHTML = "Welcome to gove";
  }
</script>
<title>Gove</title>
</head>
<body>
<p id="ptag"></p>
<input type="button" onclick="show();"
  value="button" />
</body>
</html>
```



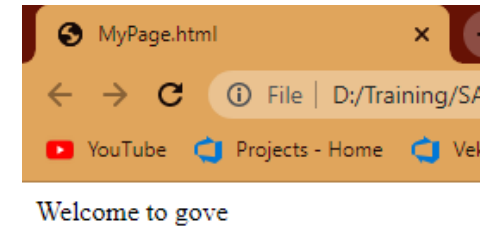
- ❑ alert().

```
<script type="text/javascript">
  function show() {
    alert('Welcome to gove')
  }
</script>
```



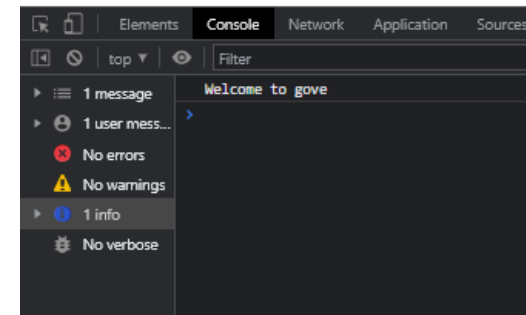
❑ Document.write()

```
<script type="text/javascript">
  function show() {
    document.writeln('Welcome to gove')
  }
</script>
```



❑ console.log().

```
<script type="text/javascript">
  function show() {
    console.log('Welcome to gove')
  }
</script>
```



Recap:

- A callback function is a function passed into another function as an argument.
- DOM selecting methods
 - `getElementById`
 - `getElementsByName`
 - `getElementsByClassName`
 - `getElementsByTagName`
- Displaying values in JavaScript
 - `innerHTML (document.getElementById("").innerHTML="");`
 - `alert()`.
 - `Document.write()`
 - `console.log()`.

4.12. Array

- ❖ Array is a special variable which can hold more than one value. Array act like containers for values.
- ❖ Array can be declared only with [].
- ❖ Array can be declared in two ways

```
window.onload=function arrayfuc(){  
    var x = Array[7]  
    x = [1,2,3,4,5,7]  
    alert(x[4])  
}
```



This page says

5

OK

```
window.onload=function constarray(){  
    const x = [1,2,3,4,5,7]  
    alert(x[4])  
}
```



This page says

5

OK

4.13. Redeclaring variable

var a = 10;

var a = 11 ;

var will take first value a = 11

let a = 10;

let a = 11;

Where a cannot be redeclared in let

```
<script type="text/javascript">
  window.onload=function calculation()
  {
    var value1 = 2;
    var value1 = 4;
    alert(value1)
  }
</script>
```

```
head>
<script type="text/javascript">
  window.onload=function calculation()
  {
    let value1 = 2;
    let value1 = 4;
    alert(value1)
  }
</script>
```

Cannot redeclare block-scoped variable 'value1'. javascript

[View Problem \(Alt+F8\)](#) No quick fixes available

const value should be assigned to a function where the value is being declared.

4.14. Object

- ❑ Object is nothing but variable which can contain more values
- ❑ Object contains the key and values
- ❑ Best Practice is using Object in const keyword

e.g.

In Programs, we can use one or more objects. Declare one object in a computer language using Single Object.

Single Object :

```
let object = {
    Key : value,
    Key : value
}
```

Multiple Object :

```
Let arrayObject = [ { object }, { object }, { object } ]
```

```
<script type="text/javascript">
    function display(){
        var value1 = {
            name : "Diya",
            age : 22
        }
        alert(calculation)
    }

    function arrayFunction(){
        var arrayObject = [
            {
                name : "Devi",
                age : 23
            },
            {
                (property) age: number
                age: 20
            }
        ]
    }
</script>
```

4.15.Operators



Operators are special symbol in JavaScript used for performing operations using operands

Operands

An operator is a special symbol used to perform operations on operands (values and variables)

e.g. $2 + 4$ **Operators** : + **Operands** : 2, 4;

Types of Operators

- ❖ Arithmetic Operators
- ❖ Assignment Operators
- ❖ Comparison Operators
- ❖ Logical Operators
- ❖ Conditional Operators

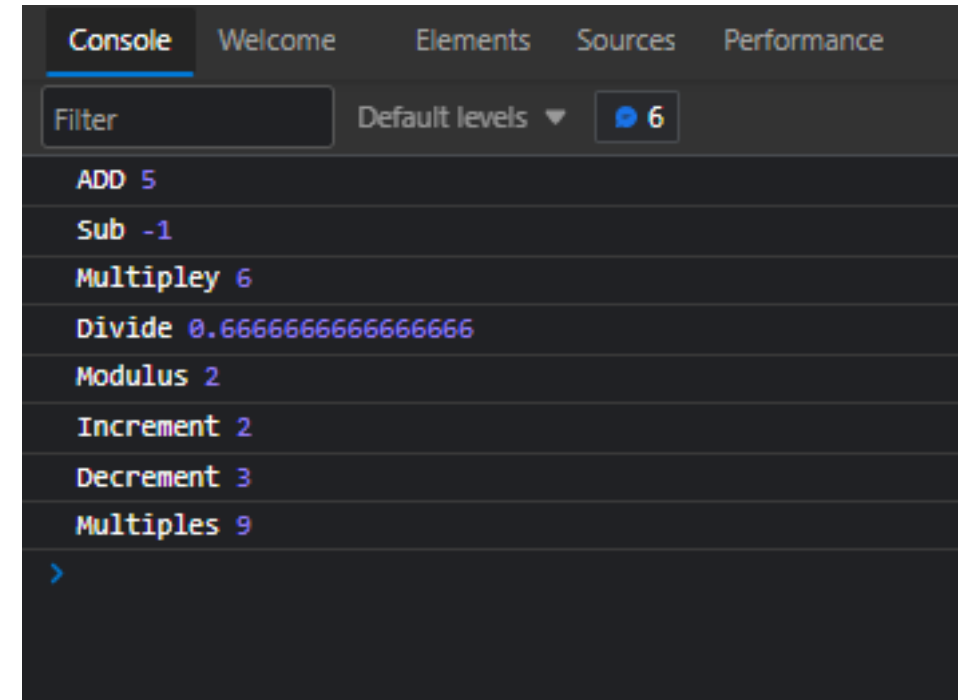
4.15.1. Arithmetic Operator

- Arithmetic operators are used to perform **arithmetic calculations**.
- Arithmetic operators (+ , - , * , / , %)

