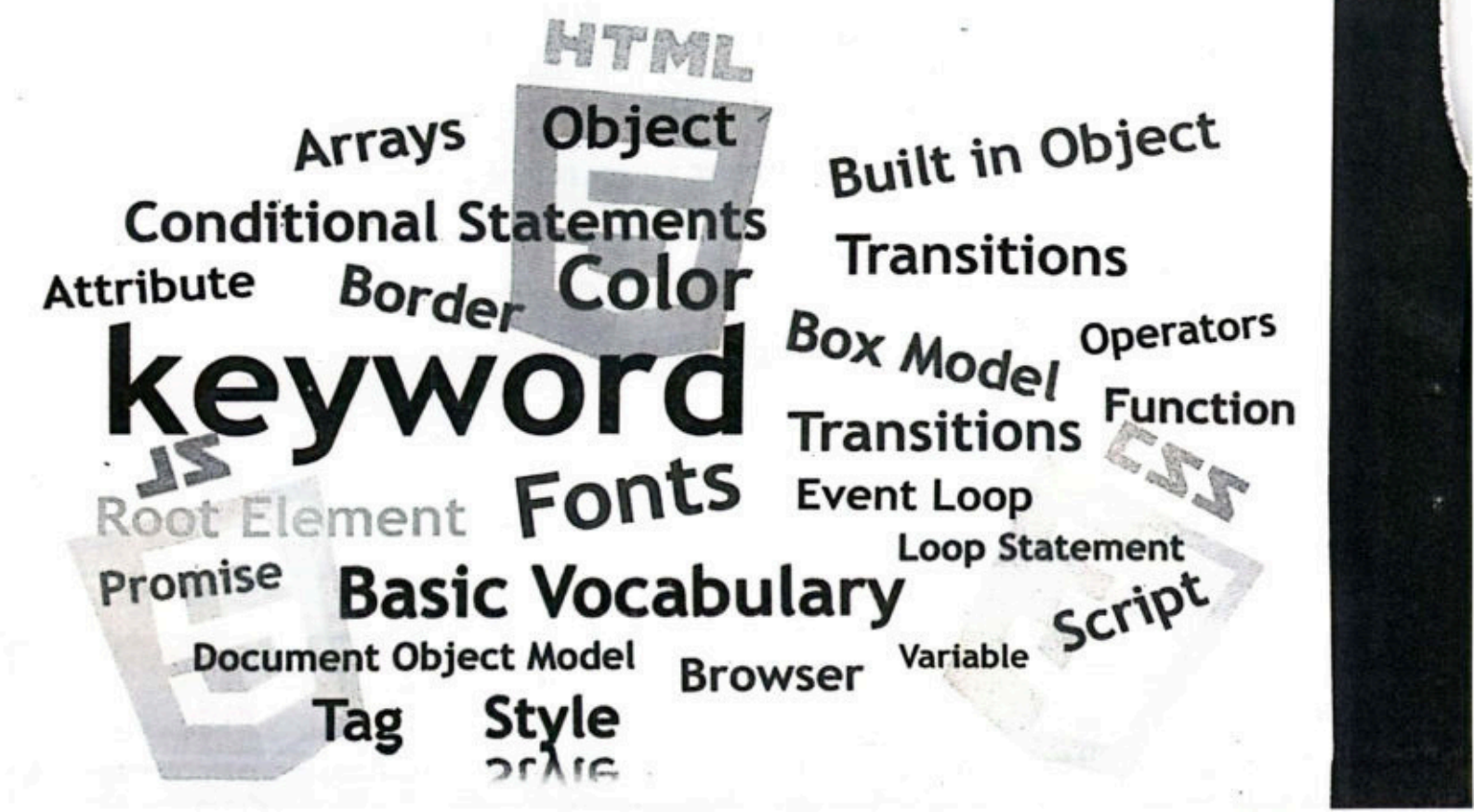
Programing Fundamentals

After completing this lesson, you will be able to:

- understand web development and differentiate between a website and a web application.
- create a static website using HTML/CSS in an appropriate environment
- create dynamic websites using JavaScript as the frontend scripting
- implement common algorithms that use sequence, selection, and repetition in JavaScript
- determine ways of debugging their code in JavaScript



3.1 Introduction

The World Wide Web's (WWW) development has made it easier to share information and data in many formats, which is becoming more prevalent and essential in our everyday lives. Documents, picture, audio, video are the main types of information but not limited to. Especially, with the rise of social media, the links and tags of information enable you to share and search related information on the internet. Additionally, the dynamic nature allows changing the contents for individuals and group of peoples. For example, the contents of the website remain same whoever visits it, but after logging in as a member customized data is shown. Hyper-Text Markup Language (HTML) is the primary language that is used for the basic website development. Though different software and tools allow you to create a website using a template, but basic knowledge of HTML is necessary to customize it according to your desire.

3.1.1 Web and Website

A document which exists and is accessible through internet is a webpage, while a set of webpages is a termed as **Website**. For example a news website Associated Press of Pakistan as shown in figure 1, has different sections and each one of them has at least one webpage in every section. To access a webpage, software namely **web browser** is used. You just provided in the Universal Resource Locator (URL) in the website which is the generally accessible address of the document. This way, web browser will locate the document and display it.

Fig-1: Website of Pakistan's Official News Agency

In case, you are unable to recall the URL of the website you are searching, help of search engines is quite useful. **Search engine** provides the service to seek relevant information based on the keywords you have entered. The search engine on the basis of keywords creates different combinations and searches the relevant information. Apart from the relevant matches, auxiliary results are also displayed to help user to explore extra information and viewpoints. Once, the results are collected by the search engine, it displays the website address and little content from every website; in a list fashion as shown in fig. 2. This way, it is easier to select the website, you are looking for.

The point to note is that every website has a 'Homepage'. As soon as a webpage opens up, its homepage is displayed and thereafter you can navigate the website to extract information. Every website and relevant pages need to be uploaded to a web hosting services, which offers web servers that are available round the clock. This way, clients and visitors can access the website on their digital devices.

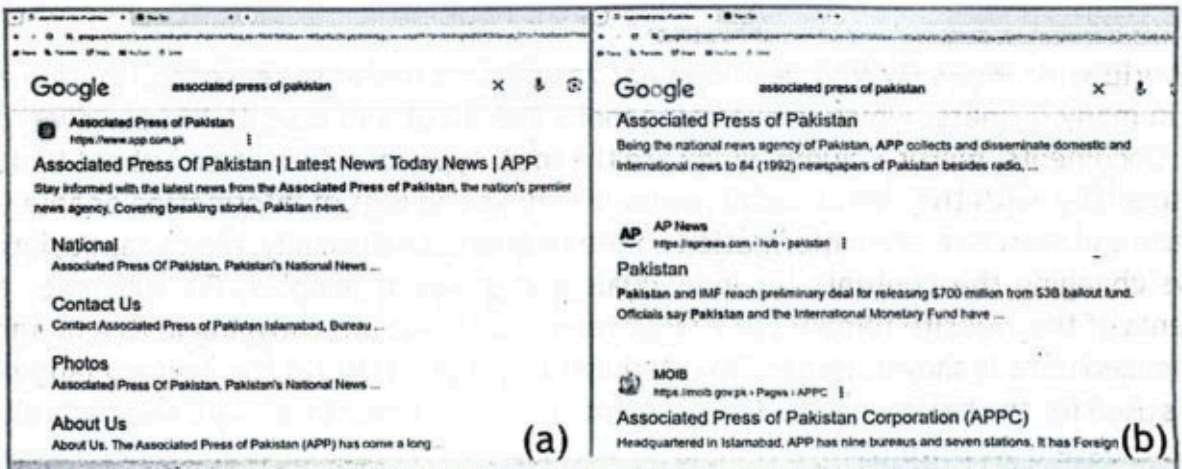


Fig.2: Search Results in a search engine (a) Primary result (b) Aux. results

3.1.2 Web Application

On the other hand, a computer program which offers a service or executes tasks via a browser and internet connection by remotely accessing a server is called web application. As shown in fig.3, through web browser, a web application can be visited by users like a Customer Relations Management (CRM) system which handles retailing, supplies, promotions, customer feedback, etc. At the backend there can be more than one server, for each type of task or a single server handling all the requests and managing it accordingly, while keeping the front-end updated. It all depends on the architecture which is being deployed to facilitate the front-end and transparent to a normal user.

3.1.3 Website Development

Website is the first step to show your presence in the digital world. Website only shares information and contents and does not allow any changes by the viewer. A website can have single or multiple pages linked together. For example, you can create a personal page of your interest highlighting your hobbies, activities and passion. You can share your ambitions and achievements. Like, if photography is your hobby, you can upload picture albums on the website on a separate page. Anyone who visits the website can view the contents but in no manner can manipulate it.

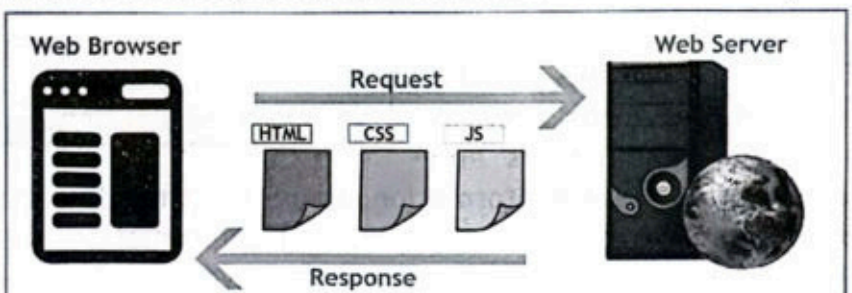


Fig-3: Communication between Web Server and Web Browser.

3.1.4 Static Website

Regardless, how much content and how many pages you increase in your website as discussed above, the contents remain unchanged; unless you change it yourself. *In other*

words the website is static and once accessed by the user; it will be loaded from the server where you have hosted it. After loading, on the user's computer the link to the server is no more required. Such static websites are easy to create and load on the client's site.

