UNIT NO. 03 PROGRAMMING FUNDAMENTALS

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

Q.No.01: Contrast between website and web application.

Ans: Website: A website is a collection of many web pages, and web pages are digital files that are written using HTML(HyperText Markup Language). The website's web pages are linked with hyperlinks and hypertext and share a common interface and design. The website might also contain some additional documents and files such as images, videos, or other digital assets. On the other hand, a computer program which offers a services or executes tasks via a browser and internet connection, remotely accessing a server is called web application.

Q.No.02: What is href refers to and how to use it?

Ans: This term is an abbreviation. It stands for the "Hypertext reference" term. Hyperlinks are essential because of providing a way to navigate between web pages. Hyperlinks are built using the HTML anchor tag, which includes the "href" attribute, and specify the URL of the target resource. The href attribute is significant because it allows linking web pages, files, images, and other resources.

Q.No. 03 What is HTML? How HTML contribute in web design?

Ans: HTML is described as below:

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Role of HTML in Web Design:

HTML acts as the building block for a web page, providing the structure that web browsers interpret to render the page correctly. It works in conjunction with CSS (Cascading Style Sheets) and JavaScript to create visually appealing, interactive, and dynamic web pages.

Q.No.04: Write down sample code for simple webpage.

Ans: Following code is used to create simple HTML page. Type following code in your favorite editor like Visual Studio Code, Atom, Note++, Notepad etc and save the same file with .html extension and also select All files from the drop down box, while saving document.

```
<!DOCTYPE html>
<html>
<head>
<title>First Web Page</title>
</head>
<body>
<h1>This tag is used for heading purpose</h1>
<p>This tag is used for paragraph in HTML page .</p>
</body>
</html>
```

Q.No.05: Explain all tags used in Q. No. 04.

Ans: Tags are explained below:

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

Q.No.03: What is JavaScript?

Ans: **JavaScript** is a lightweight, interpreted **programming** language. It is commonly used to create dynamic and interactive elements in web applications. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Q.No. Explain JavaScript Syntax.

Ans: JavaScript syntax comprises a set of rules that define how to construct a JavaScript code. JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page.

You can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the **<head>** tags.

The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

```
<script language = "javascript" type = "text/javascript">
  JavaScript code
</script>
```

Q.No.03: How to write simple JavaScript code in HTML?
Ans: Let us take a sample example to print out "Hello World". We call **document.write** method which writes a string into our HTML document. This method can be used to write text, HTML, or both. Take a look at the following code –

```
<!DOCTYPE html>
<html>
<head>
  <title> Your first JavaScript program </title>
<head>
<body>
  <script language = "javascript" type = "text/javascript">
    document.write("Hello World!")
  </script>
</body>
</html>
```

Q.No.03: Enlist the optional parameters to open a web page.

Q.No.04: List out the frequent tags used in text of a webpage and what are they used for?

Ans: Formatting elements were designed to display special types of text:

- **<b>** - Bold text
- **** - Important text
- **<i>** - Italic text
- **<em**> - Emphasized text
- **<mark>** - Marked text
- **<small>** - Smaller text
- **<del>** - Deleted text
- **<ins>** - Inserted text
- **<sub>** - Subscript text
- **<sup>** - Superscript text

Q.No.05: Explain the role of <body> tag-pair in a document.

Ans: The **<body>** tag defines the document's body. The **<body>** element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

Note: There can only be one **<body>** element in an HTML document.

Q.No.06: How the event based code is used in JavaScript?