```

window.onload=function calculation()
{
    // Arithmetic Operator
    var a = 2;
    var b = 3;
    var add = a + b ;
    console.log("ADD", add);
    var sub = a - b;
    console.log("Sub",sub);
    var multiply = a * b;
    console.log("Multiply", multiply);
    var divide = a / b;
    console.log("Divide",divide);
    var modulus = a % b;
    console.log("Modulus",modulus);
    var increment = a ++
    console.log("Increment",increment);
    var decrement = b--
    console.log("Decrement",decrement);
    var multiples = a ** b
    console.log("Multiples",multiples);
}

```



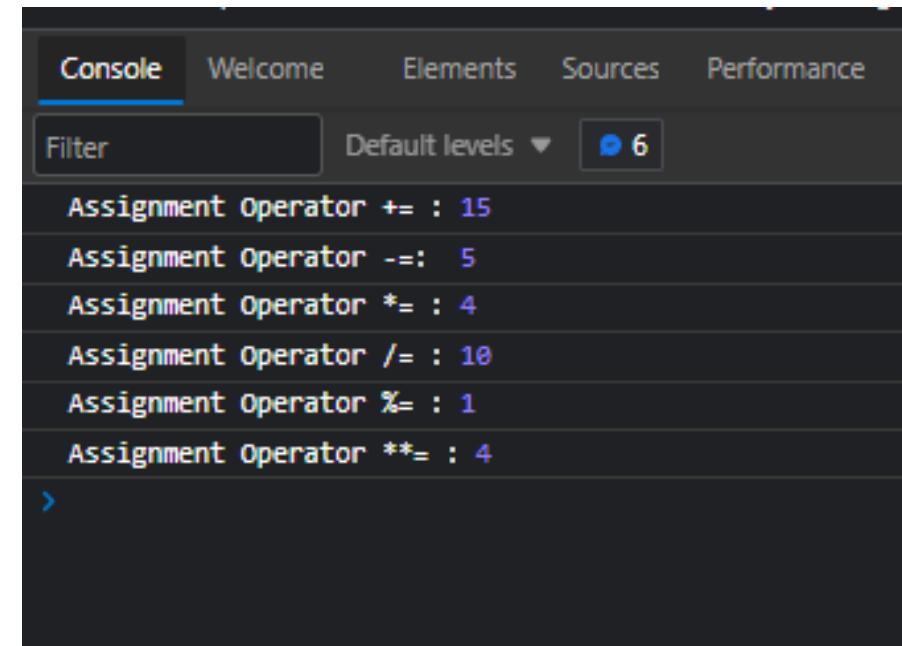
Recap:

- Array is a special variable which can hold more than one value. [].
- Redefining variable - var & let.
- const value should be assigned to a function where the value is being declared.
- Object is nothing but variable which can contain more values
- Object contains the key and values.
 - Single Object
 - Multiple Object
- Operators are special symbol in JavaScript used for performing operations using operands
 - Arithmetic Operators
 - Assignment Operators
 - Comparison Operators
 - Logical Operators
 - Conditional Operators

4.15.2. Assignment Operators

- Assignment operators are used to **assign** values to variables.
- Assignment Operators(= , +=, -=, *=, /=,%=, **=)

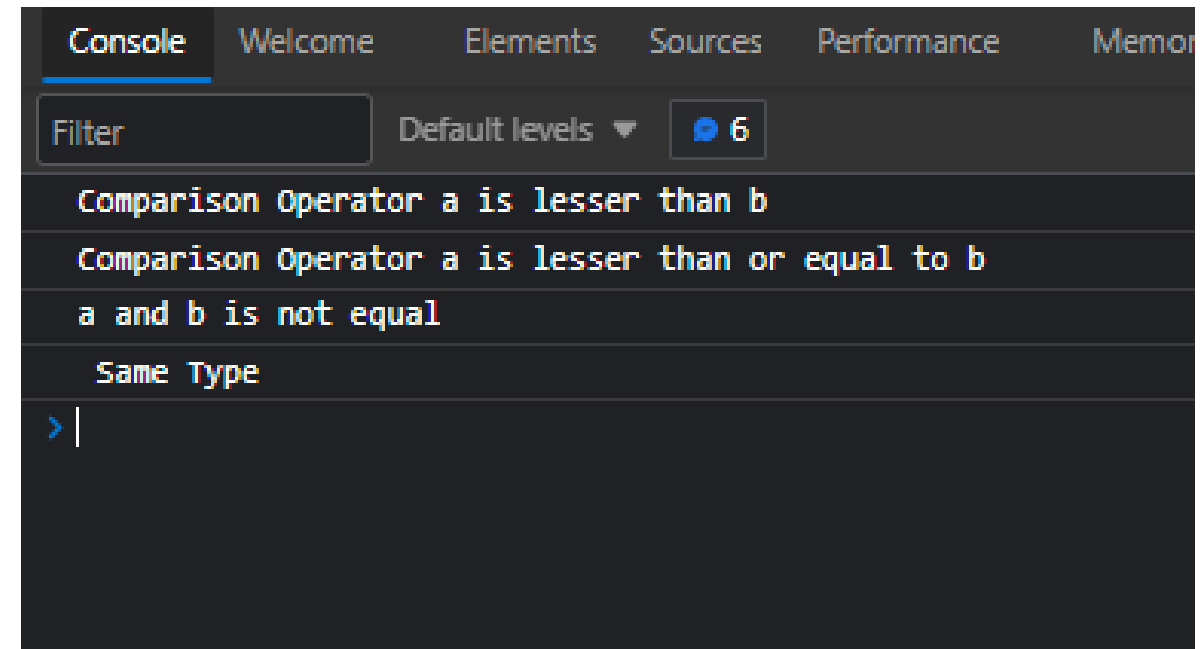
```
<script type="text/javascript">
  window.onload=function calculation()
  {
    // Assignment Operator
    let add = 10;
    add += 5;