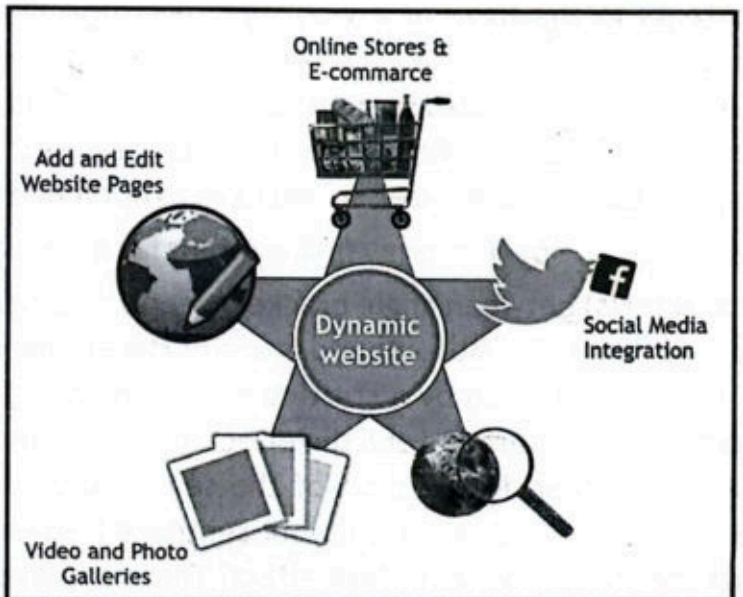


Fig-4: Features of a dynamic website of a e-store

3.1.5 Dynamic Website

A website is dynamic if the information is changed or adjusted in accordance with user input or choice. For example, alteration of background color of a webpage every time a user clicks a button. So, why a website needs to be dynamic? The intention of every website owner is to have frequent visitors and the visitors should stay long. For example, online shopping sites offer promotions and packages. So that the visitors lengthen their stay on the interested page, get awareness and shop. To achieve this, dynamicity in websites is applied using scripts, like JavaScript, python, PHP, ASP, Net, etc. In the context of online-shopping site, every member of the website can customize its webpage based on his/her interest as shown in fig. 4, while non-members and visitors get the same page to view, every time.

3.1.6 Front-End Development

A front-end of a website provides the interface which is graphical nowadays and termed as Graphical User Interface (GUI). The person visiting your website views and interacts with GUI. The front-end of a website is developed using Html, CSS and Javascript, etc. A person, who develops such front-end websites and GUIs, is termed as 'Front-end Developer'.

3.1.7 Back-End Development

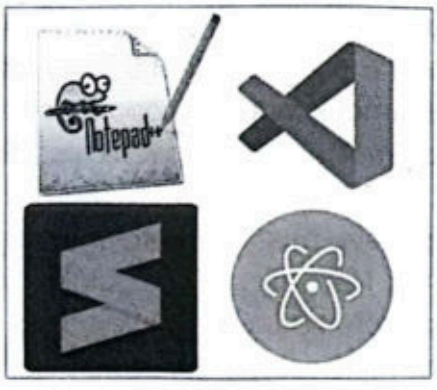
Front-end developed websites and GUIs need to communicate with the server for every event which the user generates and corresponding result is to be displayed on the front-end. This bridging between the front-end and the server is taken care by back-end development. A person who writes code about such services that are provided by the website is called a 'Back-end Developer'. Back-end development requires more knowledge and skill level in hand than front-end development, like knowledge of JavaScript, Python, PHP, ASP.Net, etc.

3.2 HTML

Hypertext Markup Language (HTML) is the language used to define and display your contents in the form of a webpage. With the help of tags, you will define different contents what they are, correspondingly HTML will display them accordingly. Html identifies and provides support for every object in a webpage on the basis of tags. For example, "This is my first attempt for a webpage." is a sentence which you want to get displayed in a web browser, so you have to put it like this:

"This is my first attempt for a webpage."

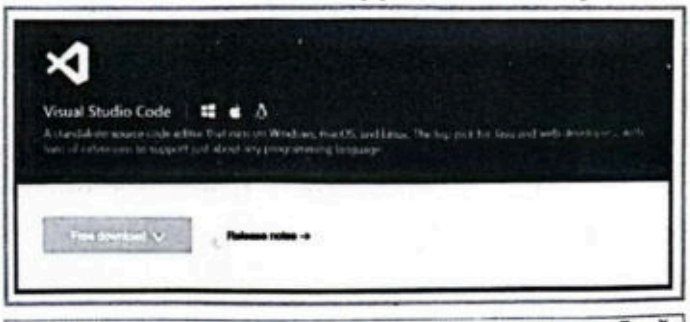
Character(s) between angle brackets '<' and '>' are called tags. The said character(s) is as per the defined HTML rules and is one of the elements that we can use. Every component of Html is identified by a starting and terminating tag. Additionally, we have to define from where to start and end the effect of every component. Therefore, we place , the starting marker, in the start of the sentence and , the closing marker, after the sentence. Now, you have put your sentence between two tags with a component to take effect. The 'p' tag is used for paragraph/sentences.

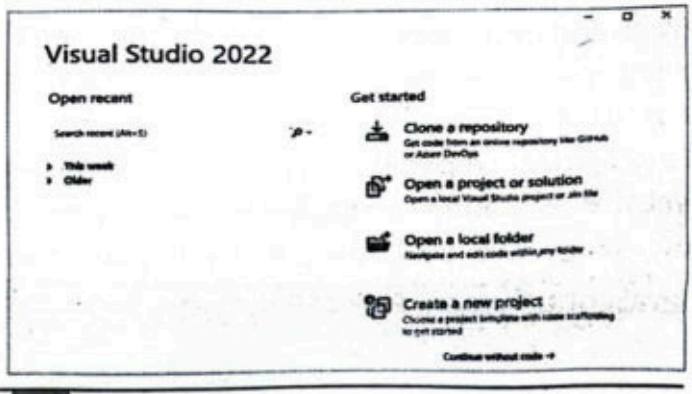


Similarly, there are tags for everything in HTML. But, the main point to note here is that you can write your html code in a notepad and run it in a browser. There are many softwares/IDE environments available that help in minimizing your coding efforts. Visual Studio, Netbeans, etc. For this chapter, we will be using the Visual Studio version 2022 environment, but the codes mentioned here will be applicable to any other environment, as well.

Installing Visual Studio

For installing the latest Integrated Development Environment (IDE) of Microsoft Visual Studio, visit https://visualstudio.microsoft.com/vs/ and download the installer. The installer is an executable file (.exe extension) and you just need to double click to start the installation.





Microsoft Visual Studio is a wide-ranging IDE which can be used for writing and running code for more than 30 languages. Therefore, it allows you to choose the language environment. Once

selected, click "install" and it will download and install the environment on your system (PC/ Laptop). After the installation, you can run the IDE from your system.

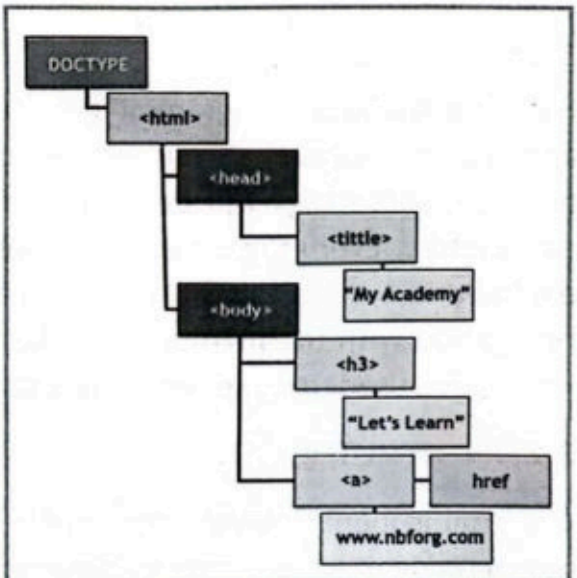


Fig-5: General DOM Tree

3.2.1 HTML Document Object Model

The Document Object Model (DOM) is a standard which provides mutual interpretation where grammar of a language can be associated with and can coexist on various operating systems. In HTML, every file is interpreted as a DOM-tree where hierarchy of the said file is defined.

Different elements which comprise of a webpage like text, images, etc. are all part of DOM. So when browser reads HTML file it creates the respective DOM automatically and correspondingly correctly displays the page.

As discussed in last chapter, a tree consists of nodes and links, so is the case here where every object and component of HTML is treated as node. Hence, the file, components, features, script and even comments which exist in a web page all are treated as objects by HTML as shown in fig 5.

'Document' comes at the top of the tree for whole webpage. Thereafter comes 'html' node which contains the HTML which has two sub-nodes, 'head' node that holds title of page and 'body' node where the content of HTML are placed like heading, paragraph, unordered list, etc.

3.2.2. HTML Comments

In programming a good practice is to use comments within the code. This way, not only for the programmer but also for others it becomes easy to understand what the code or program is about. Comments are only visible in the code and are not part of the output, as output is dependent only on the code itself. Anything between '<!--'and '-->' is treated as comments and is ignored by the browser, as shown in Fig-6.

```
1   <!-- This is a comment in HTML. It will not appear on the webpage. -->
2   <p>This paragraph will be visible on the webpage.</p>
```

OUTPUT

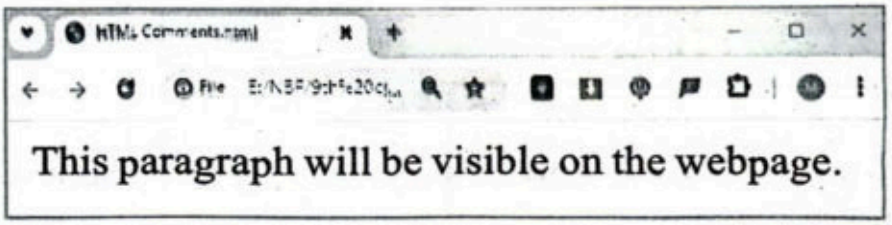


Fig-6: Displayed text in Html webpage.

3.2.3 Tags in HTML

Html is a tag based programming language so you should recognize the most important and frequent tags that you will encounter in the development. With a little bit of practice, these tags will be on your fingertips as the tags themselves are quite self-explanatory. Every tag is used for different task and is easy to recognizable as tags are enclosed in '<' and '>'. Starting tag will look like <tag> while the ending of tag will be recognized with the help of a '/' like </tag>. For example <br> is used for line break, is used to make text bold, etc.

3.2.3.1 HTML Tag

Any html is identified with the <html> tag-pair, i.e. anything written between this tag-pair is recognized as a html document. An HTML document is arranged just like a simple document that you write and prepare in Notepad, Wordpad or MS Word, etc., with headings, sub-headings, sentences, etc. Similar to <html> tag-pair, everything is characterized according to its respective tag-pair. The extension of an html file is '.html'.

3.2.3.2 Head Tag

The <head> tag-pair is defined, where tags can be placed which are not part of the main body of html, like the title of the webpage.

3.2.3.3 Title Tag

Just like every document has a name, the <title> tag-pair allows your webpage a name. In case, even if multiple pages or tabs are open in the browser, a webpage is easily identifiable. The title of the webpage is visible on the browser tag.

```
1  <html lang="en">          <!-- Start of the HTML document, language is English -->
2  <head>                    <!-- Head section containing meta information -->
3      <title> Title-comes-here </title>  <!-- Title of the webpage -->
4  </head>
5  <body>
6      <!-- Body section where webpage content goes -->
7      This text is in the body section
8  </body>
9  </html>                   <!-- End of the HTML document -->
```

Tip

To name a file, use lowercase with no spaces and clear names so that it is easier to recall what the file is used for.