Ans: When Button is clicked the date will be displayed on Web Page

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript HTML Events</h1>
<h2>The onclick Attribute</h2>
<p>Click the button to display the date.</p>
<button onclick="displayDate()">The time is?</button>
<script>
function displayDate() {
  document.getElementById("demo").innerHTML = Date();
}
</script>
<p id="demo"></p>
</body>
</html>
```

Q.No.07: Infer about the External CSS? Where are External CSS generally used?

Ans: The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file. Using external CSS ensures the look of all your pages is consistent, at least if you use 1 CSS file for the whole site. There may be a speed penalty for the first page, but from then on the CSS file is cached, and as a result subsequent pages will actually load faster.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

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

Q.No.01: What is Document Object Model? Explain with the help of example.

Ans: The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

Finding HTML Elements

When you want to access HTML elements with JavaScript, you have to find the elements first.

There are a couple of ways to do this:

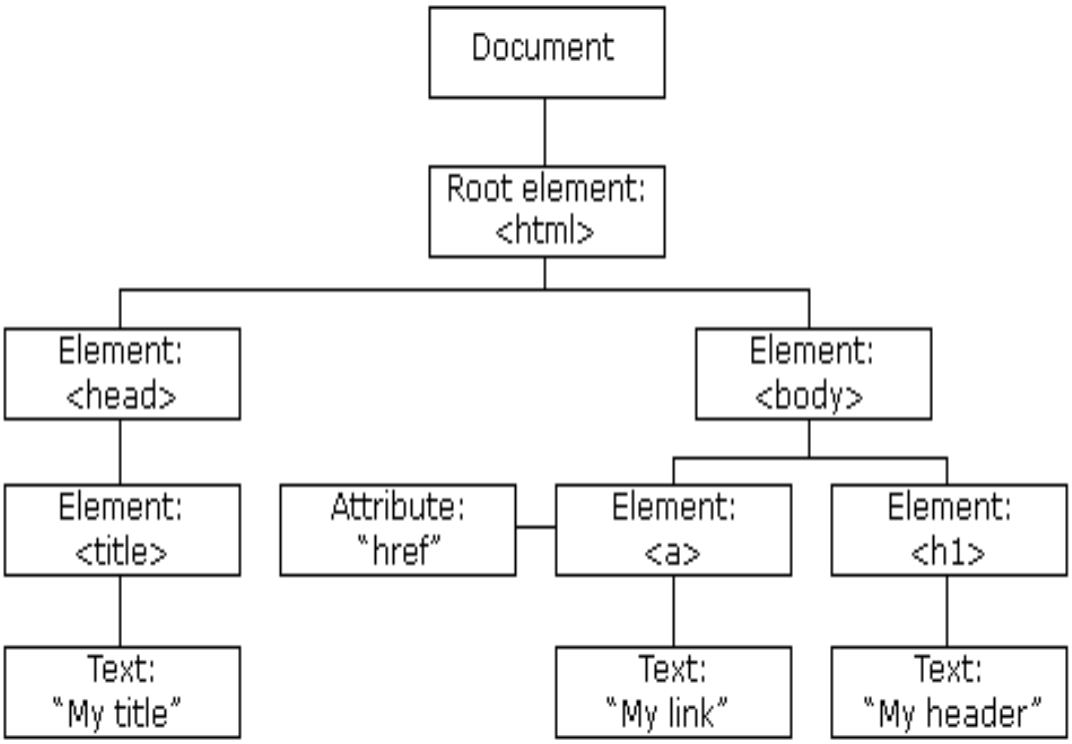
- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by CSS selectors
- Finding HTML elements by HTML object collections

JavaScript Document Object Methods

The JavaScript Document Object provides us with various methods that allow us to interact with and manipulate the HTML document. Some common Document object methods are as follows –

- **getElementById(id)** − Returns the element with the specified ID.
- **getElementsByClassName(className)** − Returns a collection of elements with the specified class name.

- **getElementsByTagName(tagName)** − Returns a collection of elements with the specified tag name.
- **createElement(tagName)** − Creates a new HTML element with the specified tag name.
- **createTextNode(text)** − Creates a new text node with the specified text.
- **appendChild(node)** − Appends a node as the last child of a node.
- **removeChild(node)** − Removes a child node from the DOM.
- **setAttribute(name, value)** − Sets the value of an attribute on the specified element.
- **getAttribute(name)** − Returns the value of the specified attribute on the element.



Q. How Does the JavaScript Code Work on Your Web Page?

Ans: JavaScript is either embedded directly into a web page or referenced via a separate .js file. When a user visits that web page, their browser will run the script along with the HTML and CSS code – creating a functional page displayed via the browser tab.

The script is downloaded to the visitors' machines and processed there. This differs from a server-side language, in which the server processes the script before sending it to the browser.

When encountering a block of JavaScript code, a web browser will process it from top to bottom. Since it's order-sensitive, make sure to reference the

objects or variables within the block first before modifying them. Having variables with no values will result in an undefined error.

Q.No. What Makes JavaScript Great?

Ans: JavaScript has a number of advantages that make it a better choice than its competitors. The following are several benefits of using JavaScript:

- **Simplicity** – having a simple structure makes JavaScript easier to learn and implement, and it also runs faster than some other languages. Errors are also easy to spot and correct.
- **Speed** – JavaScript executes scripts directly within the web browser without connecting to a server first or needing a compiler. Additionally, most major browsers allow JavaScript to compile code during program execution.
- **Versatility** – JavaScript is compatible with other languages like PHP, Perl, and Java. It also makes data science and machine learning accessible to developers.
- **Popularity** – plenty of resources and forums are available to help beginners with limited technical skills and knowledge of JavaScript.
- **Server load** – another perk of operating on the client-side is that JavaScript reduces the requests sent to the server. Data validation can be done via the web browser, and updates only apply to certain web page sections.
- **Updates** – JavaScript development team continuously update and create new frameworks and libraries, ensuring its relevance within the industry.

Q.No.02: Write code to differentiate between different types of headings in HTML.

Ans: Following code will explain different type of heading used in HTML:

<!DOCTYPE html> <html> <body> Heading 1 Heading 2 Heading 3 Heading 4 Heading 5 Heading 6 </body> </html>	**Output** **Heading 1** **Heading 2** **Heading 3** **Heading 4** **Heading 5** **Heading 6**

Q.No.03: Elaborate steps and provide code to load a background image in a webpage.

Ans: There are two approaches to setting an image as the webpage's background image, which we will learn in this article. They are –

- Using background attribute
- Using CSS

Method 1: Using background attribute

We can use the **background attribute** in the **body tag** to set an image as the background of the webpage. We will need to specify the URL or the location of the image which we want to set to the background attribute of the body tag.

Syntax

<body background = "URL or Path of Image">Body of the Webpage</body>

<!DOCTYPE html>