    console.log("Assignment Operator += :",add)
    let sub = 12;
    sub -=7;
    console.log("Assignment Operator -=: ",sub)
    let mul = 2;
    mul *=2;
    console.log("Assignment Operator *= :",mul)
    let divide = 5;
    divide /= 2
    console.log("Assignment Operator /= :",divide);
    let Modulus = 3;
    Modulus %= 2;
    console.log("Assignment Operator %= :",Modulus)
    let exp = 2;
    exp **= 2;
    console.log("Assignment Operator **= :",exp)
  }
</script>
```



4.15.3. Comparison Operator

- Comparison operators are used in logical statements to determine equality or difference between variables or values.
- Comparison Operators (== , != , === , !== , > , >= , < , <=)
- == only Compares the value ; === compares both value and data type

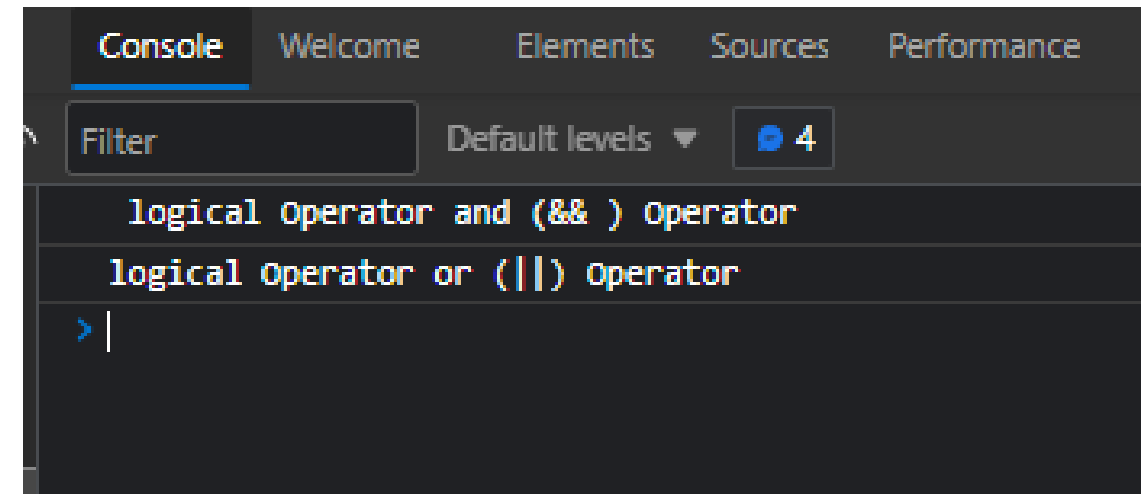
```
function compareOperators() {  
  var a = 10, b = 20; var c = '20'  
  if (a > b) {  
    console.log('Comparison Operator a is greater than b')  
  }  
  if (a < b) {  
    console.log('Comparison Operator a is lesser than b')  
  }  
  if (a >= b) {  
    console.log('Comparison Operator a is greater than or equal to b')  
  }  
  if (a <= b) {  
    console.log('Comparison Operator a is lesser than or equal to b')  
  }  
  
  if (a == b) {  
    console.log('Comparison Operator a value is equal to b value')  
  }  
  // Compare Operators === check value and data type  
  if (a === c) {  
    console.log('a and b is data type and value equal')  
  } else {  
    console.log('a and b is not equal')  
  }  
  