OUTPUT

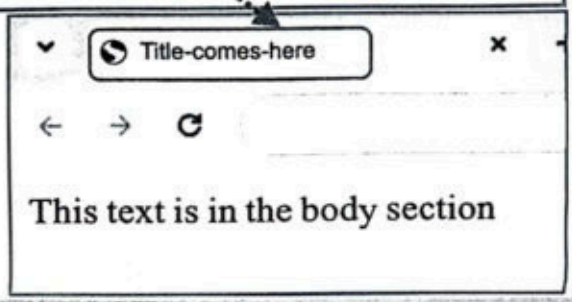


Fig-7: Title and Body Sections of a Html Webpage

3.2.3.4 Body Tag

The main part of the html is the 'body', all the main constituents html, of the document are arranged in the <body> tag-pair. Fig-7 shows placement of head, title & body tag & respective output.

3.2.3.5 Tag for Headings

The headings are of different levels provided by html and there are 6 defined levels, 1 being the largest and 6 is the smallest heading. The heading tag pair looks like <h2>...</h2>; for a second level heading. Sample of headings are shown in Fig. 8.

```
1   <html lang="en">
2   <head>
3       <title>Heading Example</title>
4   </head>
5   <body>
6       <h1>Welcome to My Website</h1>  <!-- Main heading (Largest important) -->
7       <h2>About Us</h2>               <!-- Subheading under main heading -->
8       <h3>Our Team</h3>               <!-- Further subheading -->
9       <h4>Team Members</h4>           <!-- Even smaller subheading -->
10  </body>
11  </html>
```

OUTPUT

Welcome to My Website

About Us

Our Team

Team Members

Activity

Create a webpage with heading 'My First Webpage' and a paragraph introducing yourself.

Fig-8: Different levels of Headings in Html

3.2.3.6 Tag for Line Break

To split a sentence into multiple lines a <br/> tag is used. If this tag is not used between 2 sentences and even you write the second sentence on a new line; html does not recognize this style and will put both the sentences in the same line, one after the other.

The code in Fig. 9 shows how you can setup your web page. The title of the page is between head tag-pair followed by the body and is shown on the browser's tab in the output. Note that everything written in the body is displayed on the webpage. Thereafter, you can assign suitable headings and paras, with necessary line breaks, as we applied in lines 7 and 8.

```
1   <html lang="en">
2   <head>
3       <title>Line Break Example</title>
4   </head>
5   <body>
6       <h1>Line Break Example</h1>
7       <p>This is the first line.<br>
8       Its 2nd line after line break.<br>
9       And here is the third line.</p>
10  </body>
11  </html>
```

OUTPUT

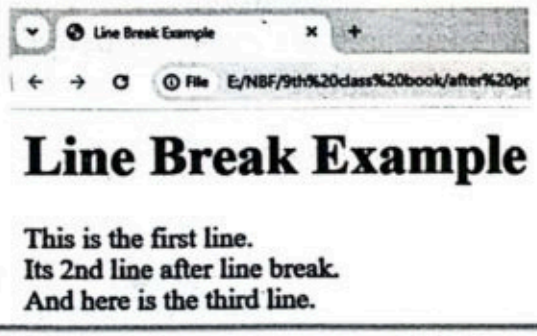


Fig-9: Line Break in Html

3.2.3.7 Tag The Text

Span is used to provide style and arrangement to a line. For multiple lines you need to assign the tag-pair on every line. Whereas, provides the same effect to a set of lines present in the page. The style and classes, etc. are generally applied in the div tag-pair. The 'i' tag-pair is used for a sentence to be in italics, like a note. The tag-pair is for emphasis, whereas <b> and tag-pair are used for display of bold and strong characters. Frequently used tags are listed in Table 3-1.

Fig. 10 shows result of various text-tags, like "i" tag-pair used in line 7, "u" tag-pair for underline, etc.

```
1   <html lang="en">
2   <head>
3       <title>Text Styling</title>
4   </head>
5   <body>
6       <p><b>Bold text</b></p>
7       <p><i>Italic text</i></p>
8       <p><u>Underlined text</u></p>
9       <p><em>Emphasized text</em></p>
10      <p><strong>Strong text</strong></p>
11      <p><sup>Superscript text</sup></p>
12      <p><sub>Subscript text</sub></p>
13      <p><small>Small text</small></p>
14  </body>
15  </html>
```

OUTPUT

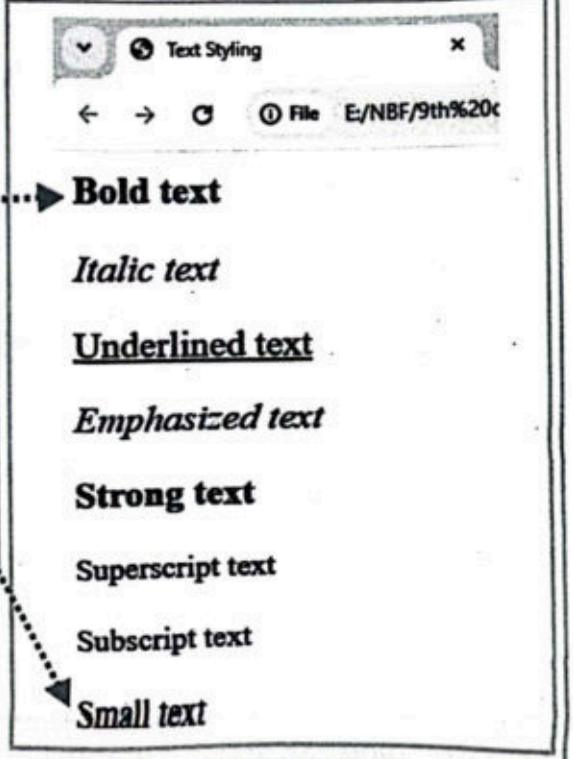


Fig-10: Text Tags

Teacher's Guide

W3Schools offers a thorough introduction to HTML which can help students understand the organization of static content by going deeper into the fundamental components of webpages.

(https://www.w3schools.com/html/)

Table.3.1: List of Frequent Tags used in Text

Tag-Pair	For the Task
p	Paragraph, sentence
b	To make characters bold
i	Text is shown in italics
em	When you need to emphasize a word but with italics.
strong	When emphasize a word but with bold
sup	Superscript, helpful in formula and footnotes
sub	Subscript, helpful in formula and footnotes
u	Underline a text
small	Smaller text size, like footnote

3.2.3.8 Image

So, as you have observed so far, that HTML provides support for a document to be presentable just like a word-editor. Additionally, we can insert an image in the webpage by assigning the 'src' meaning the path where the file is located along with the name of the file, as shown in line 7 of Fig-11. The 'alt' parameter provides description of the image. Additionally, the dimensions of the image can also be mentioned in terms of width and height of the image, otherwise it will load the image in its actual size.

```
1   <html lang="en">
2   <head>
3       <title>Image Tag Example</title>
4   </head>
5   <body>
6       <!-- Image Tag Example -->
7       <img src="html-logo.jpg" alt="Logo of HTML" width="300">
8   </body>
9   </html>
```

OUTPUT

Activity

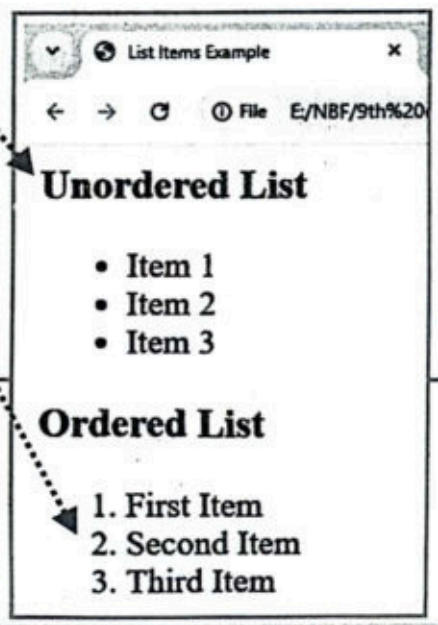
Create a webpage having an image and a caption below it.

Fig-11: Loading an image in html

3.2.3.9 Bullets & Numbering

Html treats bullets and numbering in the form of an unordered <ul> and ordered list <ol>, respectively. Bullets can be of type circle, square or disc. The numbered list have the option of numerals and alphabets to choose from, just like a word-editor.

```
1  <html lang="en">
2  <head>
3      <title>List Items Example</title>
4  </head>
5  <body>
6      <!-- Unordered List Example -->
7      <h3>Unordered List</h3>
8      <ul>
9          <li>Item 1</li>
10         <li>Item 2</li>
11         <li>Item 3</li>
12     </ul>
13     <!-- Ordered List Example -->
14     <h3>Ordered List</h3>
15     <ol>
16         <li>First Item</li>
17         <li>Second Item</li>
18         <li>Third Item</li>
19     </ol>
20 </body>
21 </html>
```



Activity

Create a list of your favorite food items and get it displayed using an ordered list.

OUTPUT

Fig-12: Bullets / Numbering in Text

In Fig-12, form lines 8-12 an unordered list is described while in lines 14-18 ordered list is stated along with respective output.

Activity

Add 5 sentence using an undordered list such that odd-numbered sentences should be bold and even-numbered should be italicized.

3.2.4 Manipulating Data with Tables

Tables in HTML

Tables are a good way to enlist data which is visually appealing. Html provides 'table' tag-pair to allocate and designate data within the table. In Fig-13 a table is created starting from line 7. Further 'tr' tag pair defines the row in a table, like on lines 8,13 and 18. The first row of the table is the header row and is generally used for headings and is defined with the 'th' tag-pair, as on lines 9-11. Whereas data is manipulated through 'td' tag-pair as on lines 14-16 and 19-21.

```
1   <html lang="en">
2   <head>
3       <title>Table Example</title>
4   </head>
5   <body>
6       <!-- Table structure -->
7       <table>
8           <tr>
9            <th>Student Name</th>
10           <th>Class</th>
11           <th>Fee Dues</th>
12          </tr>
13          <tr>
14           <td>Alia</td>
15           <td>9</td>
16           <td>200</td>
17           </tr>
18          <tr>
19           <td>Zia</td>
20           <td>9</td>
21           <td>0</td>
22          </tr>
23      </table>
24  </body>
25  </html>
```

Activity

Create a table to display a weekly schedule of your self-study.

OUTPUT

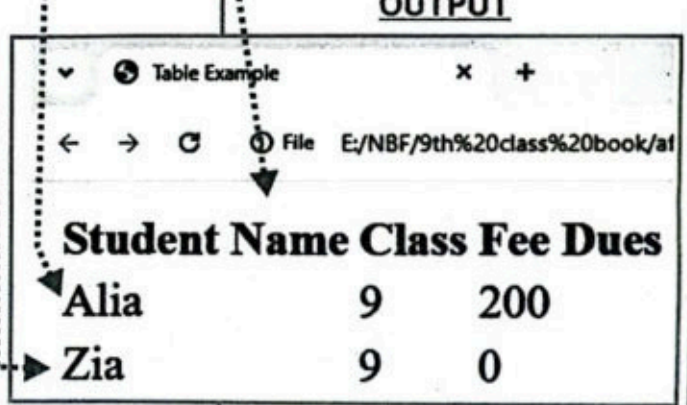


Fig-13: Table of student records with Fee submitted.