```
<html lang="en">
<head>
  <title>Background Image</title>
</head>
<body background=”Dar.png">
  <h1 style="color: green;
        text-align: center;">
    Welcome to IMCB, G-11/1, Islamabad
  </h1>
</body>
</html>
```

Method 2: Using CSS

We can also use CSS to set any image as the background of the webpage. To do so, we will need to specify the desired image's location or URL to the **background-image property**.

Syntax

```
body {
  background-image: url(" URL of the Image");
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
 background-image: url("Dar.jpg");
}
</style>
</head>
<body>

<h1> Welcome to IMCB, G-11/1, Islamabad </h1>
<p>This text is not easy to read on this background image.</p>

</body>
</html>
```

Q.No.04: With the help of sample code, highlight different methods to incorporate CSS code in HTML webpage.

Ans: CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files.

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.
An inline CSS uses the style attribute of an HTML element.
The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;">A Blue Heading</h1>
<p style="color:red;">A red paragraph.</p>
</body>
</html>
```

Internal CSS

An internal CSS is used to define a style for a single HTML page.
An internal CSS is defined in the <head> section of an HTML page, within a <style> element.
The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
```

```
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

External CSS

An external style sheet is used to define the style for many HTML pages.
To use an external style sheet, add a link to it in the <head> section of each HTML page:
The external style sheet can be written in any text editor(Notepad). The file must not contain any HTML code, and must be saved with a .css extension.
With an external style sheet, you can change the look of an entire website, by changing one file.
Here is what the **"styles.css"** file looks like:

"styles.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

HTML Page

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body> <h1>This is a heading</h1>  <p>This is a paragraph.</p> </body>
</html>
```

Q.No.05: Sketch steps and provide code to apply border and color a table in a webpage.

Ans: Defining border for table and background of a table by using CSS:

```
<!DOCTYPE html>
<html>
<head>
<style>
table,th, td {
  border-style:solid;
  border-color: blue;
  background-color:aqua;
  }
</style>
</head>
<body>
<h2>Table With Border Color</h2>
<p>Use the CSS border-color property to set the color of the borders.</p>
<table style="width:100%">
  <tr>
    <th>Firstname</th>   <th>Lastname</th>     <th>Age</th>   </tr>
  <tr>
    <td>Arsalan</td>    <td>Sikandar Dar</td>    <td>15</td>   </tr>
  <tr>    <td>Aalian</td>    <td>Sikandar Dar</td>    <td>10</td>   </tr>
  <tr>    <td>Muhammad</td>    <td>Sikandar Dar</td>    <td>40</td>   </tr>
</table>
</body>
</html>
```

Q.No.06: Discuss the functionality JavaScript can provide in a Webpage with the help of a suitable example code.

Ans: JavaScript is a versatile programming language that can add a wide range of functionalities to a web page. Some of the common functionalities that JavaScript can provide on a web page include:

1. **Dynamic Content**: JavaScript can be used to dynamically update and change content on a web page without requiring a full page reload. This can include updating text, images, or other elements based on user interactions or data from external sources.
2. **Interactivity**: JavaScript enables interactive elements on a web page such as buttons, forms, sliders, accordions, tabs, and more. These interactive elements can respond to user input, creating a more engaging user experience.
3. **Client-Side Validation**: JavaScript can be used to validate user input on forms before the data is submitted to the server. This helps improve user experience by providing immediate feedback to users on whether their input is valid.
4. **Animations**: JavaScript can be used to create animations on a web page, such as fading effects, sliding elements, or complex animations. This can help make a website more visually appealing and engaging.
5. **Cookies**: JavaScript can interact with browser cookies to store and retrieve information on the client-side. This can be useful for maintaining user sessions, storing user preferences, and tracking user behavior.
6. **DOM Manipulation**: JavaScript can manipulate the Document Object Model (DOM) of a web page to dynamically create, modify, or delete elements. This allows for dynamic updates to the content and structure of a web page based on user interactions or other events.
7. **Form Handling**: JavaScript can enhance form handling by dynamically updating form elements, validating input, and submitting form data asynchronously. This can improve the user experience when interacting with forms on a web page.
8. **Error Handling**: JavaScript allows developers to handle errors gracefully and provide meaningful error messages to users when issues occur. This helps improve the usability of a web application.

These are just a few examples of the functionalities that JavaScript can add to a web page. JavaScript is a powerful language that plays a crucial role in modern web development by enabling dynamic and interactive web experiences.

Example with Code 1