  // !== not equal type  
  if (a !== c) {  
    console.log(" Same Type");  
  }  
  else {  
    console.log(" Not Equal ")  
  }  
}
```



4.15.4. Logical Operator

- Logical operators perform logical operations and return a Boolean value, either true or false.
- Logical operator **&&**, **||**, **!**

```
<script>
onload=
//logical Operators
function logicalOperator() {
  var a = 10; b = 5;
  if (a == 10 && b == 5) {
    console.log (" logical Operator and (&& ) Operator")
  }
  if (a == 10 || b == 10) {
    console.log('logical Operator or (||) Operator')
  }
  if (a != 10) {
    console.log('logical operator (!) Not Operator')
  }
}
</script>
```



4.15.5. Conditional or Ternary Operator

Conditional operator is the only operator in JavaScript which will include three operands. It can be used to check two conditions.

Syntax

The syntax of ternary operator is :

```
testCondition ? expression1 : expression 2;
```

The testCondition is a boolean expression that results in either **true** or **false**. If the condition is

- true - **expression1** (before the colon) is executed
- false - **expression2** (after the colon) is executed

Sample program for ternary operator.

```
<body>
  <h2>Ternary Operator</h2>
  <h4>Sample EmailId validation using Ternary Operator :</h4>
  <input id="Email" type="text" placeholder="Enter Your email id" />
  <button onclick="myFunction()">Log in </button>
  <script>
    function myFunction() {
      let emailId = document.getElementById("Email").value;
      let validation = (emailId == "meera_nagarajan@gove.co" ? "Logged in successfully" : "Enter Valid Email");
      alert(validation)
    }
  </script>
</body>
```



Ternary Operator

Sample EmailId validation using Ternary Operator :

meera_nagarajan@gove.co

Log in

This page says
Logged in successfully

OK

4.16. Debugging



- ❑ Debugging skill is very important for every programmer because the programmer should be able to identify the error occurring in his own code.
- ❑ Logical thinking will enhance the debugging skills of a programmer.
- ❑ If a programmer is not capable of debugging the errors in the program, then the programmer will not be considered as a good programmer.
- ❑ When an error occurs in the program the programmer should think only based on the error to solve it.
- ❑ Stack trace is an important tool in debugging. Stack trace is nothing but finding out the error occurred in which line of a function, where the function is used and finding out the error by understanding its root.
- ❑ Try to solve the error by ourself , rather than getting help from our mates to solve a error at the first moment of the time error occurs.
- ❑ Capturing the error faced is a good practice, so the next time when we face the same error it can solved effortlessly.

4.16.1. Exception Handling in JavaScript



- ❑ Exception handling is a process or method used for handling the error statements in the code and executing them.
- ❑ It allows us to handle the flow control of the code/program. So, the code will not break due to the error occurred.

4.16.2. Exception Handling Statements

- ❑ The **try** statement defines a code block to run (to try).
- ❑ The **catch(err)** statement defines a code block to handle any error. Where err is object which is assigned to the catch. Instead of err any object can be used
- ❑ The **finally** statement defines a code block to run regardless of the result.
- ❑ The **throw** statement defines a custom error.

Statements

- ☐ throw statements
- ☐ try...catch statements
- ☐ try...catch...finally statements.

General Syntax

```
try {  
    //Block of code to try  
}  
catch(err) {  
    //Block of code to handle errors  
}  
finally {  
    //Block of code to be executed regardless of  
    the try / catch result  
}
```

```
//try catch error  
function cal(a,b){  
    var a = 2;  
    var b = 6;  
    try{  
        let sum =a + b;  
        console.log(sum)  
    }  
    catch(err){  
        console.log('catch',err.message);  
    }  
}
```



```
Filter Default levels 6  
catch d is not defined  
>
```


4.16.3. Debugger

- Debugger can be used to find out the error in the code being executed.
- By using debugger keyword, developer can understand the flow of the execution block.

```
function debugFunction(){  
    let val1 = 2;  
    let val2 = 4  
    debugger;  
    try{  
        let sums =val1 + val2;  
        console.log(sums)  
    }  
    catch(err){  
        console.log('catch',err.message);  
    }  
}
```



Elements Sources Performance Memory Application

error.html x

```
// using debugger  
function debugFunction(){  
    let val1 = 2;  
    let val2 = 4  
    debugger;  
    try{  
        let sums =val1 + val2;  
        console.log(sums)  
    }  
    catch(err){  
        console.log('catch',err.message);  
    }  
}
```

- Hence in the above example the code statements can be executed so that the error can be identified like where some values may be passed as undefined or null.

Recap:

- Assignment operators are used to **assign** values to variables.
- Comparison operators are used in logical statements to determine equality or difference between variables or values.
- Logical operators perform logical operations and return a Boolean value, either true or false.
- Conditional operator has three operands, It can be used to check two conditions.
- Stack trace is an important tool in debugging, finding out the error occurred in which line of a function, where the function is used and finding out the error by understanding its root.
- Exception handling is a process or method used for handling the error statements in the code
 - throw statements
 - try...catch statements
 - try...catch...finally statements.
- Debugger keyword can be used to find out the error in the code being executed.

Think !

1. To identify the error occurring in our own code is called _____
2. _____ will enhance the debugging skills of a programmer.
3. List out the Exception Handling Statements:
4. Keyword that used for Debugging

4.17. Type Conversion

It is the process of converting data of one type to another.

- Converting Strings to Numbers
- Converting Booleans to Numbers
- Converting Numbers to Booleans

number()-Will convert string into an integer or number and