3.2.5 Links to Resources

Links are helpful components of a webpage, via which you can redirect to another webpage or a document. Links are called Hyperlinks in html with 'a' tag-pair. Hyperlinks are easy to identify on a webpage, as the mouse cursor changes as soon as the cursor touches a link element. Hyperlinks can be associated not only to text, but to images as well.

The general syntax for defining a links is like "link text", where href refers to the address alongwith the path and link-text is for user information. Target is an optional parameter but its value defines where to open the webpage. If the value chosen is 'blank', it will open in a new tab or window; however for the value of 'self' the destination address will open in the same tab or window. Fig-14 depicts result of adding the following line of code in the body of the webpage:

National Book Foundation

```
<html lang="en">
<head>
    <title>Link Tag Example</title>
</head>
<body>
    <!-- External Link -->
    <a href="https://www.nbf.org.pk">Visit National Book Foundation</a>
<br>
    <!-- Internal Link -->
    <a href="Headings.html">Go to Headings HTML example</a>
</body>
</html>
```

OUTPUT

Fig-14: Adding a Resource Link in a webpage

3.3 Cascading Style Sheets (CSS)

In the development of HTML webpage the scheme, arrangement and presentation of the whole webpage along with the components are handled by a stylesheet language. There are various stylesheet languages available like Cascading Style Sheets (CSS), Document Style Semantics and Specification Language (DSSSL), Extensible Stylesheet Language (XSL), etc. The most common and frequently used with HTML is CSS. So, this way the contents of the webpage are defined through HTML while the look of different components is handled via CSS. This way, it is easier to manage and troubleshoot your web designing code for extension and digging out the errors.

3.3.1 Decorating Tables with CSS

By using CSS, you can provide borders in table as well. For this purpose, we initially need to state the 'style' tag-pair just before the start of table and later inside the block, we need to set which style to opt for and where to apply it. As shown in fig 15 we are defining border for table, table header cells and every other cell in the table.

```
1  <style>
2      table, th, td {
3      border: 1px solid black;
4      }
5  </style>
```

OUTPUT

Student Name	Class	Fee Dues
Alia	9	200
Zia	9	0

Fig-15: Applying Borders to Table

Table 3.2 List of frequently used color names

Color	Color Name	Color	Color Name
	aqua;		navy;
	black;		olive;
	blue;		purple;
	magenta;		red;
	gray;		silver;
	green;		teal;
	lime;		white;
	maroon;		yellow;

We may add background color of the table as well, by adding the following code, as shown in fig-16 in 'style' tag-pair.

```
1  <style>
2      th, td {
3      background-color: aqua;
4      }
5  </style>
```

OUTPUT

Student Name	Class	Fee Dues
Alia	9	200
Zia	9	0

Fig-16: Provide table with a background color

3.3.2 Homepage Decor

As a first step to decor the webpage, we define 'style' tag-pair in the body in line 4 to 25 and add line 29 which results in changing the color of 3rd level headings in the body to green as shown in fig-17. Further we define general decor like background image and centrally aligned text in the body in lines 5-10. However we further distinguish h3 level heading to be displayed in green color while the normal text will be styled as 'underlined in wavy fashion', as mentioned on line 17 and 21 respectively..

```
1   <html lang="en">
2   <head>
3     <title>Styled Page with Background</title>
4     <style>
5       body {
6         background-image: url('background.jpg');
7         background-attachment: fixed;
8         display: flex;
9         justify-content: center;
10        align-items: center;
11        height: 100vh;
12        margin: 0;
13      }
14      div {
15        text-align: center;
16      }
17      h3 {
18        color: green;
19        margin: 0;
20      }
21      p {
22        text-decoration: underline wavy;
23        margin: 0;
24      }
25    </style>
26  </head>
27  <body>
28    <div>
29      <h3>Welcome to My Website</h3>
30      <p>This is a sample paragraph with a wavy underline.</p>
31    </div>
32  </body>
33  </html>
```

Do You Know

Background-Attachment
fixed: Background image remains fixed.
scroll: Background image scrolls along with the page.
local: Background image moves with the content of the element.

Do You Know

Display
block: used to span element to full width and starts on a new line.
inline: width of element is based on the content only.
flex: flexbox is to arrange items in a row or column such that items expand (flex) to fill up the additional space.

Do You Know

flex-start: Aligns items to the start of the container (default).
center: Items are centered within the container.
space-between: Distributes items evenly, i.e. first item at the start and last item at the end.

Do You Know

center: Items are set along the center.
stretch: Stretches items to fill the container (default behavior).

OUTPUT

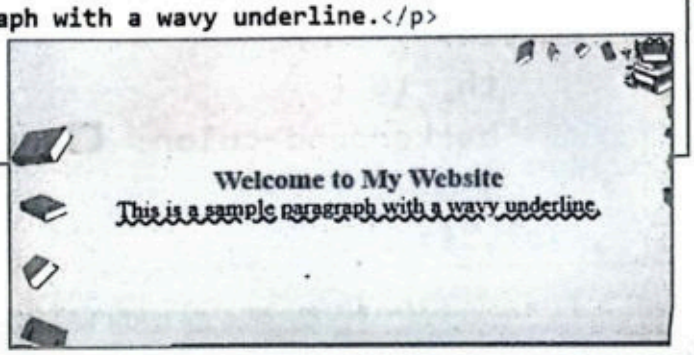


Fig-17: Code for Applying CSS Style to headings and body

3.3.3. Adding a Video Clip in Website

To add a video clip in a website, <video> tag is used where you can define the clip size and how the clip should be available at the time when website loads up. Line 6 of the code in fig-18 uses a video tag and specifies width and height of the clip in terms of pixels to adjust the size of the video clip. Additionally, the controls parameter defines that play/pause and volume controls are enabled when the video loads up as shown in fig-19 (a). In the next line, the source and type of the video are specified. Though, Html supports many formats but among different browsers the most commonly supported video type is Mp4.

```
1   <html>
2    <body style="text-align: center">
3        <h2 style="color: green">Pakistan's First Ever Win in
         International Football</h2>
4        <p>Congratulations !!!</p>
5        <video width="500px" height="500px" controls>
6            <source src="fifa-win.mp4" type="video/mp4" />
7        </video>
8    </body>
9   </html>
```

Fig-18: Sample code to load a video clip in a website

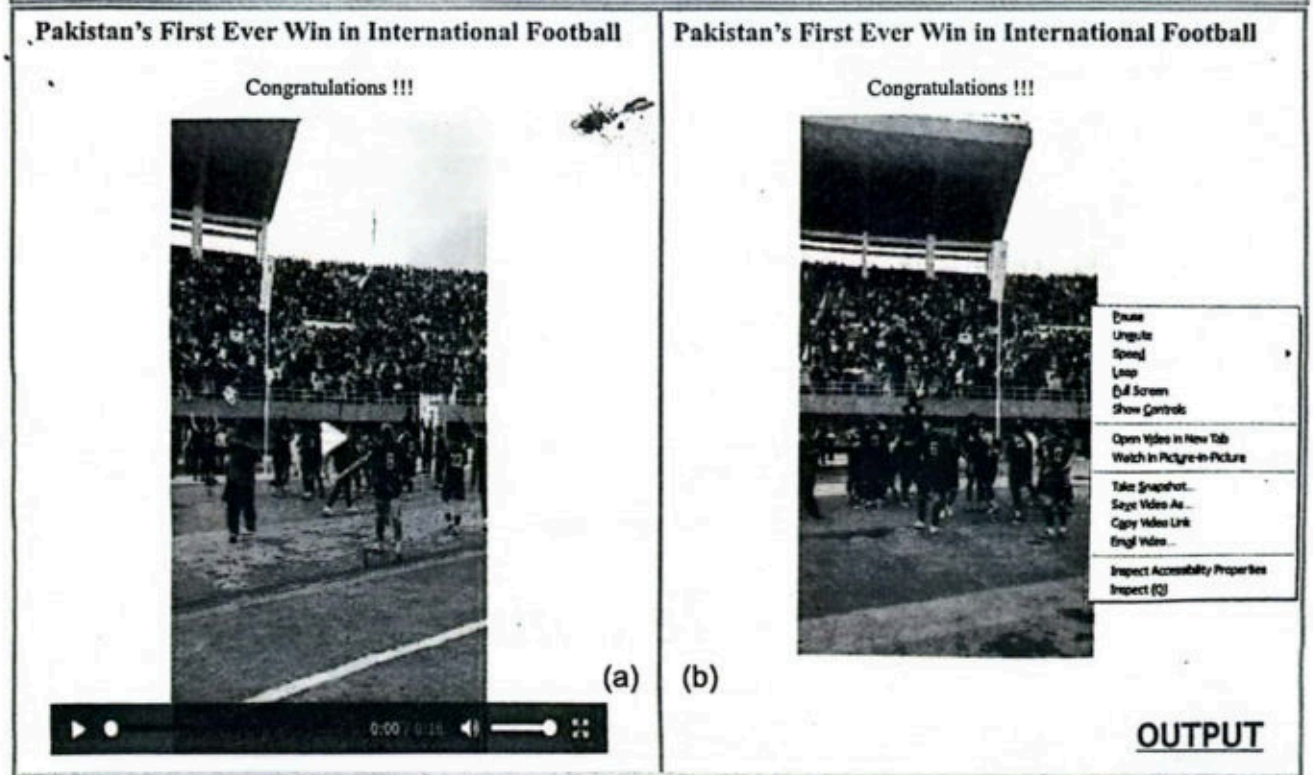


Fig-19: Loading of a video clip with (a) control parameter (b) autoplay parameter

Alternatively, autoplay parameter can be used instead of controls. Autoplay automatically loads the video clip as the webpage loads but does not provide any controls for volume or play/pause. Autoplay can be used by specifying additional features like muted, looped, etc. When muted is used, the clip loads but volume is muted and needs to be unmuted manually while looped will keep the clip running, again and again in a loop, unless it is interrupted manually. Options for manual intervention are accessible by right-clicking the mouse. Fig. 19 (b) is the output of following code:

Tip

A good programming practice is the provision of additional video formats like Ogg in video tag-pair too, like: <source src="fifa-win.mp4" type="video/mp4" />

```
<video width="500px" height="500px" autoplay muted>
```

3.3.4 Ways to Use CSS in HTML

There are three ways, via which we can use CSS styles in our HTML webpage.