```
<!DOCTYPE html>
<html>
<body>
<h2>What Can JavaScript Do?</h2>
<p id="demo">JavaScript can change HTML content.</p>
<button type="button"
onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!"'>Click Me!</button>
</body>
</html>
```

Example 2

```
<!DOCTYPE html>
<html>
<body>
<h2>Dynamic Content example
</h2> <button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>
<p id="demo"></p>
</body>
</html>
```

Q.No.07: Articulate steps and write code to create a scrolling text on a webpage.

Ans: Scrolling text in web page

```
<!DOCTYPE html>

<html>
<style>
.scroll-left {
 height: 50px;
 overflow: hidden;
 position: relative;
```

```
 background: white;
 color: orange;
 border: 1px solid orange;
}
.scroll-left p {
 position: absolute;
 width: 100%;
 height: 100%;
 margin: 0;
 line-height: 50px;
 text-align: center;
 /* Starting position */
 transform:translateX(100%);
 /* Apply animation to this element */
 animation: scroll-left 10s linear infinite;
}
/* Move it (define the animation) */
@keyframes scroll-left {
 0%   {
 transform: translateX(100%);
 }
 100% {
 transform: translateX(-100%);
 }
}
</style>
<div class="scroll-left">
<p>This text will scroll from left side... </p>
</div>
</html>
```

Q.No.08: Enlist steps to add a video clip in a website which starts playing as the webpage loads.

Ans: Following code will add a video clip in website:

```
<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" autoplay>
 <source src="movie.mp4" type="video/mp4">
 <source src="movie.ogg" type="video/ogg">
 Your browser does not support the video tag.
</video>
</body>
</html>
```

Q.No.09: Cite steps on compiling the result of your last examination in a tabular form and display it in a webpage.

Ans: Result of a student will be displayed in tabular form:

```
<!DOCTYPE html>
<html>
<body>
<table>
 <tr>
  <th>Name</th>  <th>Subject</th>  <th>Marks</th> </tr>
 <tr>
  <td rowspan = "2">Hillary</td>  <td>Advanced Web</td>  <td>75</td>
 </tr>
 <tr>  <td>Operating Syatem</td>  <td>60</td> </tr>
   <tr>
  <td rowspan = "2">Lary</td>  <td>Advanced Web</td>  <td>80</td>
</tr>
 <tr>  <td>Operating Syatem</td>  <td>75</td> </tr> <tr>  <td
colspan="3">Total Average: 72.5</td> </tr> </table> </body>
</html>
```

Q.No.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.

Ans:

```
<!DOCTYPE html>
<html>
<body>
<script>
var index;
document.write("For loop starts here <br>");
for(index=0;index<10;index++)
{
document.write("Index No.",index,"<br>");
}
document.write("For loop stops here...");
function desorder()
{
for(index=10;index>0;index--)
{
document.write("Index No.",index,"<br>");
document.body.style.backgroundColor="peachpuff";
}
}
function revert()
{
for(index=0;index<10;index++)
{
document.write("Index No.",index,"<br>");
}
}
</script>
<p id="content"> Click button to change output in Descending Order</p>
<button onclick="desorder()"> Descending Order</button>
<p id="color"> Click button to change background color</p>
<button onclick="revert()"> Descending Order Color removed</button>
</body>
</html>
```

Short Questions

Q.No. 01 Define responsive design and its significance in website development.

Ans: Responsive Web Design is **designing websites that contain flexible layouts** that can scale itself according to the screen size of the device it is being viewed on. A Responsive Web Design enlarges, shrinks, resizes, or hides content present as per the screen size of the device. Using websites that look bad on the screen can be displeasing for the users and they might just close and stop using that specific website.

If we are using websites that look the same on every screen size, it might be bad for the viewers as it can decrease their interest in the website and the brand. A brand that uses a **non-responsive website loses customers** as not every customer has the same system to open a website. Making your website **responsive helps the brand to increase its customers** from different platforms and helps them to build trust from a variety of users.
Using a Responsive Web Design for a website can help the brand to set its unique and positive identity in the market as customers can visit that specific website through different devices and the experience will still be good.

Q.No. 02 Explain the function of web hosting in making a website accessible.

Ans: Web Hosting is like renting space on the Internet or the **web browser**, its equivalent to allocating server space on the **World Wide Web**. Which secures your dedicated environment for your web domain. **Web hosting** provides a space to keep your website's **data on a server**. When someone enters your domain name into their browser, this server promptly displays your site to them.

Q.No. 03 What is version control, and why is it useful in web development?

Ans: Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done in the code.

Q.No.04 Why Version Control system is so Important?

As we know that a software product is developed in collaboration by a group of developers they might be located at different locations and each one of them contributes to some specific kind of functionality/features. So in order to contribute to the product, they made modifications to the source code(either by adding or removing). A version control system is a kind of software that

helps the developer team to efficiently communicate and manage(track) all the changes that have been made to the source code along with the information like who made and what changes have been made

Q.No. 05 How do web browsers facilitate the display of websites?

Ans: The web browser is an application software to explore www (World Wide Web). It provides an interface between the server and the client and it requests to the server for web documents and services. It works as a compiler to render HTML which is used to design a webpage. Whenever we search for anything on the internet, the browser loads a web page written in HTML, including text, links, images, and other items such as style sheets and JavaScript functions. Google Chrome, Microsoft Edge, Mozilla Firefox, and Safari are examples of web browsers.

Q.No. 06 Define security measures like HTTPS and SSL in web applications.

Ans: HTTP (Hypertext Transfer Protocol) is the fundamental protocol of the World Wide Web, governing data transmission, formatting, and server responses. While widely adopted, HTTP lacks security measures such as data encryption and authentication, leaving transmitted data vulnerable. To address this, HTTPS (Hypertext Transfer Protocol Secure) establishes an encrypted connection between clients and servers, safeguarding websites from eavesdropping, tampering, and data theft.

It's important to understand that SSL/TLS is a part of HTTPS; together, they are a single protocol. The difference between HTTPS and HTTP is that HTTPS uses SSL/TLS to provide more security than HTTP alone. This leads to the question: Is SSL the same as TLS?

While the terms SSL and TLS are used interchangeably, TLS is considered more trustworthy due to its improvements over SSL. TLS addresses known vulnerabilities of SSL, supports stronger cipher suites and algorithms, and offers a faster handshake process. It includes additional security features like the "close notify" message, HMAC (Hash-based Message Authentication Code), and upgraded cipher suites to mitigate previous SSL security concerns.

Long Question

1. **How does JavaScript enhance user engagement on web pages, and what functionalities does it enable?**
2. **Explain the significance of responsive design in ensuring optimal website performance across different devices and screen sizes.**
3. **Describe the role of web hosting in making a website accessible to users and ensuring its availability on the internet.**

Q.No. 04 How does version control, such as Git, aid in managing code changes and facilitating collaboration among web developers?

Ans: Version control is the management of changes to documents, files, or any other type of data. In software development, it is essential for managing and tracking changes to the codebase, ensuring code quality, reducing errors, and improving collaboration among team members.

What is Git?

Git is a popular version control system used by developers to manage changes to code. It allows developers to track changes made to their codebase, collaborate with team members, and revert to previous versions if needed.

Git is widely used in software development due to its flexibility, speed, and ability to handle large codebases with ease. It also offers a range of features and tools for managing and organizing code, such as branching and merging. And it has a large and active community of users who contribute to its development and provide support.

A key benefit of using Git is its ability to facilitate collaboration between developers. Git allows you to work on the same codebase with others simultaneously, without overwriting other developers' changes.

To collaborate on a Git project, you typically use a central repository that serves as the source of truth for the project. Each developer has a local copy of the repository on their machine, and they make changes and commit them to their local repository.

When you're ready to share your changes with the rest of the team, you push your changes to the central repository. Other team members can then pull those changes down to their local repositories.

4. **Elaborate on the importance of security measures like HTTPS and SSL in maintaining data integrity and privacy during online interactions.**

CO2

Short Questions

1. **Explain the main function of CSS in web development and provide an example of a style property it can control.**
2. **Describe the difference between static and dynamic websites, including their characteristics and use cases.**

3. **What is the purpose of the HTML <a> element with the href attribute, and how does it contribute to web navigation?**
4. **Explain the concept of "hyperlink capability" in HTML and provide an example of creating a hyperlink in an HTML document.**
5. **Differentiate between CSS and JavaScript in web development. How do they contribute to the overall functionality of a web page?**
6. **Describe the characteristics of both static and dynamic websites, and provide examples of situations where each type of website is most suitable.**

Long Questions

1. **Discuss the essential functions of HTML in web development, including content structuring, text formatting, image integration, and hyperlink capability.**
2. **How is JavaScript used in web development to modify a website? Explain its role in DOM manipulation, event handling, AJAX requests, and form validation.**
3. **Compare and contrast HTML, CSS and JavaScript in terms of their roles and functions in web development. Provide examples of how they work together to create dynamic web page.**
4. **Discuss the role of HTML in structuring and formatting web content. Provide examples of HTML tags and their functions.**
5. **Explain how JavaScript can be used to enhance user interactivity on a website. Describe its applications in DOM manipulation, event handling, AJAX requests, and form validation.**
6. **Compare and contrast the primary technologies used in web development: HTML, CSS and JavaScript. How do they work together to create a modern and interactive website?**

Short Questions

Q.No. 01 What are the differences between primitive and structural data types in JavaScript?

Ans: Primitive data types are the built-in data types provided by all programming languages. Examples of primitive data types in JavaScript are **string, number, boolean, null, symbol,** and **undefined**. These data types can store only a single value of a particular type in the memory. The primitive data types are explained below:

- **String:** It is a data type that stores the text values assigned using the **double(" ")** or **single(' ')** quotes.
- **Number:** The number data type represents a numerical value either a floating point or an integer.
- **Boolean:** It only contains the two values either **true** or **false.**
- **null:** It refers to a variable that stores only **null** as its value.
- **Undefined:** This data type means a variable that is declared but not yet initialized with a value.
- **Symbol:** It is introduced in ES6. It creates a private object property by referring to that key only.

Structural Data Types

The structural data types contains some kind of structure with primitive data.

Data Type	Description

Object	An object holds multiple values in terms of properties and methods.
	Example:
	Let person={
	firstName:"Muhammad",
	lastName:"Sikandar Dar",
	age:40
	};

Array	An array stores multiple values using special syntax. Example: Let nums=[1,2,3,4]; Let stringarray=["One","Two"]; Let deciarray=[1.2, 3.5]; Let boolarray=[true, false];

Date	The Date object represents date and time including days, months, years, hours, minutes, seconds and miliseconds. Example: Let today=new Date("10 June 2024");

Difference between primitive and Structural data types in JavaScript:

Primitive	Non-Primitive
Primitive Data types are predefined.	Non-Primitive data types are created by the programmer
Primitive Data types will have certain values.	Non-Primitive data types can be NULL.
Size depends on the type of data structure.	Size is not fixed
Examples are numbers and strings.	Examples are Array and Linked List.
It can start with a lowercase.	It can start with uppercase.

Long Question

Q.No. 01 Compare and contrast the three variable declaration keywords in JavaScript: var, let, and const. Explain the advantages and use cases for each.
Ans: In JavaScript a variable can be declared using **var**, **let** and **const** keywords.

Var keyword is used to declare variables since JavaScript was created. It is confusing and error-prone when using variable declared using **var**.

var a = 10

```
function f() {
   var b = 20
   console.log(a, b)
}
f();
console.log(a);
```

let keyword removes the confusion and error of var. It is the new and recommended way of declaring variables in JavaScript.

```
let a = 10;
function f() {
   let b = 9
   console.log(b);
   console.log(a);
}
f();
```

const keyboard is used to declare a constant variable that can not be changed once assigned a value.

```
const a = 10;
function f() {
   a = 9
   console.log(a)

}

f();
```

TypeError: Assignment to constant variable.

1. **Discuss the role of JavaScript in modifying the content of styling of HTML elements. Provide examples of how JavaScript can be used to dynamically change content and apply styles to HTML elements on a web page.**

2. **Explain the various ways JavaScript can be used to add interactivity to a web page, including form validation, dynamic content loading, and real-time feedback to users. Provide examples of each.**
3. **Compare and contrast primitive and structural data types in JavaScript. Provide examples of each type and discuss their use cases.**
4. **Illustrate the steps and code examples for dynamically changing the content of an HTML element using JavaScript. Explain the significance of this in web development.**
5. **Discuss the various ways JavaScript can be used to enhance interactivity and user experience on a web page. Provide real-world examples of dynamic content loading, event handling, and form validation.**
6. **Explain the relationship between JavaScript and CSS in web development, emphasizing how JavaScript can modify CSS properties to create dynamic visual effects. Provide example and use cases.**

CO4

Short Questions

1. **What is an algorithm, and why is the sequence of instruction crucial in its execution?**
2. **How the sequence of instructions affects the flow of control within an algorithm.**
3. **How does the sequence of instructions impact data dependencies in an algorithm?**
4. **What is the significance of correct sequencing in an algorithm's execution and its output?**
5. **How does the order of instructions influence the efficiency of an algorithm's performance?**
6. **In JavaScript, what defines a sequence of instruction executed one after another?**
7. **Please provide an example of a sequence of instructions in JavaScript and explain its execution order.**
8. **How is selection implemented in JavaScript, and what role does the "if" statement play?**
9. **What is the concept of iteration/repetition in JavaScript using a specific loop example?**

Q.No 10 What is the practical use of loops in JavaScript, and how do they benefit coding task?

Ans: The 'for loop' is a fundamental construct in JavaScript that allows you to iterate over a block of code a specified number of times. It is particularly useful when you need to perform a repetitive task for a known number of times. Some practical uses of the 'for loop' in JavaScript include:

1. Iterating over an array or object to perform a specific task on each element or property.
2. Generating a sequence of numbers or characters.
3. Creating a countdown or timer.
4. Creating a loop with a conditional statement to filter or manipulate data.

Here is an example of a 'for loop' that iterates over an array and logs each element to the console:```javascript
const array = ['apple', 'banana', 'orange'];

for (let i = 0; i < array.length; i++)

{
console.log(array[i]);
}
```This code will output:

1. apple
2. banana
3. orange

**Q.No. 10 Define arrays in JavaScript and explain their significance in storing multiple values.**

**Ans:** An array in JavaScript is a data structure used to store multiple values in a single variable. It can hold various data types and allows for dynamic resizing. Elements are accessed by their index, starting from 0.

**Basic Terminologies of JavaScript Array**

- **Array:** A data structure in JavaScript that allows you to store multiple values in a single variable.
- **Array Element:** Each value within an array is called an element. Elements are accessed by their index.
- **Array Index:** A numeric representation that indicates the position of an element in the array. JavaScript arrays are zero-indexed, meaning the first element is at index 0.
```

- **Array Length:** The number of elements in an array. It can be retrieved using the length property.

Uses of Arrays in Java

There are various uses of java array, such as:

1. They allow us to store multiple elements of the same type in a single array. This is useful when we need to store large amounts of data and process them efficiently.
2. Arrays are also used for sorting, searching, accessing, and manipulating data.
3. Also, arrays provide various advantages such as memory efficiency, better performance, and faster execution time.
4. Arrays can solve complex problems such as calculating the sum or finding maximum or minimum values in an array.
5. They can also be used to implement multi-dimensional arrays, which are useful for representing matrices and graphs.
6. Arrays have applications in computer programming, engineering, mathematics etc.

10. What is the advantage of using arrays over individual variables to store data in JavaScript

Ans: Here are the **advantages of array**:

1. **Easy to use:** Arrays are easy to use and require less coding than traditional data structures. This makes arrays a great choice for rapid development.
2. **High performance:** Arrays provide fast and efficient access to elements as compared to other data structures such as linked lists, trees etc.
3. **Flexible size:** The size of arrays can be changed easily at runtime, making them a very flexible type of data structure.
4. **Memory efficient:** Arrays are memory efficient as they can store multiple values in the same location. This reduces the amount of RAM required to store data and improves overall performance.
5. **Random Access:** Arrays support random access, meaning elements can be accessed directly using their index. This makes arrays an ideal choice for applications requiring fast data access.

11. **Illustrate different ways to create array in JavaScript using examples?**
12. **How does the const keyword relate to array declaration in JavaScript?**
13. **What is the significance of the "new" keyword in creating arrays in JavaScript?**

Long Question

1. **Discuss the significance of proper sequencing of instructions in algorithms. Provide examples to illustrate how changing the order of instructions can impact the outcome.**
2. **Explain the use of JavaScript arrays and how they differ from single variables. Provide examples of creating and accessing elements in arrays.**
3. **Describe the advantages of using loops in programming, specifically in JavaScript. Provide real-world scenarios where loops are beneficial and explain how they solve problems.**
4. **Illustrate the concept of a "for loop" in JavaScript and compare it to a "while loop". Provide examples of when and how each loop is used.**
5. **Discuss the importance of efficient algorithms in computer science and provide examples of how the sequence of instructions can impact the overall efficiency of the algorithm.**
6. **Explore the concept of arrays in JavaScript. Explain how arrays are created, accessed, and manipulated. Provide real-world scenarios where arrays are useful.**
7. **Describe the role of loops in programming, with a focus on JavaScript. Provide detailed examples of how for and while loops can be used to solve practical problems.**
8. **Compare and contrast conditional statements (e.g., if statements) and loops in JavaScript. Explain when to use each construct and provide examples to illustrate their applications.**

C05

Short Questions

1. **What is the primary purpose of using functions in JavaScript?**
2. **Explain why testing code inside a function is often considered easier than testing global code.**
3. **Define a breakpoint in JavaScript debugging and its significance.**

4. **How does the Scope pane in DevTools aid in debugging JavaScript code?**
5. **Describe the role of parseInt() function in JavaScript?**
6. **How do functions typically communicate their output in JavaScript?**
7. **Enumerate two advantages of using functions in JavaScript.**
8. **Identify the panel in DevTools primarily used for inspecting JavaScript files and setting breakpoints.**
9. **What is the significance of the keyword functions in JavaScript?**

Long Questions

1. **Describe a situation where you encounter a JavaScript function that is expected to manipulate a give array but is producing incorrect output. Outline a step-by-step process using DevTools to debug this function, including setting breakpoints, analyzing variable values, and proposing a fix.**
2. **Explain the challenges associated with testing code written in JavaScript. Discuss how the lack of encapsulation and the dependence on external state impact the testing process, and propose strategies to overcome these challenges.**
3. **Discuss two key advantages of using functions in JavaScript. Elaborate on how functions typically communicate their outputs, emphasizing the importance of the return statement in conveying results.**

Q.No. 04:Detail the primary functions and features of Dev Tools, specifically the Sources panel, and explain how it aids in debugging JavaScript code. Include the significance of breakpoints , variable inspections and scope pane in this process.

Ans: Chrome DevTools is a set of web developer tools built directly into the Google Chrome browser. DevTools can help you edit pages on-the-fly and diagnose problems quickly, which ultimately helps you build better websites, faster.

There are some reasons why you need Chrome DevTools for testing:

- DevTools gives you powerful tools for inspecting and editing your code. With DevTools, you can quickly find and fix any errors in your code.
- DevTools lets you measure the performance of your pages. You can use the Performance panel to find and fix bottlenecks in your code and see how your page performs on different devices and browsers.
- DevTools lets you simulate different types of devices and browsers. With Device Mode, you can test your page on different devices and see how it looks and performs. The Browser Mode also lets you simulate different versions of browsers, making sure your page looks good and functions correctly in all browsers.

The Sources panel and how to use it?

The Sources panel lets you view and edit your page's JavaScript and CSS code. With the Sources panel, you can set breakpoints, step through code, and see the values of variables.
Select a file in the "Files" pane to get started with the Sources panel. This will bring up the contents of that file in the "Editor" pane. From here, you make changes to the code and see those changes immediately reflected on the page. You can easily set breakpoints in your code with the Sources panel too. To set a breakpoint, click on the line number where you want to set the breakpoint. This will be showing a dialog where you can enter the breakpoint conditions. Once you've entered the breakpoint conditions, click "OK" to set the breakpoint.
Breakpoints: A breakpoint is a stopping point in the code that, when encountered, halts the program's execution. It allows programmers to examine or manipulate variables and execute lines of code one at a time, primarily used for debugging purposes.
Inspect the Scope
When you're paused on a line of code, the **Scope** tab shows you what local and global variables are defined at this point in execution, along with the value of each variable. It also shows closure variables, when applicable. When you're not paused on a line of code, the **Scope** tab is empty.

Q: What are the best keyboard shortcuts for Chrome DevTools?

Ans: There are many keyboard shortcuts for Chrome DevTools. Some of the most commonly used shortcuts are:

Ctrl+Shift+I: Open DevTools

Ctrl+F: Open the Find bar

Ctrl+Shift+P: Open the Command palette

Q: How can I reload my page without refreshing the entire tab?

Ans: To reload your page without refreshing the entire tab, press Ctrl+R (Cmd+R on Mac).

Q: How can I save a screenshot of my page?

Ans: To save a screenshot of your page, press Ctrl+S (Cmd+S on Mac). The screenshot will be saved as a .png file in your downloads folder.

Q: How can I measure the performance of my page?

Ans: To measure the performance of your page, open the Performance panel. The panel will show you how long it took to load your page, how many resources were loaded, and where any bottlenecks are.

Q: How can I find and fix errors on my page?

Ans: Open the Console panel to find and fix errors on your page. The Console will show you all the JavaScript errors on your page and any warnings or messages from the DevTools themselves.

Q: How can I simulate a different browser?

Ans: To simulate a different browser, open the Settings panel and click the 'Change user agent' button. Select your desired one from the list of browsers.

Q: How can I test my page on Headless Chrome?

Ans: To test your page on Headless Chrome, open the Command Palette and type 'Headless.' Select the 'Run in headless mode' option from the list of options.

Q: How can I close DevTools?

Ans: To close DevTools, press Ctrl+W (Cmd+W on Mac). Otherwise You can click the Close button in the top-right corner of the DevTools window.

Q.No. 05 Define the purpose of the paresInt() function in JavaScript and illustrate its usage with an example. Describe a scenario where incorrect output results from misunderstanding of data types and explain how you would correct this issue within a function using appropriate data type conversions.

Ans: The **parseInt method** parses a value as a string and returns the first integer. It accepts a string argument and optional radix parameter, defining the numeral system base. This method is commonly used for converting string representations of numbers into integers. The method takes two parameters: the string to be pasrsed and the radix(optional, default is 10).

2= binary, 8= octal, 10= decimal, 16= hexadecimal.

Syntax:

parseInt(value,radix);

vaule: This parameter contains a string that is converted to an integer.

radix: This parameter represents the radix or base to be used and it is optional.

It returns a number and if the first character can't be converted to a number then the function returns NaN. It returns a number parsed up to that point where it encounters a character that is not a number in the specified radix(base).

Example 1

Here, we are using the parseInt() method to parse the given float value.

```
let v1 = parseInt("3.14");
console.log('Using parseInt("3.14") = '+ v1);
```

Output

```
Using parseInt("3.14") = 3
```

Example 2: Parsing value with given radix

Here, we will also mention radix with the number.

```
// Base 10
a = parseInt("100", 10);
console.log('parseInt("100",10) = ' +
    a);

// Base 8
```

```
b = parseInt("8", 8);
console.log('parseInt("8",8) = ' +
    b);

// Base 8
c = parseInt("15", 8);
console.log('parseInt("15",8) = ' +
    c);

// Base 16
d = parseInt("16", 16);
console.log('parseInt("16",16) = ' +
    d);

// Leading and trailing spaces are ignored
// in parseInt() function
e = parseInt(" 100 ");
console.log('parseInt(" 100 ") = ' +
    e);

// Base 16(hexadecimal)
f = parseInt("0x16");
console.log('parseInt("0x16") = ' +
f);
```

Output

```
parseInt("100",10) = 100
parseInt("8",8) = NaN
parseInt("15",8) = 13
parseInt("16",16) = 22
parseInt(" 100 ") = 100
parseInt("0x16") = 22
```