```
let sum = Number('4')
console.log("Sum",sum)

let hello = Number("Hello")
console.log("hello",hello)

let boolean = Number(true)
console.log("Boolean",boolean)

let Value1 = Number(4) + Number(true)
console.log("Value1",Value1)
```



```
Console Welcome Elements S
Filter Default levels ▼
Sum 4
hello NaN
Boolean 1
Value1 5
```

parseInt()-Will convert the string into a integer or number.

```
//Convert String Number to Integer
let Values = parseInt(a) + b;
console.log("Values",Values)

// Convert String Character to Number
let hello = b + parseInt(c)
console.log("Hello",hello)
```

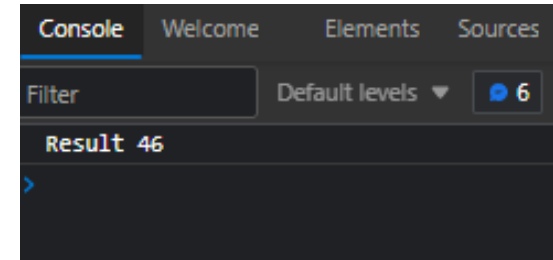


```
Values 10
Hello NaN
```

➤ Converting Numbers to Strings

String() - will convert numbers or integers to strings

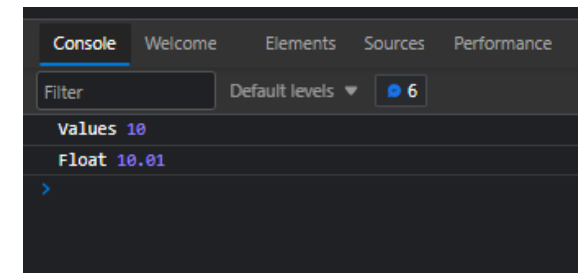
```
let a = '4';  
let b = 6;  
// Convert integer to String  
let result = a + b.toString()  
console.log("Result", result)
```



➤ Converting Numbers to Strings

parseFloat() - will convert a string to float value

```
// Convert String Number to Integer  
let Value = parseInt("4.02") + 6;  
console.log("Values", Value)  
  
// convert to float  
let float = parseFloat("6.01") + parseInt("4")  
console.log("Float", float)
```



Exercise 2:



Exercise 2 Continues Followed by Exercise 1, As a Departmental store owner ,Able to see the HTML Form Data in the created webpage.

Following things are to be considered while creating the Functional webpage:

1. Write a Function to get the HTML form data and store it into an array of object.
2. Display an array in Tabular structure using for loop.
3. Declare a variable outside the for loop and initialized as "1".
4. Do the Type Conversion and Increment the count using Assignment operator for displaying the serial number in above tabular column
5. Shows alert message as customer details saved successfully
6. Write Exception Handling for the Written function.
7. Calls the function, when button clicks.

Think !

1. _____ is used to transfer of data from one data type to another.

2. _____ convert the string into a integer or number.

Option1:parseInt()

Option2:Parsefloat()

Looping Statements

Loops are used to repeat a block of code.

1. If statement:

Syntax:

```
if (condition) {  
    // the body of if  
}
```

Sample Code:

```
<script>  
|   var x = 5; y = 10;  
//if Statement  
if (x < y) {  
|   console.log('x is greater than y')  
}  
</script>
```



Output:

```
x is greater than y  
> |
```


2.if else Statement

Syntax:

```
if (condition)
{ // block of code if
condition is true }
else { // block of code if
condition is false }
```

Sample Code:

```
<script>
  var x = 5; y = 10;
  if (x > y) {
    console.log('x is greater than y')
  }
  else {
    console.log('y is greater than x')
  }
</script>
```



Output:

y is greater than x

if, else if & else Statement



Syntax:

```
if (condition1) {  
    // block of code to be executed if condition1 is true  
} else if (condition2) {  
    // block of code to be executed if the condition1 is  
    false and condition2 is true  
} else {  
    // block of code to be executed if the condition1 is  
    false and condition2 is false  
}
```

Sample Code:

```
<script>  
    var x = 5; y = 5;  
    if (x > y) {  
        console.log('x is greater than y')  
    }  
    else if (x < y) {  
        console.log('x is lesser than y')  
    }  
    else {  
        console.log('x and y are equal')  
    }  
</script>
```



Output:

```
x and y are equal  
>
```

Switch statement

Syntax:

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

Sample Code:

```
<script>
  let a = 2;
  // switch statement
  switch (a) {

    case 1:
      a = 'one';
      break;
    case 2:
      a = 'two';
      break;
    case 3:
      a = 'three';
      break;
    default:
      a = 'not found';
      break;
  }
  console.log(`The value is ${a}`);
</script>
```



Output:

```
The value is two
>
```

While Statement

Syntax:

```
while (condition) {  
    // code block to be  
    executed  
}
```

Sample Code:

```
let i = 0;  
let text = ""  
while (i < 5) {  
    text += "The number is " + i + "\n";  
    i++;  
}  
  
console.log(text)
```



Output:

```
The number is 0  
The number is 1  
The number is 2  
The number is 3  
The number is 4
```

Do while statement

Syntax:

```
do {  
    code block to be  
    executed  
}  
while (condition);
```

Sample Code:

```
let i = 0, text="";  
do {  
    text += "The number is " + i+"\n";  
    i++;  
}  
while (i < 5);  
console.log(text)  
</script>  
<title>  
    Looping Statements  
</title>  
</head>  
<body>
```



Output:

```
The number is 0  
The number is 1  
The number is 2  
The number is 3  
The number is 4
```

> |

for statement

Syntax:

```
for (initialExpression; condition; updateExpression) {  
    // for loop body  
}
```

Sample Code:

```
<script>  
    let text = "", i=0  
    for (let i = 0; i < 5; i++) {  
        text += "The number is " + i + "\n";  
    }  
    console.log(text);  
</script>
```

Output:

```
The number is 0  
The number is 1  
The number is 2  
The number is 3  
The number is 4
```

For of Statement

Syntax:

for (variable of iterable)
statement

Sample Code

```
//for of statement  
const a = ["MongoDB", "Express", "ReactJS", "nodeJS"];  
  
for (const x of a) {  
  console.log(x);  
}  
  
</script>  
<title>
```

Output:

MongoDB
Express
ReactJS
nodeJS

for each statement

Syntax

`array.forEach(function(currentValue, index, arr), thisValue)`

Sample Code

```
const arr=["apple","mango","pineapple","watermelon","orange"]
arr.forEach(x => {
  console.log("The fruit is "+x);
});

</script>
```

Output:

The fruit is apple
The fruit is mango
The fruit is pineapple
The fruit is watermelon
The fruit is orange

4.18. Break & Continue Statement

Break Statement

- ❑ The break statement usually jumps out of the looping statement.
- ❑ When we use break statement the code will be just stop executed and will return the value.

```
<h2>JavaScript Break Statement</h2>
<p id="demo"></p>
<script>
  let text = "";
  for (let i = 0; i < 10; i++) {
    if (i === 3) { break; }
    text += "The number is " + i + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
</script>
```



JavaScript Break Statement

The number is 0
The number is 1
The number is 2

Continue Statement

The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

```
<h2>JavaScript Continue Statement</h2>
<p>A loop which will skip the step where i = 3.</p>
<p id="demo"></p>
<script>
  let text = "";
  for (let i = 0; i < 10; i++) {
    if (i === 3) { continue; }
    text += "The number is " + i + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
</script>
```



JavaScript Continue Statement

A loop which will skip the step where i = 3.

The number is 0
The number is 1
The number is 2
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

Think !

28. Which statement of the code will be just stop executed and will return the value.

Option1: Break

Option2: Continue

4.19. JavaScript Arrow Function

- Arrow function (`=>`) helps us to write smaller function syntax code.
- If the function has a single statement, it can return the value by its own. There is no need to use `return()` statement.

```
//Without arrow function
hello = function() {
    return "Hello World!";
}

//By using arrow function
hello = () => {
    return "Hello World!";
}

//Hence the function has one statement no need to use of reutur()
hello = () => "Hello World!";
```

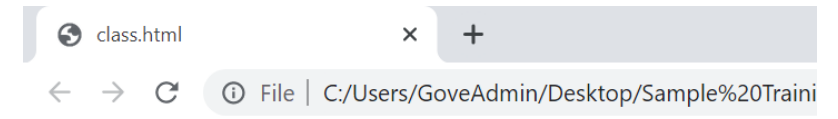
4.20. Class

- ❖ Class is a template for an object.
- ❖ Class is a collection of function and properties.
- ❖ e.g. : Consider a car having
 - 1.properties like color, seats, structure, type, model etc..
 - 2.functions like acceleration, braking, gear shifting etc..

Constructor in class

The constructor method is a special method of a class for creating and initializing an object instance of that class.

```
<body>
<h2>Javascript Class</h2>
<p id="demo"></p>
<script>
  class CarDetails {
    constructor(brand, model, year) {
      this.carName = brand;
      this.model = model;
      this.year = year;
    }
  }
  myCar = new CarDetails("Audi", "XUV", "2021");
  document.getElementById("demo").innerHTML =
  "My first car is " + myCar.carName + " which is a " + myCar.model + " model that i bought in the year" + myCar.year + ".";
</script>
</body>
```



Javascript Class

My first car is Audi which is a XUV model that i bought in the year2021.

Recap

- Type Conversion is the process of converting data of one type to another.
 - `number()`
 - `parseInt()`
 - `String()`
 - `parseFloat()`
- Break statement would stop executed the current loop and will return the value.
- The `continue` statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.
- Arrow function (`=>`) helps us to write smaller function syntax code.
- Class is a template for an object, Collection of functions & properties.
- The constructor method is a special method of a class for creating and initializing an object

Think !

- 1.How to represent an arrow function?
- 2.Whether there is a need of return statement, if there is a single function statement-yes/No
- 3._____ is a template for an object.
4. Class is a collection of _____ and _____.
5. Which method of a class is used for creating and initializing an object instance of that class?

4.21. JavaScript Module

- ❑ JavaScript modules help us to break the code into separate code which makes the easier to maintain the code-base.
- ❑ JavaScript modules basically rely on import and export statements.
- ❑ JavaScript basically run on the browser and the codes are rendered form the server.
- ❑ It runs either on http or https protocol.
- ❑ The import statements is rendered from the server during the runtime of the application.
- ❑ The export function module is an individual file `JS moduleExport.js`

```
export default function printText()  
{  
  console.log("Welcome To GOVE")  
}
```

- ❑ The export module can be imported in another `JS moduleImport.js` file using

```
import printText from "./moduleExport";
```

4.22. Regular Expression

A regular expression is basically a pattern of characters used for pattern-matching "search-and-replace" functions on text

Syntax for Regular Expression: */pattern/modifier(s);*

Example : let pattern = /Gove/i

Modifiers

Modifiers are used to perform case-insensitive and global searches.

Modifier	Description
g	Perform a global match (find all matches rather than stopping after the first match)
i	Perform case-insensitive matching
m	Perform multiline matching

Search using regular expression

```
<body>
<h2>JavaScript Regular Expressions</h2>
<p id="demo"></p>
<script>
let text = "Welcome to GOVE";
let pattern = /gove/i;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
```



← → ↻ ⓘ File | C:/Users/GoveAdmin/Desktop/S

JavaScript Regular Expressions

GOVE

Replace using regular expression

```
function myFunction() {
let text = document.getElementById("demo").innerHTML;
document.getElementById("demo").innerHTML =
text.replace(/gove/i, "gove");
}
```



JavaScript String Methods

Replace "gove" with "GOVE" in the paragraph below:

Try it

Welcme to gove

Think !

1. _____ help us to break the code into separate file .
2. List out two types of statements in Module.
3. Syntax for regular expression

4.23.StringMethod

- Strings are used to represent and work with a sequence of characters.
- JavaScript automatically converts primitive strings to String objects so that it's possible to use String methods and access properties even for primitive strings.

```

// String Methods
let test = "ABCDEFGHIJKLMNPOQRSTUVWXYZ";
let length = test.length;

text = 'Model5, ReactJS, MongoDB';
let part = text.slice(7, 13);

text = 'Model5, ReactJS, MongoDB';
part = text.slice(7);

let str = 'Model5, ReactJS, MongoDB';
part = str.substring(7, 13);

text = 'Please visit XYZ!';
let newText = text.replace('XYZ', 'GOVE');

text = 'Please visit XYZ and XYZ!';
newText = text.replace(/XYZ/g, 'Gove');

text = text.replaceAll('Cats', 'Dogs');

let text1 = 'Hello World!';
let text2 = text1.toUpperCase();
let text3 = text1.toLowerCase();

text1 = 'Welcome to';
text2 = 'Gove';
text3 = text1.concat(' ', text2);

text1 = '      Welcome to GOVE      ';
text2 = text1.trim();
text3 = text1.trimStart();
let text4 = text1.trimEnd();

text = '5';
let padded = text.padStart(4, 'x');

str = 'Please locate where "Gove" occurs!';

```

Math Object

The math object provides you properties and methods for mathematical constants and functions.

Unlike other global objects, Math is not a constructor. All the properties and methods of Math are static and can be called by using Math as an object without creating it.

```
<script type="text/javascript">  
  // Math Object  
  let text = "Mathematics consists of true facts about imaginary objects";  
  let length = text.length;  
  alert( length )  
</script>
```

Date Object

- Date objects represent a single moment in time in a platform-independent format.
- Date objects encapsulate an integral number that represents milliseconds

```
<script type="text/javascript">  
  let date = new Date();  
  const d = new Date("October 13, 2014 11:13:00");  
  console.log ('Date:',date.getDate())  
  console.log ('Day:',date.getDay())  
  console.log ('Fullyear',date.getFullYear())  
  console.log ('Hours:',date.getHours())  
  console.log ('Milliseconds:',date.getMilliseconds())  
  console.log ('Month',date.getMonth())  
  console.log ('Seconds',date.getSeconds());  
  console.log ('Time:',date.getTime());  
  console.log ('UTC:',date.getUTCDate())  
  console.log ('Number:',Number(date))  
  console.log('Time Zone ',date.getTimezoneOffset())  
  
</script>
```

4.24.Event Handler

- **OnClick** - The onclick event occurs when the user clicks on an HTML element.
- **OnChange** - The onchange event occurs when the value of an element has been changed.
- **OnSubmit** - The onsubmit event occurs when a form is submitted.
- **onload** - The onload event occurs when an object has been loaded.

4.25. JavaScript Strict Mode

- ❑ The purpose of "use strict" is to indicate that the code should be executed in "strict mode".
- ❑ Using "use strict " in our program will help us to write cleaner code and will prevent us from undeclared variables.

```
"use strict";  
myFunction();  
function myFunction() {  
  y = 3.14;    // This will also cause an error because y is not declared  
}
```

4.26. JavaScript Best Practices

1. Maintaining clean code by having a good format providing indentations and can use code formatter like prettier.
2. Avoid Global variable declaration rather declare variable locally.
3. const Variable must be initialized once it's been declared.
4. Declare array and object with const.
5. Being aware of automatic type conversion.

4.27. JavaScript Scope

The variables can be declared in java Script within the below mentioned scopes

Block Scope

Variables declared inside a { } block cannot be accessed from outside the block. Variables can be declared inside the block by using const or var or let.

```
{
  let x = 5;
}
```

Function Scope

Variables defined inside a function are not accessible from outside the function.

```
// Function Scope
function myFunction() {
  var carName = "Volvo";
}
```

Global Scope

Global variables can be accessed from anywhere in a JavaScript program.

```
// Global scope
var x = 2;
```

4.28. JSON

- JSON is JavaScript Object Notation
- It can be store as an array of an object also.
- It can be used to store and transport data
- It is used to send data from a server to web-page .

Sample JSON data.

```
{  
  "company": {  
    "name": "GOVE",  
    "moto": "Positive",  
    "active": true  
  }  
}
```

Recap

- JavaScript modules help us to break the code into separate code.
- It rely on import and export statements.
- A regular expression is basically a pattern of characters used for pattern-matching "search-and-replace"
- Scope - Block, Function & Global.
- JSON is used to send data from a server to web-page .
- "use strict " is used to write cleaner code.
- Best Practices, Avoid Global variable declaration rather declare variable locally.

Think !

- 1) Converting to Upper and Lower Case ?
- 2)The _____ method can be used instead of the plus operator.
- 3)Replace syntax
- 4)what is the length property returns the length of a "Gove Enterprises "
- 5)_____ refers to the part of a program where we can access a variable
- 6) What is the object used to get numbers randomly?
- 7)What is the object used to get current date ?
- 8). List out the types of scope?
- 9). Variables defined inside a function are called?
- 10) Variables can be accessed from anywhere in a program is _____
- 11). JSON Stands for _____
- 12). _____ is used to write cleaner code and will prevent us from undeclared variables.

Think !



1. JavaScript is a _____ programming language.
2. Whether JavaScript is a case Sensitive Language? Yes/No
3. JavaScript can be defined as,
option1: Client-side Programming Language.
option2: Server-side Programming Language.
4. What runtime environment is used for JavaScript?
5. Java script built on Google Chrome's JavaScript ____ Engine.
6. _____ program is executed line by line, one line at a time.
Option1:synchronous
Option2:Asynchronous
7. _____ are container for storing value.
8. What are all the ways variable can be declare?
9. Whether variable name start with \$(dollar) and _ (Underscore)? Yes/No
10. _____ is an object whose state can be modified after it is created.

11 _____ are the objects whose state cannot be changed once the object is created.

12. _____ is a block of code designed to perform a particular task.

Option1: Constant

Option2: Function

13. In call back function, Assume there is 2 Functions named as functionA and functionB, Which function will execute first?

14. How to display the values in javascript?

Option1: Document.write

Option2: Document.read;

Option3: Document.write()

15. _____ is a special variable which can hold more than one value.

16. Whether by using "let" variable can be redeclared? yes/no

17. Object contains the ____ and ____ .

18. Difference between operators and operands

19. Types of Operators

20. Logical operations return _____ value
21. Syntax for conditional or Ternary Operator
22. To identify the error occurring in our own code is called _____
23. _____ will enhance the debugging skills of a programmer.
24. List out the Exception Handling Statements:
25. Keyword that used for Debugging
26. _____ is used to transfer of data from one data type to another.
27. _____ convert the string into a integer or number.
Option1:parseInt()
Option2:Parsefloat()
28. Which statement of the code will be just stop executed and will return the value.
Option1: Break
Option2: Continue
29. How to represent an arrow function?
30. Whether there is a need of return statement, if there is a single function statement- yes/No
31. _____ is a template for an object.
32. Class is a collection of _____ and _____.

33. Which method of a class is used for creating and initializing an object instance of that class?
34. _____ help us to break the code into separate file .
35. List out two types of statements in Module.
36. Syntax for regular expression
37. ____ refers to the part of a program where we can access a variable
38. List out the types of scope?
39. Variables defined inside a function are called?
40. Variables can be accessed from anywhere in a program is _____
41. JSON Stands for _____
42. _____ is used to write cleaner code and will prevent us from undeclared variables.

Exercise 3:

Topic: Class & Construction in Class

1. Create a class “Student” having probable attributes – student’s name, address, roll number and other related parameters.
2. Students like John, Ricky, Deep, etc., will be the objects of a class Student
3. Should have the same class members as student’s name, address, roll number, etc.

Topic: Break & JavaScript Continue

1. Create a for loop from 1 to 20
2. It should print only a prime numbers using break and continue statement.

Topic: Regular expression & Modifier

1. Create a Paragraph which has a sample text.
2. Two input box for find and replace and submit button in HTML
3. Enter the find value in find input box.
4. Enter the value in replace value in replace input box.
5. After clicking the submit button, the text is replaced in the above paragraph.
6. Use Regular expression for Find and Replacing the text in the paragraph.

Topic: Arrow Function

1. Create a function called Arithmetic Operations which has two parameters x and y.
2. Inside the Created Arithmetic Operations functions create an arrow function for sum, diff, product and division.
3. Display the output of the arrow function one by one.
4. Call the Arithmetic Operations function in onload like Module.

Happy Coding!