Inline CSS:

Any CSS attribute that we want to incorporate can be added using a HTML tag (like the ones, we have covered in the last section) and incorporated in the body section, as shown in fig-12.

```
1   <html lang="en">
2   <head>
3     <title>Inline CSS Example</title>
4   </head>
5   <body>
6     <!-- This is a paragraph with inline CSS for styling -->
7     <p style="color: red; font-size:40px; font-style:italic;
      text-align:center;">
8       My Teaching Academy
9     </p>
10  </body>
11  </html>
```

OUTPUT

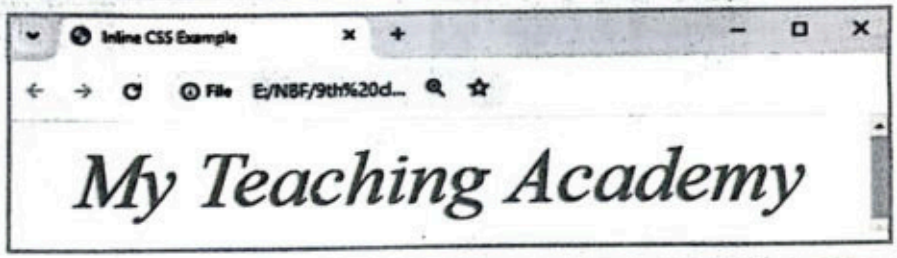


Fig-20: Inline CSS sample

Embedded (Internal) CSS

Instead of assigning styles for every heading and other component at the time of its first occurrence in the code, a better approach is to outline all the styles in the header under the tag-pair of 'style' as shown in fig-21. This way, all the presentation related CSS code is separated and do not indulge with the already written HTML code. Additionally, change in one line in the CSS section will be reflected throughout the respective components.

```
<html lang="en">
  <head>
    <title>Embedded CSS Example</title>
      <style>
      body {
        color: blue;
          }
                h2 {
                  color: red;
        font-size:30px;
          font-style:italic;
        text-align:center;
      }
                  </style>
    </head>
<body>
  <h2>My Teaching Academy</h2>
  <p>My Teaching Academy</p>
</body>
</html>
```

OUTPUT

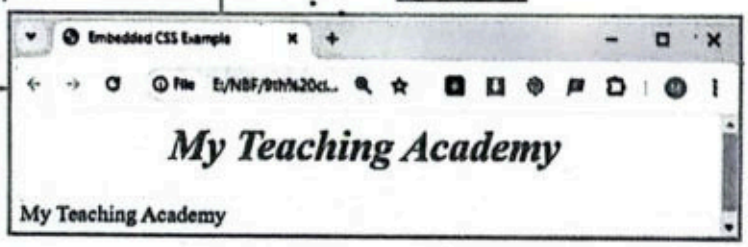


Fig-21: Embedded CSS sample

External CSS

Alternatively, a file with extension '.css' can be made and all relevant CSS code according to your schema can be present there. Once, the contents of HTML are finalized, just attach the CSS file in the head portion of HTML by passing the link. External CSS are used with large projects, like in commercial purposes.

```
<link rel="mystylesheet" href="my_own_SS.css" />
```

NOTE: The priority of Inline is highest, followed by embedded styles and lastly the attributes of external are considered; if all three are present in a webpage.

3.4 Java Script

Javascript is an exciting language primarily used in development of web pages and scripts. It does not consume much of memory and that is why it is used at the client-end in developing websites, for making pages dynamic. It easily works with programming languages like Java and HTML, on any operating system.

3.4.1 Let's Meet Javascript

Javascript code can be embedded in HTML with starting and ending tag of <script>, in a webpage. There is no limitation of where to place the code inside a HTML file. For example, the following Javascript code embedded in the body, displays a sentence (string) using 'document.write()' function, as shown in fig-22.

```
1. <html>
2.  <body>
3.   <script language="javascript" type="text/javascript">
4.     document.write ("Let's Meet Javascript!")
5. //   document.write ("    Let's Meet Javascript    !        ")
6.   </script>
7.  </body>
8. </html>
```

OUTPUT

Fig-22: A simple JavaScript Text is displayed.

In the sentence, displayed above we put additional spaces and tabs, which javascript simply discarded. For example, we will get the same output if the statement is like the one commented on line 5. Try it for yourself.

In programming everything that a user or another program does with your program that can be sensed and triggers some task to be done, is called an 'event'. Events are important and critical in the functioning and flow-control of your program. Similarly, in website development, easiest way to introduce dynamicity is to allow some event to occur and respond accordingly. Alert is the commonly used functionality that

Tip: Javascript like C/C++, uses '//' for line comments and '/* .. */' for block comments.

Javascript provides to inform user, about the result of his action or notification. Event based code like 'onclick' are put between the start and end of <head> tag; unlike the above example, where we put our code within the <body> tag. The scenario in fig-23, allows the message to be displayed when the button is clicked. Functionality of button is added on lines 4-6, i.e. the message to be displayed when the buttons gets clicked.

```
1. <html>
2.  <head>
3.   <script type="text/javascript">
4.      function msgSure(){
5.          alert("Are You Sure???")
6.      }
7.   </script>
8.  </head>
9.  <body>
10.    Do You Mind, CLICKING on the Button -
11.    <input type="button" onclick="msgSure()" value="Be Sure" />
12. </body>
13. </html>
```

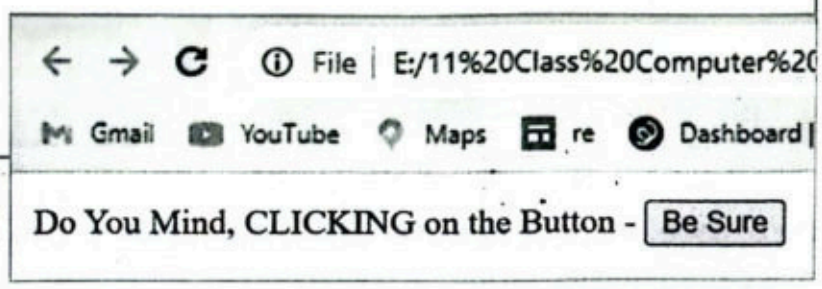
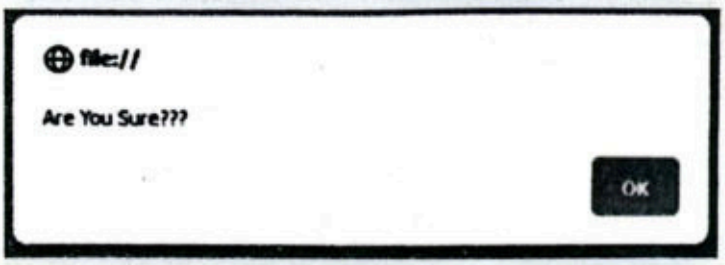


Fig-23: Adding an event via a button.

And after the 'Be Sure' button is pressed, the generated alert is as follows.



You can try for yourself, by replacing different messages and even arithmetic operations like:
alert(2+2)
which results as:

3.4.2 Variables

A variable is an entity that stores some value for later use. In mathematics, a variable is generally represented by a single character, but not limited to. Similarly, in programming languages, variable should be named in a meaningful manner. Additionally, the type of values the said variable can store is another important aspect. Such that the developer after defining and assigning a variable, is later able to recall about the task and type of values the said variable holds. The basic value that a variable can hold in JavaScript is either a number or set of characters (called string) or a Boolean which is either 'true' or 'false'. It is critical to note that variable naming convention does not support a number to be the first character of the variable name. Since, Javascript is case sensitive; therefore 'reward' and 'Reward' are two different variables. Table 3-3 enlists basic datatypes of JavaScript.

	Sample Value
Number	7 3 5 9
String	J A V A
Boolean	TRUE

Table3-3: Basic datatypes supported by Javascript with sample values.

Teacher's Guide

Experienced users may choose feature-rich code editors like Visual Studio Code, but beginners can start with simple text editors like Notepad. Real-time collaboration is made possible by online tools such as JSFiddle (https://jsfiddle.net/).

A variable is declared with the 'var' keyword and multiple variables can be declared in the same line of code, too. The first ever assignment of a value to a variable in the life span of program is called 'initialization'. A good programming practice is to declare and initialize the variable at the same time.

```
1.  <html>
2.  <body>
3.    <script type="text/javascript">
4.          var a,b,reward;
5.          reward = 5555;
6.          a = reward % 10;
7.          b = reward / 10;
8.          document.write(a, " ---- ",b);
9.    </script>
10. </body>
11. </html>
```

OUTPUT

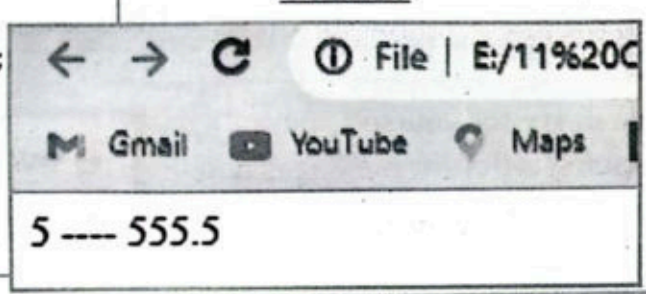


Fig-24: Multiple variables.

In the following line of code in fig-24, two variables are defined. One is a boy's name in the variable 'name' while other is 'reward' having value 5000. So, it displays the result in browser.

```
1. <html>
2.  <body>
3.   <script type="text/javascript">
4.          var name = "Ahsan";
5.          // document.write(reward);
6.          var reward;
7.          reward = 5000;
8.          document.write(name, " gets a reward of Rupees ", reward);
9.   </script>
10. </body>
11. </html>
```

OUTPUT

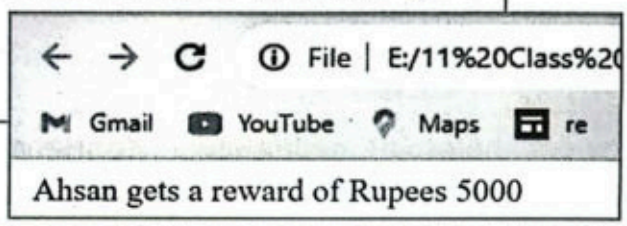


Fig-25: Use of Variables

Activity

Create a list of subjects like 'HTML', 'CSS' and 'JavaScript' when clicked a new tab opens up. Relevant keywords and a logo (or image) should be displayed on the new tab.

Now, let's amend the same program for user input, where we want to change the value 'reward'; by taking input from the user. For input from the user we can use the prompt() function, which pops up a message window like that of alert() function. This input will be assigned to a variable at the backend which can be used, later. A sample code is as follows.

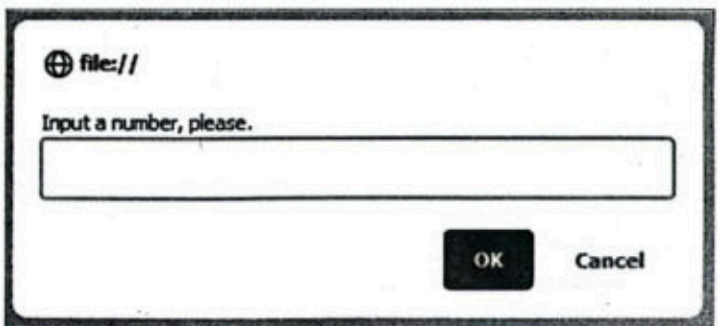


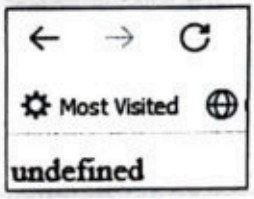
Fig-26: An Alert Window for User Input.

```
1. var ip = prompt("Input a number, please.");
2. document.write("Input from the user was, number :", ip)
```

You can replace these lines in the previous example and will get the outputs, as shown in fig-25.

It is important to note that sequence of instructions in programming matters. For example let's look at the following code, where we defined two variables and keeping the good practice in mind, declared and initialized variable name in line 7. However, just to highlight how things can be alternatively done, we declared the reward variable in line 8 but initial value as assigned to it in the next line. In line 12 we print the statement using both the variables, as shown in fig-26.

Note that, if you take line 7 and put after line 11, then the following message is displayed.



This is due to the reason that the variable is defined after the write statement is using the variable. Unless, the variable is declared and holds a value, only then it can be used as per the programming sequence. Hence, sequence of instructions is important and though you might have written a program syntax-wise correctly, but the change of sequence might lead you to unexpected bugs.

3.4.3 Operators

Javascript supports arithmetic operators to be used which are Addition (+), Subtraction (-), Multiplication (*) and Division (/). Other than this, the Modulus (%) operator can also be used which gives remainder of a division operation.

The code shown in fig-27, takes 3 variables namely, 'a', 'b' and reward. Only reward is initialized through a constant value in line 5. In line 6 and 7, remainder of reward divided by 10 is stored in 'a' and 'b' is assigned the value when reward is divided by 10, respectively.

```
1.  <html>
2.  <body>
3.    <script type="text/javascript">
4.          var a,b,reward;
5.          reward = 5555;
6.          a = reward % 10;
7.          b = reward / 10;
8.          document.write('Value of variable a is:',a);
9.          document.write("<br />")
10.         document.write('Value of variable b is:',b);
11.   </script>
12. </body>
13. </html>
```

OUTPUT

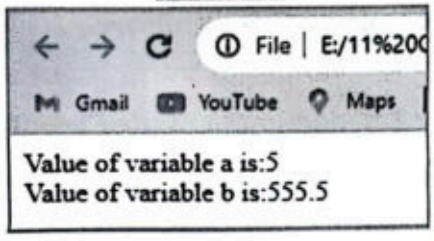


Fig-27: Handling and Printing multiple variables.

3.4.4 Conditional Statement

Conditional or selection statement is an essential part of the program where amongst choices, the program chooses on the basis of some constraint. Applying an 'if' statement before one or more lines of code on the basis of some condition is met makes a typical selection scenario. That is, if the condition is met, then those line(s) will be executed otherwise skipped.

Operator	Name	Example
==	Is equal	x == y
!=	Is not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Table:3.4: JavaScript Conditional Operators

Now, to check the condition, Javascript provides set of comparison operators to be used for evaluating the condition. The conditional operators are listed in table.

For example, the admission office of a Montessori school checks the age of a child, if the kid is of at least 4 years old, then admission is granted. So, the code should look something like, as shown in fig-28.

```
1. <html>
2.   <body>
3.     <script type="text/javascript">
4.           var kid_age = 5;
5.           if( kid_age > 3 )
6.           {
7.              document.write("<b> Admission Granted !!! </b>");
8.           }
9.     </script>
10.  </body>
11. </html>
```

OUTPUT

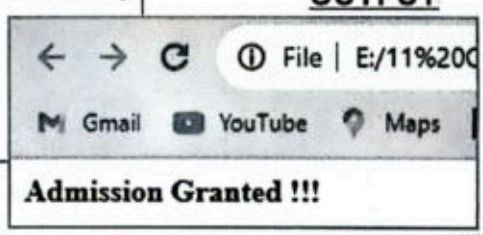


Fig-28: 'if' statement.

You may check it for different values of the variable and also for different comparison operators.

A better notion is to align both the scenarios, i.e. if condition is met and vice versa. This is achieved using an 'if-else' statement, as shown in Fig-29. This way, either of the two situations will definitely happen.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.          var kid_age = 3;
5.          if( kid_age > 3 ){
6.            document.write("<b> Admission Granted !!! </b>");
7.          }
8.          else {
9.            document.write("<b> Age Requirement is not met !!! </b>");
10.         }
11.   </script>
12.  </body>
13. </html>
```

OUTPUT

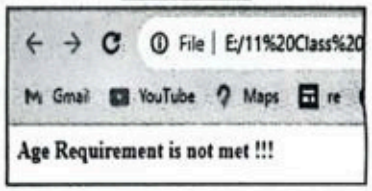


Fig-29: 'if-else if' statement.

There are scenarios where more than two possibilities exist and for that reason, we can modify our selection statement to be an 'if-else if-else' statement. This way, the set of conditions apply, first with 'if' and thereafter with 'else if' statements. For any other condition that has not been catered for, 'else' will take care of it. In Fig-30 the previous code is extended by checking multiple conditions.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.    var kid_age = 3;
5.    if( kid_age > 3 ){
6.      document.write("<b> Admission Granted in Nursery Class !!! </b>");
7.    }
8.    else if (kid_age > 4){
9.      document.write("<b> Admission Granted in KG Class !!! </b>");
10.   }
11.   else if (kid_age > 6){
12      document.write("<b> Admission Should be taken in Primary School !!! </b>");
13.   }
14.   else {
15.     document.write("<b>  Kid is Underage !!! </b>");
16.   }
17.   </script>
18.  </body>
19. </html>
```

OUTPUT

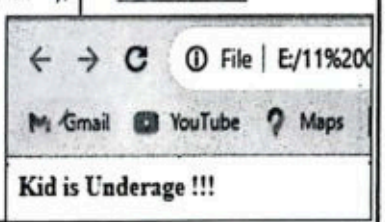


Fig-30: 'if-else if-else' statement.

Activity

Take a number as input from the user and write a program to calculate the grade of the student in a particular subject. If the number is greater than equal to 90 then assign A+ grade, for 80 and above but below 90, A-grade is designated. Similarly in the range of 70-79, B grade needs to be awarded. If the number lies between 60-69, C grade should be printed. In case the number is between 50 and 59, D grade and less than 50, F grade should be displayed on screen.

3.4.5 Iterative Statement

Iterative statement like 'For Loop' is used to get similar kind of task done. Rather than writing the same line of code multiple times, the same task is achieved in much lesser line of code. The 'for loop' works on the basis of an index, which you can initialize in the loop. Next is the terminating condition which needs to be set for the loop to terminate. Lastly, step-size needs to be defined that how many steps the index will take after each iteration; till the terminating condition is met. In the following example 'for loop', an index is initialized to 0, the value of index will increment with 1 and loop will execute till index value remains less than 10. Fig-31 depicts start, end and iteration of a for-loop.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.          var index;
5.          document.write("For-Loop Starts After This ... <br />");
6.          for(index = 0; index < 10; index=index+1){
7.                document.write("Index No. : ", index, "<br />");
8.          }
9.          document.write("For-Loop Stopped!");
10.   </script>
11.  </body>
12. </html>
```

OUTPUT

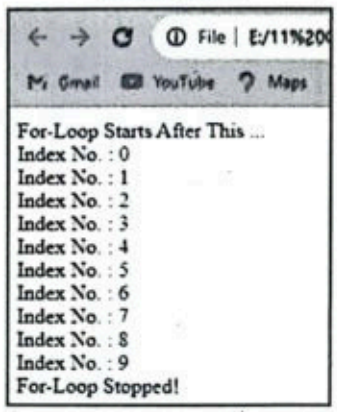


Fig-31: A simple For-Loop, shows how index value increases.

Alternatively, we can decrease the index value and set the condition accordingly; in the code of fig-32 we increased the step size, too.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.          var index;
5.          document.write("For-Loop Starts After This ... <br />");
6.          for(index = 10; index > 0; index=index -2){
7.                document.write("Index No. : ", index, "<br />");
8.          }
9.          document.write("For-Loop Stopped!");
10.   </script>
11.  </body>
12. </html>
```

OUTPUT

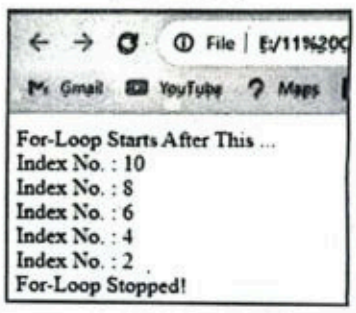


Fig-32: For-Loop with step-size of 2.

3.4.6 Nested Loops

Multiple iterative tasks, if can be related then they can be incorporated in such a way that one loop can reside inside the other and are termed as 'Nested Loop'. In nested loop, initially the outer loop will start and then the inner loop will run and finish. Thereafter, the index value of outer loop will increment and the inner loop will start and end again, and so forth and so on; till the outer loop terminates. Fig-33 demonstrates a nested loop.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.       document.write("About to Enter the Loops !!!<br /> ");
5.          for (var i = 0; i < 5; i++)
6.           {
7.                document.write("<br /> The Outer Loop: ",i,"<br />");
8.                for (var j = 0; j < 3; j++)
9.                {
10.                   document.write("Inner Loop: ",j,", ");
11.               }
12.     }
13.       document.write("<br /> <br /> <br /> Loops Terminate !!!<br /> ");
14.   </script>
15.  </body>
16. </html>
```

OUTPUT

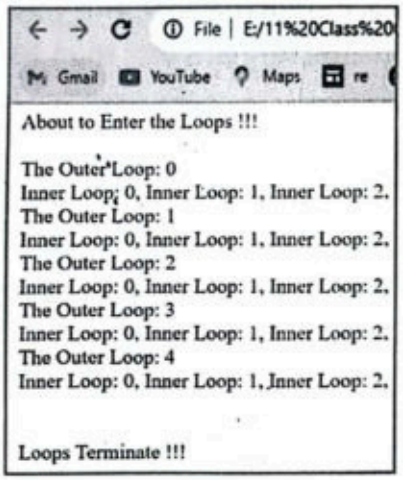


Fig-33: Nested Loop

In a similar fashion, the code in fig-34 prints Mathematical Tables from 2 to 5. The outer loop assigns the value, for which the table is required while the inner loop prints the table.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.       for (var i = 2; i <= 5; i++)
5.       {
6.            document.write("<br /> Table of: ",i,"<br />");
7.            for (var j = 1; j <= 10; j++)
8.            {
9.               document.write(i,"x",j,"=",i*j,", ");
10.           }
11.      document.write("<br /> <br /> ");
12.      }
13.
14.   </script>
15.  </body>
16. </html>
```

OUTPUT

Fig-34: Printing Mathematical Tables from 2-5.

3.4.7 Arrays

An array is a datatype which can hold a number of homogenous set of elements. Such that we do not need to define multiple variables of the same type like num1, num2, ... num25. Instead, we can declare an array which contains 25 values. This way, we can directly access any value just by passing the respective array-index number. It is trivial to note that array-index starts with 0. Declaration of an array is of the form as shown in line 7 of fig-35.

```
1.  <html>
2.   <head>
3.    <title>Arrays in JavaScript</title>
4.   </head>
5.   <body>
6.    <script type="text/javascript">
7.          var Arr = [4, 2, 5, 1, 3];
8.          document.write("Array is:", Arr);
9.    </script>
10.  </body>
11. </html>
```

OUTPUT

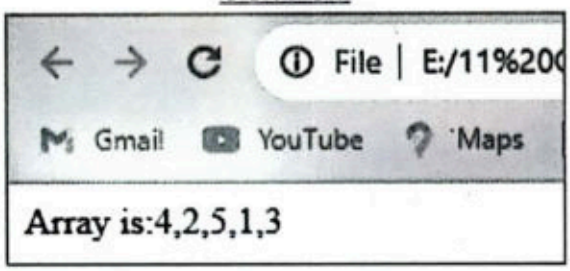


Fig-35: Array declaration and initialization in a single line.

Alternatively, we can declare a null array first and assign values to it later, as shown below:

```
1.  <html>
2.   <head>
3.    <title>Arrays in JavaScript</title>
4.   </head>
5.   <body>
6.    <script type="text/javascript">
7.          var Arr = []
8.          Arr[0] = 4;
9.          Arr[1] = 2;
10.         Arr[2] = 5;
11.         Arr[3] = 1;
12.         Arr[4] = 3;
13.         document.write("Elements of Array are, after declaring a null array: ", Arr);
14.   </script>
15.  </body>
16. </html>
```

OUTPUT

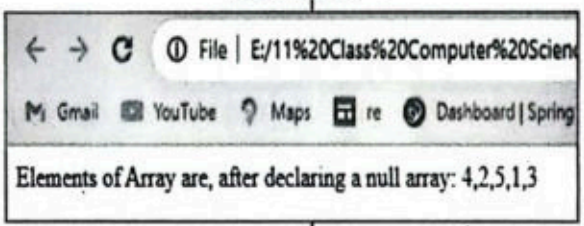


Fig-36: Initialization of an array, element by element.

So, rather than assigning values one by one, as highlighted in the lines 8-12 in above code, we can alternatively use for-loop to populate an array. The code in fig-37 initializes a null-array on line 7. In the next couple of lines a for-loop is taken into account in which user input is taken, via prompt() function.

```
1. <html>
2.  <head>
3.          <title>Arrays in JavaScript</title>
4.  </head>
5.  <body>
6.   <script type="text/javascript">
7.          var Arr = []
8.          for (i =0; i< 5; i++)
9.                  Arr [i] = prompt("Input Array element");
10.         document.write("Elements of Array using For-Loop are, after declaring a null array: ", Arr);
11.  </script>
12. </body>
13. </html>
```

OUTPUT

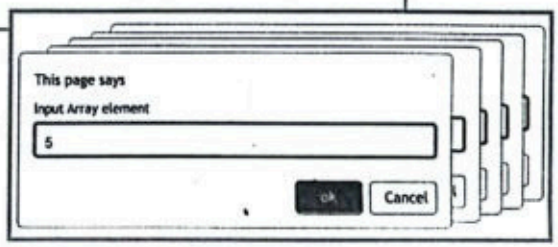


Fig-37: Use input for multiple values

Once, the user input is taken and the loop terminates, in line 12 the elements of the array are displayed as in pervious example.

Elements of Array using For-Loop are, after declaring a null array: 5,3,7,1,9

3.4.8 Functions

Function is a set of line which occurs in the code quite often that can be segmented once, and called again and again. This way, rewriting the same set of code for similar results can be eliminated. Through functions, different sets of code can be separated resulting in fewer lines of code assigning them meaningful names based on their functionality. This leads to efficiently managing a large computer program. Programming languages provide built-in such functions like earlier we have used functions prompt(), input(), etc. Whenever the function is called, the caller does not necessarily need to know the code behind that function, to use it.

A function has a name through which it is identified and called. Additionally, a function can have arguments which are variables local to that function and their life span is limited to the said function. Variables outside functions are global variables and can be accessed anywhere from the program. Recall the earlier program of fig-23 where we defined a function, namely msgSure() and whenever the button is pressed an alert pops up asking "Are you sure?". In line 4 of the said code, we used the 'function' reserved word to define and named it. However, on line 11, we called the same function just with its name.

We extend the same code, and provide it with values which are assigned to arguments of the function. And thus, we can use them in the function, as shown in Fig-38.

```
1.  <html>
2.   <head>
3.    <script type="text/javascript">
4.          function msgSure(currency, amount) {
5.                alert("Are You Sure, you gave me "+currency+"."+amount,"???")
6.          }
7.    </script>
8.   </head>
9.   <body>
10.         Do You Mind, CLICKING on the Button -
11.         <input type="button" onclick="msgSure('Rupees', 5000)" value="Be Sure" />
12.  </body>
13. </html>
```

OUTPUT

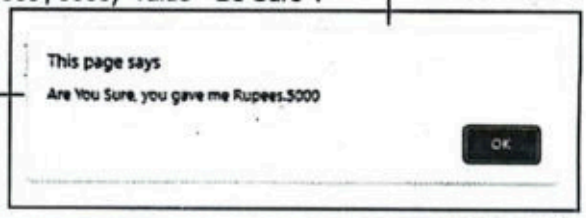


Fig-38: A function with arguments

Lastly, lets define another function calcBill() which is similar to the above as it accepts 2 arguments, namely bill and 'amount_rcvd'. The function subtracts bill from 'amount_rcvd' and returns the result. An important point to note in fig-39 is the lines 5 and 10, where the former prints inside the function and latter displays the result outside the scope of the function. Similar approach can be used for testing the values of a function, when frequent occurring set of codes are selected to form a new function.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.          function calcBill(bill, amount_rcvd) {
5.                document.write("Your Balance is: ");
6.                return(amount_rcvd - bill);
7.          }
8.
9.          var balance =calcBill(2345, 5000);
10.         document.write(balance);
11.   </script>
12.  </body>
13. </html>
```

OUTPUT

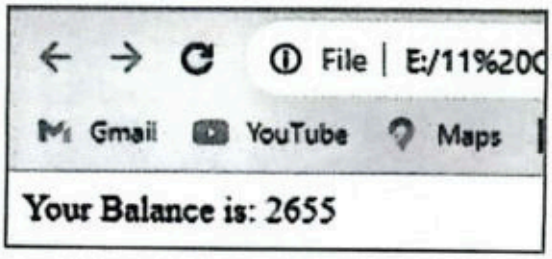


Fig-39: Function returns value.

3.5 Debug The Code

Debugging refers to locating an error or a bug in the code. In Visual Studio .Net, Select Run from the menu and apply choose **'Run Debugging'**. Thereafter, you will get the Debugging menu with some buttons and a debug console. Rather than every time accessing the menu and choose options from there to debug the code, the debugging menu is *quite* handy and allows you to **'Pause/Continue'** your debugging code with the first button.

'Step Into' will go through the code line by line, as the program will normally execute, so that you have a thorough tracing capability of the code.

To emphasize on lines where you want to check the values of variables, you can assign a breakpoint as shown in fig-40.

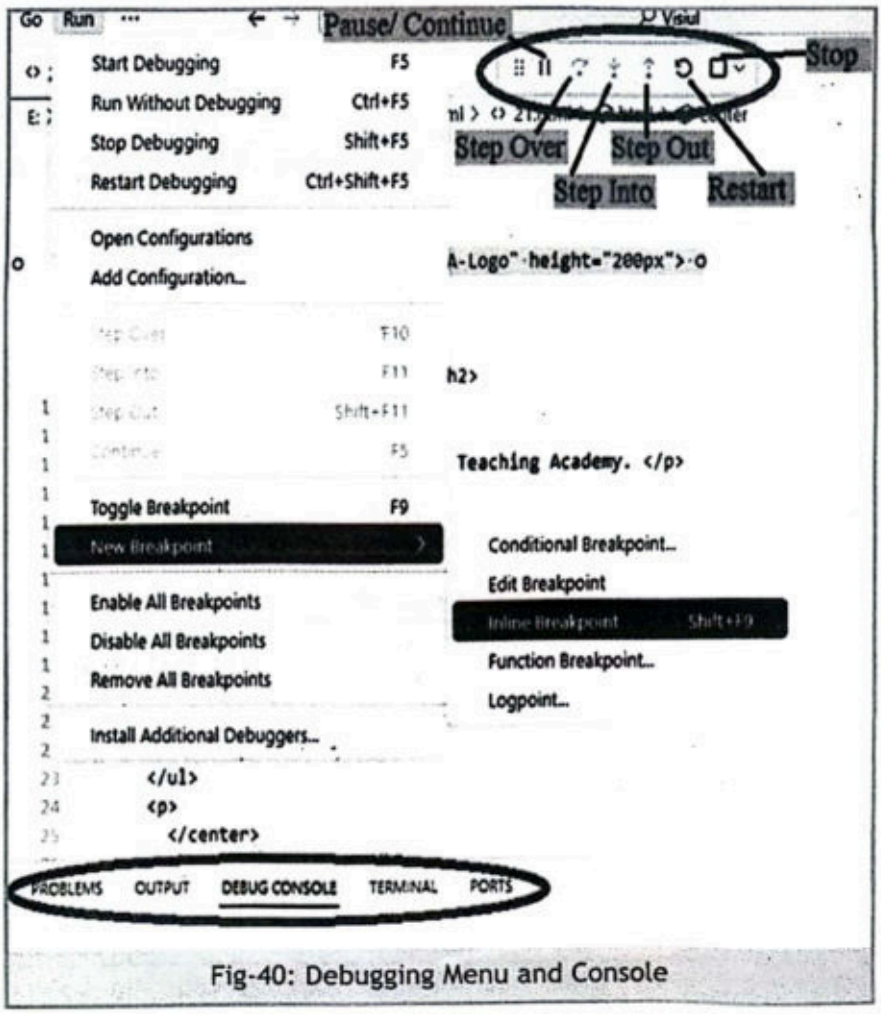


Fig-40: Debugging Menu and Console

You can assign multiple breakpoints for the sake of traceability. In Visual Studio .Net, a breakpoint is shown as a red dot at the start of line, as shown in fig-41. While the code is executed in debug mode, the execution halts at every breakpoint such that behavior of code according to the values can be analyzed. This way, it becomes easier to trace a bug and fix it accordingly.

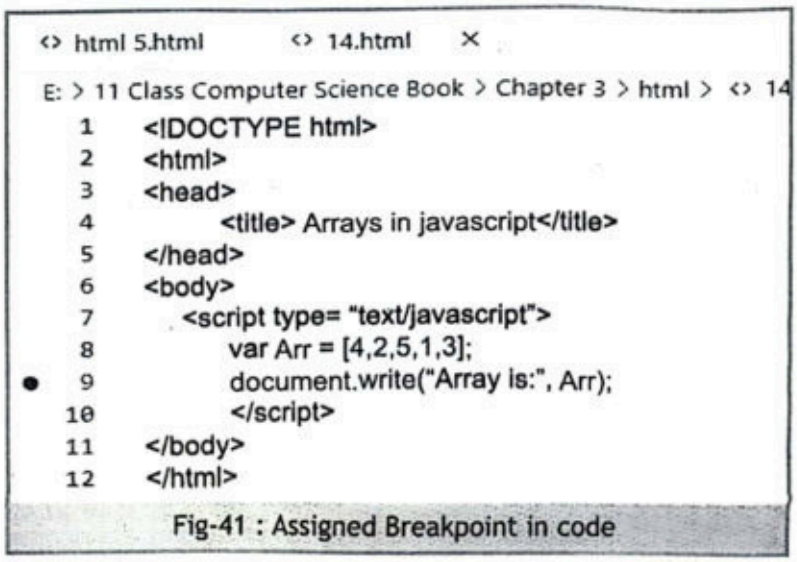


Fig-41 : Assigned Breakpoint in code

3.6 Create a Dynamic Website

If a school wants to display the result of each student on their website, then they have to create a different static webpage for each student. And as the students increase, so does the handling and managing of these webpages. The solution is to provide dynamic website such that contents change according to the user input. Such dynamic website can be created via JavaScript along with Html and CSS.

Recall Fig-31, where we printed index number via a loop from 0-9. We extend the same code and the revised code is shown in fig.42 (a), where the initial 9 lines are the same. At line 10, a new function is added namely 'descOrder()' which when is called prints the index in descending order. Additionally, the program has an id of 'dynamic content' and a button which when pressed calls the descOrder() function. A dynamic content is component/ portion of the website that alters based on user input.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.          var index;
5.          document.write("For-Loop Starts After This ... <br />");
6.          for(index = 0; index < 10; index=index+1){
7.                document.write("Index No. : ", index, "<br />");
8.          }
9.          document.write("For-Loop Stopped!");
10.         function descOrder () {
11.               for(index = 10; index > 0; index=index -1){
12.                     document.write("Index No. : ", index, "<br />");
13.         }}
14.   </script>
15.   <p id="dynamicContent">Click the button to change output to Descending Order.</p>
16.   <button onclick="descOrder()">Descending Order</button>
17.  </body>
18. </html>
```

Fig-42 (a): Code to create dynamic websit

As the webpage loads, the index numbers are printed in ascending order and button with a message is visible as shown in fig- 42(b). For the sake of understanding and tracking the start and end of the for-loop are marked with sentences on line 5 and 10. On pressing the button, the descOrder() function is called as can be seen on line16. The function prints the index numbers in reverse order, as shown in fig- 42(c).

Moreover, after line 13 in the code, if we add the following line of code:

```
document.body.style.backgroundColor = "peachpuff";
```

Save the file, refresh the webpage and press the button, then the output changes to fig-42(d) where the background color of the webpage also changes. The document refers to the whole HTML page while body relates to the content visible and using style property we can dynamically set properties of the element, like changing the background color of the page.As an activity, you may try to add another button and separate the functionality of changing the background color.

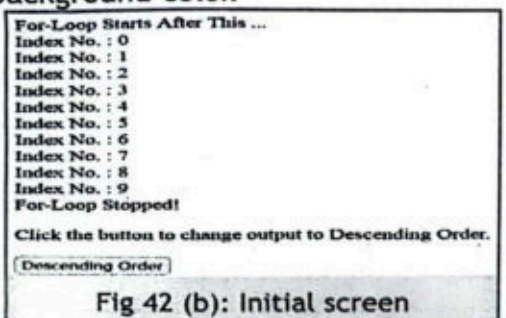


Fig 42 (b): Initial screen

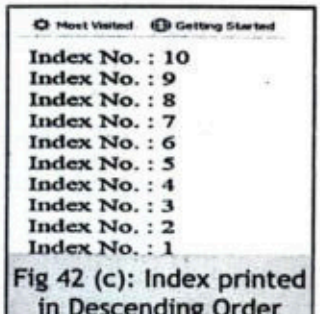


Fig 42 (c): Index printed in Descending Order

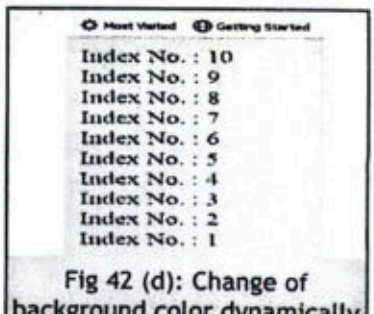


Fig 42 (d): Change of background color dynamically

Summary

- A document which exists and is accessible through internet is a webpage, while a set of webpages is termed as Website.
- Universal Resource Locator (URL) is the accessible address of the document.
- Search engine provides the service to seek relevant information based on the keywords you have entered.
- A computer program which executes tasks via a browser and internet connection, remotely accessing a server, is called web application.
- Static Website once loaded on the user's computer the link to the server is no more required. Whereas dynamic website contains pages which are created on demand basis, at the spot.
- The front-end of a website is developed using Html, CSS and Javascript, etc., while back-end development requires more knowledge and skill level than front-end development, like knowledge of JavaScript, Python, PHP, ASP.Net, etc.
- Hypertext Markup Language (HTML) is the language used to define and display your contents in the form of a webpage.
- Anything between angle brackets '<' and '>' and the character(s) between these angle brackets is said to be a tag.
- Links are called Hyperlinks in html with 'a' tag-pair. Hyperlinks are easy to identify on a webpage, as the mouse cursor changes as soon as the cursor touches a link element.
- Arrangement and presentation of the whole webpage along with the components are handled by a stylesheet language, like CSS.
- There are three ways, via which we can use CSS styles in our HTML webpage, i.e. Inline, Embedded or External CSS.
- JavaScript is primarily used in development of web pages and scripts. It does not consume much of memory and that is why it is used at the client-end in developing websites, for making pages dynamic. It easily works with programming languages like HTML, similar to CSS.
- For input from the user we can use the prompt() function, which pops up a message window like that of alert() function.
- A variable is an entity that stores some value for later use. In programming languages, variable should be named in meaningful manner. A good programming practice is to declare and initialize the variable at the same time.
- In conditional statements, on the basis of some constraint the program flow is chosen.

- The 'for loop' starts with an initial value of index, and iterates till the terminating condition is met; and adjusts every iteration on the basis of step-size.
- Multiple iterative tasks, which are incorporated one loop inside the other and are termed as Nested Loop.
- An array is a datatype which can hold a number of homogenous set of elements.
- Function is a set of code that occurs in the code quite often which can be segmented once and called again and again. A function has a name through which it is identified and called. A function can return a value.

Teacher's Guide

The Odin Project's "Debugging Challenges" offers helpful activities specifically designed for JavaScript debugging.
(https://www.theodinproject.com/paths/full-stack-javascript/courses/javascript)

Exercise

Select the suitable answer for the following Multiple Choice Questions (MCQs).

1) Everything in HTML is identified on the basis of _____.
   a) Brackets b) Title
   c) Tags d) Image
2) The output of HTML code is visible in _____.
   a) Notepad b) File
   c) Browser d) Spreadsheet
3) Name of a web page can be given using _____ tag.
   a) Body b) Title
   c) Head d) Footer
4) Main parts of a document are arranged in _____ tag.
   a) Head b) Title
   c) Body d) Line Break
5) The heading tag-pair for 5th level heading is _____.
   a) <h5 .. h5> b) <h5 .. /h5>
   c) h5> .. </h5 d) <h5> .. </h5>
6) Span is used to provide _____ to a line.
   a) Font b) Border
   c) Style d) Color

7) First row of table in HTML is called _____.
   a) Title row b) Top Row
   c) Header Row d) Upper Row'
8) p' tag-pair is used for _____.
   a) Print b) Page align
   c) Page break d) Paragraph
9) A variable cannot start with a _____.
   a) Alphabet b) Number
   c) Character d) String
10) The first value assigned to a variable after declaration is called _____.
   a) Beginning value b) Starting value
   c) Initialization d) Substitution

Give Short answers to the following Short Response Questions (SRQs).

1) Contrast between website and web application.
2) What is 'href' refers to and how to use it?
3) Enlist the optional parameters to open a webpage.
4) List out the frequent tags used in text of a webpage and what are they used for?
5) Explain the role of <body> tag-pair in a document.
6) How the event based code is used in JavaScript?
7) Infer about the External CSS? Where are External CSS generally used?

Give Long answers to the following Extended Response Questions (ERQs).

1) What is Document Object Model? Explain with the help of an example.
2) Write code to differentiate between different types of headings in HTML.
3) Elaborate steps and provide code to load a background image in a webpage.
4) With the help of sample code, highlight different methods to incorporate CSS code in a HTML webpage.
5) Sketch steps and provide code to apply border and color to a table in a webpage.
6) Discuss the functionality JavaScript can provide in a webpage with the help of a suitable example code.
7) Articulate steps and write code to create a scrolling text on a webpage.
8) Enlist steps to add a video clip in a website which starts playing as the web page loads.
9) Cite steps on compiling the result of your last examination in a tabular form and display it in a webpage.
10) In context of Fig. 40(d), add another button namely 'Revert' which when is pressed, it will reverse both the color and index values.

Project - 1

Design a webpage for Annual Function of your school.

Provide with:

- Schedule of Events
- List of Invited Guest Speakers
- Major awards that will be distributed by Chief Guest.

Add some relevant text paragraphs and photographs for each point, along with an introductory paragraph.

Project - 2

Create a webpage about you and your hobbies (like stamp collecting, pets, etc.)
The webpage initially should highlight about your family background, your personal achievements and activities. Later, provide details about your every hobby for creating awareness and motivation for others.
You may add some related links for others who want to start and opt for the same hobbies.

Project - 3

Home-made food items are getting popularity, but majority are lacking a website to market their products and new innovations. Your target is to come up with a colorful webpage emphasizing the awareness of organic and home-based meals.
Moreover, you need to put in a menu of food items with the prices and number of persons servings.
The major ingredients need to be listed against each item of the menu with a picture.
Design a corresponding logo for the website, too. And use the same logo as a background for the website.
Lastly, provide contact details and delivery options.

Activity

Create a timetable for the subjects mentioned in the fig-16, and display on a new page by providing a 'Time Table' link on the main page.

Activity

Using '\' and '/' draw a big 'X', as shown: