**Assignment 1**

**1. Define HTML? Explain the layout of HTML?**

**HTML** is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications.

Hyper Text: Hyper Text simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hyper Text is a way to link two or more web pages (HTML documents) with each other.

Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

Web Page: A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. With the help of HTML only, we can create static web pages.

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

**An example of HTML.**

<!DOCTYPE>

<html>

<head>

<title>Web page title</title>

</head>

<body>

<h1>Write Your First Heading</h1>

<p>Write Your First Paragraph.</p>

</body>

</html>

**HTML layouts** provide a way to arrange web pages in well-mannered, well-structured, and in responsive form or we can say that HTML layout specifies a way in which the web pages can be arranged. Web-page layout works with arrangement of visual elements of an HTML document.

Web-page layout is the most important part to keep in mind while creating a website so that our website can appear professional with the great look. You can also use CSS and JAVASCRIPT based frameworks for creating layouts for responsive and dynamic website designing.

Every website has a specific layout to display content in a specific manner.

Following are different HTML5 elements which are used to define the different parts of a webpage.

* <header>: It is used to define a header for a document or a section.
* <nav>: It is used to define a container for navigation links
* <section>: It is used to define a section in a document
* <article>: It is used to define an independent self-contained article
* <aside>: It is used to define content aside from the content (like a sidebar)
* <footer>: It is used to define a footer for a document or a section
* <details>: It is used to define additional details
* <summary>: It is used to define a heading for the <details> element

## **Features of HTML**

1) It is a very easy and simple language. It can be easily understood and modified.

2) It is very easy to make an effective presentation with HTML because it has a lot of formatting tags.

3) It is a markup language, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a link on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.

5) It is platform-independent because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.

6) It facilitates the programmer to add Graphics, Videos, and Sound to the web pages which makes it more attractive and interactive.

7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

**2. Define the terms:**

**· HTML element & HTML tags**

An HTML file is made of elements. These elements are responsible for creating web pages and define content in that webpage. An element in HTML usually consist of a start tag <tag name>, close tag </tag name> and content inserted between them. Technically, an element is a collection of start tag, attributes, end tag, content between them. Some elements does not have end tag and content, these elements are termed as empty elements or self-closing element or void elements.

## **Example**

<!DOCTYPE html>

<html>

<head>

<title>Web Page</title>

</head>

<body>

<h1>This is my first webpage</h1>

<h2>How it looks?</h2>

<p>It looks Nice!!!!!</p>

</body>

</html>

**· HTML attributes**

* HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.
* Each element or tag can have attributes, which defines the behaviour of that element.
* Attributes should always be applied with start tag.
* The Attribute should always be applied with its name and value pair.
* The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
* You can add multiple attributes in one HTML element, but need to give space between two attributes.

Syntax: <element attribute\_name="value">content</element>

EXAMPLE

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<h1>This is Style attribute</h1>

<p style="height: 50px; color: blue">It will add style property in element</p>

<p style="color: red">It will change the color of content</p>

</body>

</html>

**· HTML iframe**

HTML Iframe is used to display a nested webpage (a webpage within a webpage). The HTML <iframe> tag defines an inline frame, hence it is also called as an Inline frame.

An HTML iframe embeds another document within the current HTML document in the rectangular region.

The webpage content and iframe contents can interact with each other using JavaScript.

### **Iframe Syntax**

An HTML iframe is defined with the <iframe> tag:

<iframe src="URL"></iframe>

Here, "src" attribute specifies the web address (URL) of the inline frame page.

## **Set Width and Height of iframe**

You can set the width and height of iframe by using "width" and "height" attributes. By default, the attributes values are specified in pixels but you can also set them in percent. i.e. 50%, 60% etc.

### Example: (Pixels)

<!DOCTYPE html>

<html>

<body>

<h2>HTML Iframes example</h2>

<p>Use the height and width attributes to specify the size of the iframe:</p>

<iframe src="https://www.javatpoint.com/" height="300" width="400"></iframe>

</body>

</html>

### Example: (Percentage)

<!DOCTYPE html>

<html>

<body>

<h2>HTML Iframes</h2>

<p>You can use the height and width attributes to specify the size of the iframe:</p>

<iframe src="https://www.javatpoint.com/" height="50%" width="70%"></iframe>

</body>

</html>

You can also use CSS to set the height and width of the iframe.

### Example:

<!DOCTYPE html>

<html>

<body>

<h2>HTML Iframes</h2>

<p>Use the CSS height and width properties to specify the size of the iframe:</p>

<iframe src="https://www.javatpoint.com/" style="height:300px;width:400px"></iframe>

</body>

</html>

## Remove the border of iframe

By default, an iframe contains a border around it. You can remove the border by using <style> attribute and CSS border property.

### Example:

<!DOCTYPE html>

<html>

<body>

<h2>Remove the Iframe Border</h2>

<p>This iframe example doesn't have any border</p>

<iframe src="https://www.javatpoint.com/" style="border:none;"></iframe>

</body>

</html>

You can also change the size, color, style of the iframe's border.

### Example:

<!DOCTYPE html>

<html>

<body>

<h2>Custom Iframe Border</h2>

<iframe src="https://www.javatpoint.com/" style="border:2px solid tomato;"></iframe>

</body>

</html>

## Iframe Target for a link

You can set a target frame for a link by using iframe. Your specified target attribute of the link must refer to the name attribute of the iframe.

### Example:

<!DOCTYPE html>

<html>

<body>

<h2>Iframe - Target for a Link</h2>

<iframe height="300px" width="100%" src="new.html" name="iframe\_a"></iframe>

<p><a href="https://www.javatpoint.com" target="iframe\_a">JavaTpoint.com</a></p>

<p>The name of iframe and link target must have same value else link will not open as a frame. </p>

</body>

</html>

## Embed YouTube video using iframe

You can also add a YouTube video on your webpage using the <iframe> tag. The attached video will be played at your webpage and you can also set height, width, autoplay, and many more properties for the video.

Following are some steps to add YouTube video on your webpage:

* Goto YouTube video which you want to embed.
* Click on SHARE ➦ under the video.
* Click on Embed <> option.
* Copy HTML code.
* Paste the code in your HTML file
* Change height, width, and other properties (as per requirement).

### Example:

<iframe width="550" height="315" src="https://www.youtube.com/embed/JHq3pL4cdy4" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen style="padding:20px;">

</iframe>

<iframe

width="550" height="315" src="https://www.youtube.com/embed/O5hShUO6wxs" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" style="padding:20px;">

</iframe>

**3. HTML 5 VS. HTML?**

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (at the note for computer) text so that a machine can understand it and manipulate text accordingly. Most of the markup (e.g. HTML) languages are human readable. The language uses tags to define what manipulation has to be done on the text. It is used for structuring and presenting the content on the web pages. HTML5 is the fifth version of HTML. Many elements are removed or modified from HTML5.

There are many differences between HTML and HTML5 which are discussed below:

|  |  |
| --- | --- |
| HTML | HTML5 |
| It didn’t support audio and video without the use of flash player support. | It supports audio and video controls with the use of <audio> and <video> tags. |
| It uses cookies to store temporary data. | It uses SQL databases and application cache to store offline data. |
| Does not allow JavaScript to run in browser. | Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5. |
| Vector graphics is possible in HTML with the help of various technologies such as VML, Silver-light, Flash, etc. | Vector graphics is additionally an integral a part of HTML5 like SVG and canvas. |
| It does not allow drag and drop effects. | It allows drag and drop effects. |
| Not possible to draw shapes like circle, rectangle, triangle etc. | HTML5 allows to draw shapes like circle, rectangle, triangle etc. |
| It works with all old browsers. | It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc. |
| Older version of HTML are less mobile-friendly. | HTML5 language is more mobile-friendly. |
| Doctype declaration is too long and complicated. | Doctype declaration is quite simple and easy. |
| Elements like nav, header were not present. | New element for web structure like nav, header, footer etc. |
| Character encoding is long and complicated. | Character encoding is simple and easy. |
| It is almost impossible to get true GeoLocation of user with the help of browser. | One can track the GeoLocation of a user easily by using JS GeoLocation API. |
| It can not handle inaccurate syntax. | It is capable of handling inaccurate syntax. |
| Attributes like charset, async and ping are absent in HTML. | Attributes of charset, async and ping are a part of HTML 5. |

Many new elements are added in HTML5 like nav, audio, fig caption, progress, command, time, datalist, video, figure, meter, data, section, time, aside, canvas, summary, rp, rt, details, wbr, header, footer, keygen, embed, article, hgroup, bdi, mark, output, source, track, section, ruby and many more.

**4. What is SVG in HTML?**

SVG stands for Scalable Vector Graphic. It can be used to make graphics and animations like in HTML canvas.

The <image> SVG element includes images inside SVG documents. It can display raster image files or other SVG files. The only image formats SVG software must support are JPEG, PNG, and other SVG files.

Syntax:

<image attributes="values" >

Attribute:

* x: x-axis coordinates positioning of the image.
* y: y-axis coordinates positioning of the image.
* width: Width of the image.
* height: Height of the image.
* href: The source of the image.
* preserveAspectRatio: Scaling of the image.
* Global Attributes: Some global attributes used like core attributes and styling attributes, etc.

Example 1:

<!DOCTYPE html>

<html>

<body>

<svg width="200" height="200"

xmlns="http://www.w3.org/2000/svg">

<image

href=

"https://media.geeksforgeeks.org/wp-content/cdn-uploads/20190710102234/download3.png"

height="200" width="200"/>

</svg>

</body>

</html>

Example 2: Changing the x and y coordinates of the image.

<!DOCTYPE html>

<html>

<body>

<svg width="400" height="400"

xmlns="http://www.w3.org/2000/svg">

<image

href=

"https://media.geeksforgeeks.org/wp-content/cdn-uploads/20190710102234/download3.png"

x="100" y="100" />

</svg>

</body>

</html>

**5. Define CSS. What are its components and types?**

CSS (Cascading Style Sheets) is a style-sheet language used to design the webpage to make it attractive. The reason of using CSS is to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

There are three types of CSS which are given below:

* Inline CSS
* Internal or Embedded CSS
* External CSS

CSS is must.

* Base for web development: HTML and CSS is the basic skill that every web developer should know. It is the basic skill that is required for building a website.
* Makes your website look attractive: A website that’s dull and plain will not attract the user most probably, so adding some style would surely make your website presentable to the user.
* Makes the design come live: A web developer is responsible in making the design given to him as a live product. CSS is used for styling to develop the design of the website.
* Increases user experience of website: A website with a simple yet beautiful UI would help the users to go through the website easily. CSS is used to make the user interface better.
* More career opportunities: Since CSS is a basic requirement while learning Web Development, therefor there are abundant career opportunities for it. As a freelancer too you can land up to many projects.

Basic Format:It is the basic structure of HTML webpage and we use CSS style inside webpage. In a web page, we use internal CSS (i.e. adding CSS code inside <head> tag of HTML code).

Example:

<!DOCTYPE html>

<html>

<head>

    <title>

        Simple HTML webpage with CSS style

    </title>

    <!-- Stylesheet of web page -->

    <style>

        body {

            text-align: center;

        }

        h1 {

            color: green;

        }

    </style>

</head>

<body>

    <h1>Welcome</h1>

    <p>Hello this is my world</p>

</body>

</html>

**CSS selectors** are used to "find" (or select) the HTML elements you want to style.

|  |  |  |
| --- | --- | --- |
| Selector | Example | Example description |
| [#*id*](https://www.w3schools.com/cssref/sel_id.asp) | #firstname | Selects the element with id="firstname" |
| [.*class*](https://www.w3schools.com/cssref/sel_class.asp) | .intro | Selects all elements with class="intro" |
| [*element.class*](https://www.w3schools.com/cssref/sel_element_class.asp) | p.intro | Selects only <p> elements with class="intro" |
| [\*](https://www.w3schools.com/cssref/sel_all.asp) | \* | Selects all elements |
| [*element*](https://www.w3schools.com/cssref/sel_element.asp) | p | Selects all <p> elements |
| [*element,element,..*](https://www.w3schools.com/cssref/sel_element_comma.asp) | div, p | Selects all <div> elements and all <p> elements |

With an **external style sheet**, you can change the look of an entire website by changing just one file.Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section. An external style sheet can be written in any text editor, and must be saved with a .css extension. The external .css file should not contain any HTML tags.

An **internal style** sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section.

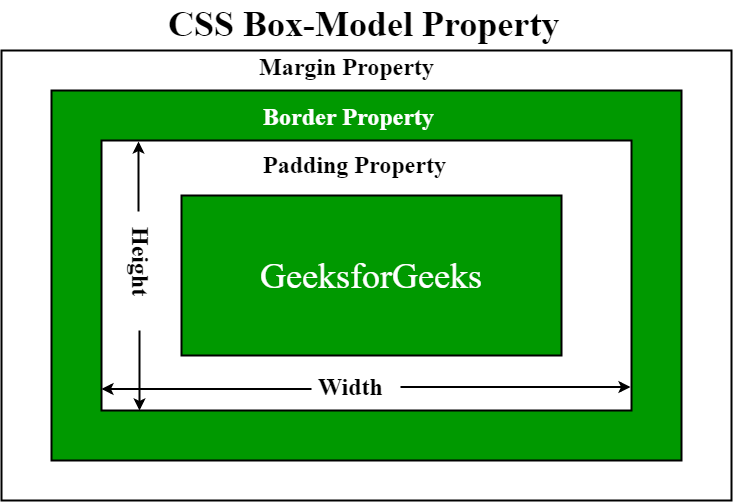
An **inline style** may be used to apply a unique style for a single element.To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

**6. Differentiate CSS box model and flex box with examples.**

CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.   
Box-Model has multiple properties in CSS. Some of them are given below:   
  borders

margins

* padding
* Content



Border Area:It is the area between the box’s padding and margin. Its dimensions are given by the width and height of border.Margin Area:This area consists of space between border and margin. The dimensions of Margin area are the margin-box width and the margin-box height. It is useful to separate the element from its neighbors.Padding Area:It includes the element’s padding. This area is actually the space around the content area and within the border box. Its dimensions are given by the width of the padding-box and the height of the padding-box.Content Area:This area consists of content like text, image, or other media content. It is bounded by the content edge and its dimensions are given by content box width and height.

<!DOCTYPE html>

<head>

<title>CSS Box Model</title>

<style>

.main {

font-size:36px;

font-weight:bold;

Text-align:center;

}

.gfg {

margin-left:60px;

border:50px solid #009900;

width:300px;

height:200px;

text-align:center;

padding:50px;

}

.gfg1 {

font-size:42px;

font-weight:bold;

color:#009900;

margin-top:60px;

background-color:#c5c5db;

}

.gfg2 {

font-size:18px;

font-weight:bold;

background-color:#c5c5db;

}

</style>

</head>

<body>

<div class = "main">CSS Box-Model Property</div>

<div class = "gfg">

<div class = "gfg1">GeeksforGeeks</div>

<div class = "gfg2">A computer science portal for geeks</div>

</div>

</body>

</html>

**CSS flexible box**module is one dimensional layout model.

1. Flexible and efficient layouts.
2. Distribute space among items.
3. Control their alignment.

Before flexbox we had 4 layout modes

1. Block: It is used to make sections in a webpages.
2. Inline: It is used for text.
3. Table: It is used for two dimensional data.
4. Positioned: It is used for explicit position of an element.

Features of flexbox:

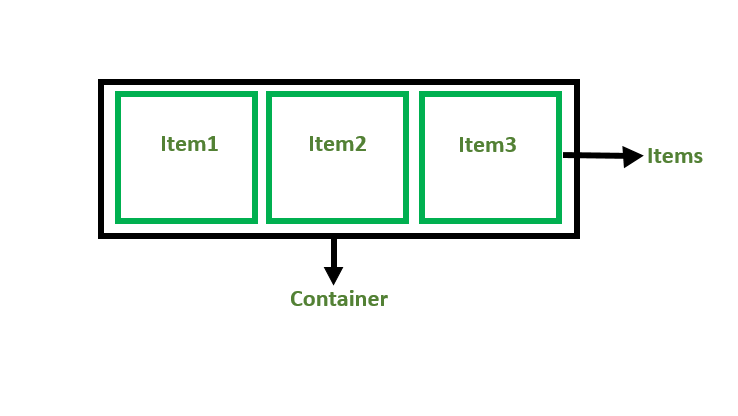
1. A lot of flexibility is given.
2. Arranged items.
3. Proper spacing
4. Alignment of items.
5. Order of items.
6. Bootstrap 4 is build on the top of the flex layout.

Terminologies: There are 2 main components –

1. Flex Container
2. Flex Items

Flex Container: The parent “div” which contains various divisions is called a flex container.

Flex Items:The items inside the container “div” are flex items.

****

<div class="container">

<!-- CONTAINER -->

<div>Item1</div>

<!--ITMES-->

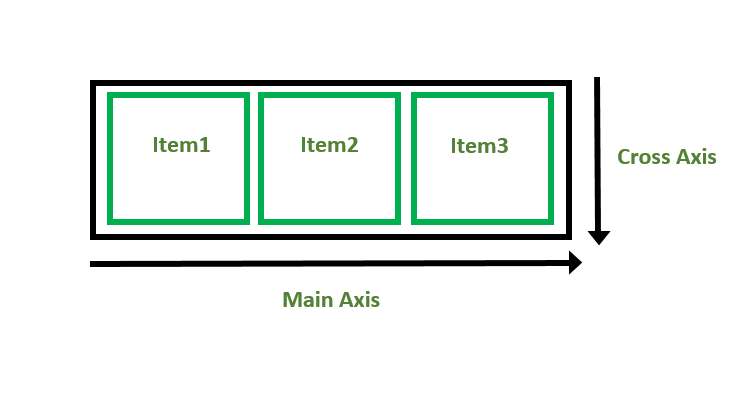
<div>Item2</div>

<div>Item3</div>

</div>

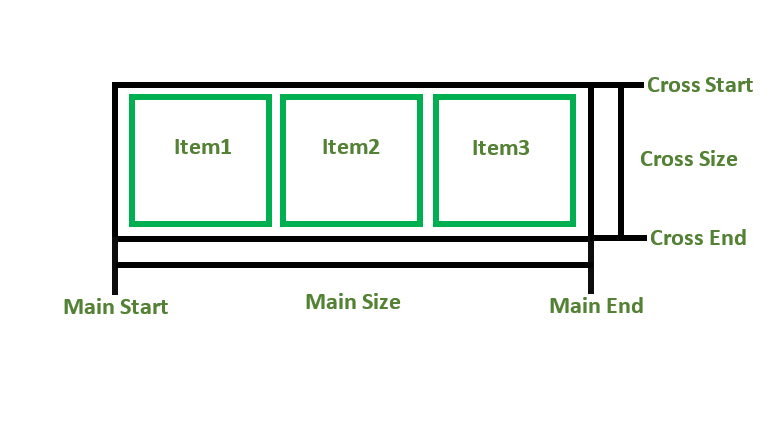
Flexbox Axes: While working with Flexbox, we deal with 2 axis –

1. Main Axis
2. Cross Axis



Main Axis:

* In default, main axis run from left to right.
* **Main Start:** Start of main axis is called Main Start.
* Main Size: The length between Main Start and Main End is called Main Size.
* Main End: The end point is called Main End.
* Main And Cross Axis

  
Cross Axis:

* In default, Cross Axis runs perpendicular to the Main Axis i.e. from top to bottom.
* Cross Start: Start of Cross axis is called Cross Start.
* Cross Size: The length between Cross Start and Cross End is called Cross Size.
* Cross End: The end point is called Cross End.

## **CSS Box Model**

The box model has been around the longest and remains a viable approach to layout design. It relies on margins, borders, and padding to define the spacial relationships between UI elements. Using things like float and clear, these "boxes" of content can be aligned both vertically and horizontally on the page.

### Advantages

The box model has been around for a long time and is the most widely supported by browsers. You'll experience the fewest rendering inconsistencies using the box model.

### Disadvantages

While the box model is fairly easy to understand, it relies on a janky combination of floats and clears to make things work. If you want to achieve a truly responsive design, you'll have to supplement your layout with more media queries using the box model.

## CSS Flexbox

Flexbox is a more recently released layout feature that emphasizes responsive design. Using flexbox, you can specify how elements should grow/shrink to fit different screen sizes. Parent flex container divs contain flex items in either a vertical or horizontal direction.

### Advantages

Flexbox provides a simpler syntax for controlling layout. You don't have to rely on a janky combination of floats and clears or media queries to make things responsive. By specifying a flex direction, flexbox makes it easy to render components like navigation menus and side bars.

### Disadvantages

While flexbox is supported by most major browsers, it's still newer than the traditional box model. This means older browsers don't support it as well (some not at all). There are also more inconsistencies across different browsers. Flexbox also gets complicated with more complex layouts and is limited to only one direction (horizontal or vertical).

**7. What is cell spacing and cell padding?**

Cellpadding:

Cellpadding specifies the space between the border of a table cell and its contents (i.e) it defines the whitespace between the cell edge and the content of the cell.

Syntax:

<table cellpadding="value" >.....</table>

where, value determines the padding

(space between the border of a table and its content)

* Cellspacing:

Cellspacing specifies the space between cells (i.e) it defines the whitespace between the edges of the adjacent cells.

Syntax:

<table cellspacing="value" >.....</table>

where, value determines the padding

(space between adjacent cells)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width,

initial-scale=1.0">

<title>Document</title>

<style>

span{

text-decoration-style: solid;

width: 25px;

font-size: x-large;

color: blueviolet;

}

</style>

</head>

<body>

<table border="1"

cellpadding="4"

cellspacing="5">

<thead>

<td><span>Name</span></td>

<td><span>Age</span></td>

</thead>

<tbody>

<tr>

<td>Rani</td>

<td>30</td>

</tr>

<tr>

<td>Rajan</td>

<td>35</td>

</tr>

<tr>

<td>Akshaya</td>

<td>17</td>

</tr>

<tr>

<td>Ashick</td>

<td>13</td>

</tr>

</tbody>

</table>

</body>

</html>

|  |  |
| --- | --- |
| Cellpadding | Cellspacing |
| It specifies the space between the border of a table cell and its contents. | It specifies the space between adjacent cells. |
| It is created by using HTML <table> tag but type attribute is set to cellpadding. | It is also created by using HTML <table> tag but type attribute is set to cellspacing. |
| It is mainly meant for a single cell. | Cellspacing can get subjected to more than one cell. |
| The default cellpadding value is 1 | Whereas, the default cellspacing value is 2 |
| Cellpadding is widely used and considered to be an effective mean | Cellspacing is less effective than Cellpadding. |
| Cellpadding is an attribute | Cellspacing is also an attribute. |

**8. What is bootstrap?**

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones).

it is Faster and Easier way for Web-Development.

it creates Platform-independent web-pages.

It creates Responsive Web-pages.

It designes the responsive web pages for mobile devices too.

It is Free and open-source framework available on www.getbootstrap.com

**9. Define the terms:**

**· bootstrap carousel**

The Bootstrap Carousel is used to create an image slide show for the webpage to make it look more attractive. It can be included in the webpage using bootstrap.jsor bootstrap.min.js. Carousels are not supported properly in Internet Explorer, this is because they use CSS3 transitions and animations to achieve the slide effect.

Bootstrap 5 provides Carousel which is a slideshow component for cycling through elements. It can be included in the webpage using bootstrap.jsor bootstrap.min.js. Carousels are not supported properly in Internet Explorer, this is because they use CSS3 transitions and animations to achieve the slide effect. It is built with CSS 3D transforms and a bit of JavaScript.

**· bootstrap grid system**

Grid System:Bootstrap Grid System allows up to 12 columns across the page. You can use each of them individually or merge them together for wider columns. You can use all combinations of values summing up to 12. You can use 12 columns each of width 1, or use 4 columns each of width 3 or any other combination.

Grid Classes: The Bootstrap grid system has four classes that can be combined to make more flexible layouts:

* xs (<576px): For Portrait Mobile Phones.
* sm (>=576px): For Landscapes phones
* md (>=768px): For Tablets/Phablets
* lg (>=992px): For Small-sized Desktops/Laptops
* xl (>=1200px): For Larger-sized Desktops/Laptops

**Components of Grid System**:

1. Containers: Bootstrap requires a containing element to wrap site contents and house our grid system. The word ‘container’ is a container of row elements and row elements are ‘containers’ of the column elements. You will understand it more in the latter part of the article where we have dealt with columns.

Use ‘container’ for a responsive fixed width container and use ‘container-fluid’ for a full width container, spanning the entire width of your viewport.

1. Rows: Rows must be placed within a ‘container’ or ‘container-fluid’ for proper alignment and padding. We use rows to create horizontal groups of columns
2. Columns: Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three “col-sm-4”.
3. Column Resets: With the four tiers of grids available, we are bound to run into issues where at certain breakpoints, the columns don’t quite clear right as one is taller( has more text) than the other column. A command called clearfix is there which fixes any issues regarding that viewport. We just need to write a div command with class clearfix after the block where the column isn’t clearing right.
4. Columns Offset: We can move the columns to the right by x columns using  col-md-offset-x in the class.

<div class="col-xs-3 col-sm-4 col-md-6 col-lg-1 col-lg-offset-2“>

This change results in making an offset of 2 grid columns before the fourth column.

1. Nesting Columns: For nesting columns within a column, we need to add a new row and set of columns. Nested rows should include a set of columns that add up to 12 or less than that.

**· bootstrap navbar**

A navigation bar is a navigation header that is placed at the top of the page.A "navbar" is an area on a page that contains navigation components (links, buttons, etc) for getting to other pages of the website.

Navbars require a wrapping .navbar with .navbar-expand{-sm|-md|-lg|-xl} for responsive collapsing and [color scheme](https://getbootstrap.com/docs/4.3/components/navbar/" \l "color-schemes) classes.

* Navbars and their contents are fluid by default. Use [optional containers](https://getbootstrap.com/docs/4.3/components/navbar/" \l "containers) to limit their horizontal width.
* Use our [spacing](https://getbootstrap.com/docs/4.3/utilities/spacing/) and [flex](https://getbootstrap.com/docs/4.3/utilities/flex/) utility classes for controlling spacing and alignment within navbars.
* Navbars are responsive by default, but you can easily modify them to change that. Responsive behavior depends on our Collapse JavaScript plugin.
* Navbars are hidden by default when printing. Force them to be printed by adding .d-print to the navbar.
* Ensure accessibility by using a <nav> element or, if using a more generic element such as a <div>, add a role="navigation" to every navbar to explicitly identify it as a landmark region for users of assistive technologies.

Navbars come with built-in support for a handful of sub-components. Choose from the following as needed:

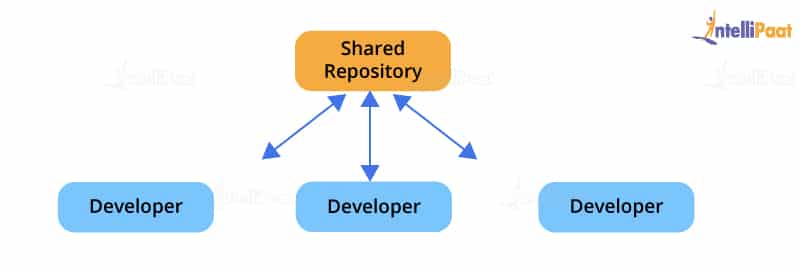
* .navbar-brand for your company, product, or project name.
* .navbar-nav for a full-height and lightweight navigation (including support for dropdowns).
* .navbar-toggler for use with our collapse plugin and other [navigation toggling](https://getbootstrap.com/docs/4.3/components/navbar/" \l "responsive-behaviors) behaviors.
* .form-inline for any form controls and actions.
* .navbar-text for adding vertically centered strings of text.
* .collapse.navbar-collapse for grouping and hiding navbar contents by a parent breakpoint.

**10. What is GIT? Explain the Git architecture?**

Git is a distributed version control system (DVCS) for tracking changes to files.Git is an open-source VCS, which is not file-based, unlike other systems. Rather, it stores information as snapshots. Being a VCS, helps coders revert to their previous code when they hit a roadblock in the newer version, without affecting the original source code. On the other hand, what makes it different from other VCS is the way it sees data, which is more like a series of snapshots. It basically clicks a picture of how all your files look at the moment and saves the changes made to them over time.

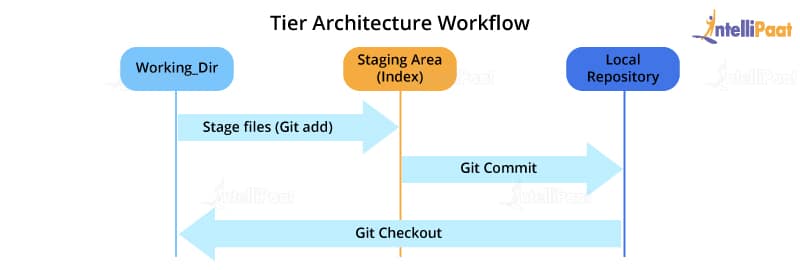
## **Features of Git**

* **Works on a distributed system**: A distributed system is one that allows collaborators to access the central repository using a VCS, even from the remotest corner of the world. As Git maintains a snapshot every time a user pulls a file, the risk of data loss due to system failure or lack of Internet connection is mitigated. Users are allowed to work on the same bit of code simultaneously without getting interfered by others.



* **Compatible with all operating systems**: Git is compatible with almost all operating systems that are available today. Even the repositories created by other version control systems can be accessed by the Git repository.
* **Allows for non-linear development**: As users from remote parts of the world can access the Git repository, work on it, and update the project at any time they want, Git allows for development in a non-linear fashion. Git supports such a kind of development by providing its branching and merging features, and it uses specific tools for navigating through them. The projects are viewed in a tree form.
* **Branches like a tree**: While users are working on their projects, branches parallel to the main project file are created, so the original code is not affected. There is no restriction upon the number of branches created.
* **Light as a cotton ball**: One might think that making multiple copies of data from a central repository to a local one will eventually lead the system to crash due to overload. But, Git has got it covered. It compresses the data in such a way that it takes up minimal space, and whenever you need to retrieve data, the reverse technique is used. This helps save a lot of memory.
* **Fast as a flash**: Unlike other version control systems, Git is written in a language known to be the closest to the machine language, that is, C. Hence, it processes information much faster.
* **Reliable**: There will never be an issue of data loss as long as the copies of data in the central repositories are available in the local repositories of different collaborators.

## **Git Architecture**



Most of the version control systems have a two-tier architecture. However, Git has a layer more, making it a three-tier architecture. But, why are there three layers of Git? Let’s find out.

**The three layers are:**

* **Working directory**: This is created when a Git project is initialized onto your local machine and allows you to edit the source code copied.
* **Staging area**: Post the edits, the code is staged in the staging area by applying the command, **git add**. This displays a preview for the next stage. In case further modifications are made in the working directory, the snapshots for these two layers will be different. However, these can be synced by using the same ‘git add’ command.
* **Local repository**: If no further edits are required to be done, then you can go ahead and apply the **git commit** command. This replicates the latest snapshots in all three stages, making them in sync with each other.

**11. Difference between 1. GIT and GitHub**

**Git**: Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

**GitHub:** GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

| **S.No.** | **Git** | **GitHub** |
| --- | --- | --- |
| 1. | Git is a software. | GitHub is a service. |
| 2. | Git is a command-line tool | GitHub is a graphical user interface |
| 3. | Git is installed locally on the system | GitHub is hosted on the web |
| 4. | Git is maintained by linux. | GitHub is maintained by microsoft. |
| 5. | Git is focused on version control and code sharing. | GitHub is focused on centralized source code hosting. |
| 6. | Git is a version control system to manage source code history. | GitHub is a hosting service for Git repositories. |
| 7. | Git was first released in 2005. | GitHub was launched in 2008. |
| 8. | Git has no user management feature. | GitHub has built-in user management feature. |

**2. git pull and git fetch?**

Pulling or Fetching of data from a central repository is done to update the collaborator’s local copy of the repository. This helps to replace the older version with the latest one. This process is done by the use of a Git command termed as git pullor git fetch.

Pullcommand and fetchcommand can be both used for the same job, as they both update the repository to the latest version. The only difference between a fetch command and a pull command is that the fetchcommand only updates the remote branches and not the local branch, while pullcommand updates both the local and remote repositories

**12. What is java script? List its features.**

JavaScript is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for [Client-side](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments as well as [Server-side](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments. JavaScript contains a standard library of objects, like [Array](https://www.geeksforgeeks.org/arrays-in-javascript/), [Date](https://www.geeksforgeeks.org/javascript-date-objects/), and [Math](https://www.geeksforgeeks.org/javascript-math-object/), and a core set of language elements like operators, control structures, and statements.

Client-side: It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Useful libraries for the client-side are [AngularJS](https://www.geeksforgeeks.org/introduction-to-angularjs/), [ReactJS](https://www.geeksforgeeks.org/react-js-introduction-working/), VueJS and so many others.

* Server-side: It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is [node.js](https://www.geeksforgeeks.org/introduction-to-nodejs/).

JavaScript can be added to your HTML file in [two ways](https://www.geeksforgeeks.org/where-to-put-javascript-in-an-html-document/):

* Internal JS: We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
* External JS: We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

**Syntax:**

<script>

// JavaScript Code

</script>

**Features of JavaScript:**

JavaScript is the most popular language on earth.   
With advances in browser technology and JavaScript having moved into the server with Node.js and other frameworks, JavaScript is capable of so much more.

* JavaScript was created in the first place for DOM manipulation. Earlier websites were mostly static, after JS was created dynamic Web sites were made.
* Functions in JS are objects. They may have properties and methods just like another object. They can be passed as arguments in other functions.
* Can handle date and time.
* Performs Form Validation although the forms are created using HTML.
* No compiler needed.

**Applications of JavaScript:**

* Web Development: Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.
* Web Applications: With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces(APIs) that provide extra power to the code. The Electron and React is helpful in this department.
* Server Applications: With the help of Node.js, JavaScript made its way from client to server and node.js is the most powerful in the server-side.
* Games: Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.
* Smartwatches: JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the internet for its functioning.
* Art: Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, make the sound more effective also can be used [**p5.js**](https://www.geeksforgeeks.org/p5-js-introduction/) library.
* Machine Learning: This JavaScript ml5.js library can be used in web development by using machine learning.

**Limitations of JavaScript:**

* Performance: JavaScript does not provide the same level of performance as offered by many traditional languages as a complex program written in JavaScript would be comparatively slow. But as JavaScript is used to perform simple tasks in a browser, so performance is not considered a big restriction in its use.
* Complexity: To master a scripting language, programmers must have a thorough knowledge of all the programming concepts, core language objects, client and server-side objects otherwise it would be difficult for them to write advanced scripts using JavaScript.
* Weak error handling and type checking facilities: It is weakly typed language as there is no need to specify the data type of the variable. So wrong type checking is not performed by compile.

**13. What is the difference between Local storage & Session storage?**

**SessionStorage** and **LocalStorage** are known as the web storage API. Data can be stored on the client side by using these APIs.

SessionStorage:

* SessionStorage is used for storing data on the client side.
* Maximum limit of data saving in SessionStorage is about 5 MB.
* Data in the SessionStorage exist till the current tab is open if we close the current tab then our data will also erase automatically from the SessionStorage.

LocalStorage:

* Like SessionStorage, LocalStorage also used for storing the data on the client side.
* Maximum limit of data saving is about 5 MB in LocalStorage also.
* LocalStorage has no expiration time, Data in the LocalStorage persist till the user manually delete it. This is the only difference between LocalStorage and SessionStorage
* Both are Object type
* Format of storing data in SessionStorage and LocalStorage: Data must be stored in key-value pair in the SessionStorage and LocalStorage and key-value must be either number or string
* For storing data in web storage:

LocalStorage.setItem("key", "value"); //key and value both should be string or number;

SessionStorage.setItem("key", "value"); //key and value both should be

* For getting data from web storage:

LocalStorage.getItem("key");

SessionStorage.getItem("key");

* For Getting the length of web storage object:

LocalStorage.length;

SessionStorage.length;

* For deleting a particular key-value pair:

LocalStorage.removeItem("key");

SessionStorage.removeItem("key");

* For clearing complete storage:

LocalStorage.clear();

SessionStorage.clear();

* For getting nth key name from web storage we will pass the number n:

LocalStorage.key(n);

SessionStorage.key(n);

**14. What is 'this' keyword in JavaScript?**

The JavaScript this keyword refers to the object it belongs to.

The this keyword behaves differently in JavaScript compared to other languages. In Object Oriented languages, the this keyword refers to the current instance of the class. In JavaScript the value of this is determined by the invocation context of function ( context. function() ) and where it is called.

It has different values depending on where it is used:

* In a method, this refers to the owner object.
* Alone, this refers to the global object.
* In a function, this refers to the global object.
* In a function, in strict mode, this is undefined.
* In an event, this refers to the element that received the event.
* Methods like call(), and apply() can refer this to any object.

**15. What are all types of Pop-up boxes available in JavaScript?**

JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

## **Alert Box**

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

### Syntax

window.alert("*sometext*");

The window.alert() method can be written without the window prefix.

## **Confirm Box**

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

### Syntax

window.confirm("*sometext*");

The window.confirm() method can be written without the window prefix.

## **Prompt Box**

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

### Syntax

window.prompt("*sometext*","*defaultText*");

The window.prompt() method can be written without the window prefix.

## **Line Breaks**

To display line breaks inside a popup box, use a back-slash followed by the character n.

### Example

### alertHello\nHow are you?");

**16. What is the difference between === operator and == operator? Explain with an example.**

**Double equals (==)** is a comparison operator, which transforms the operands having the same type before comparison.

So, when you compare string with a number, JavaScript converts any string to a number. An empty string is always converts to zero. A string with no numeric value is converts to NaN (Not a Number), which returns false.

The == operator is an equality operator. It checks whether its two operands are the same or not by changing expression from one data type to others. You can use == operator in order to compare the identity of two operands even though, they are not of a similar type.

## Example of ==

In the below program, we have declared one variable "a" having value 10. Lastly, the statement a == 20 returns false as the value of a is 10.

<!DOCTYPE html>

<html>

<body>

<p id="demonstration"></p>

<script>

var a = 10;

document.getElementById("demonstration").innerHTML = (a == 20);

</script>

</body>

</html>

o/p:false

**=== (Triple equals)** is a strict equality comparison operator in JavaScript, which returns false for the values which are not of a similar type. This operator performs type casting for equality. If we compare 2 with "2" using ===, then it will return a false value.

## How === Works Exactly?

* Strict equality === checks that two values are the same or not.
* Value are not implicitly converted to some other value before comparison.
* If the variable values are of different types, then the values are considered as unequal.
* If the variable are of the same type, are not numeric, and have the same value, they are considered as equal.
* Lastly, If both variable values are numbers, they are considered equal if both are not NaN (Not a Number) and are the same value.

## Example of ===

In the below program, the value of variable x is 10. It is compared to 10 written in double-quotes, which is considered as a string, and therefore, the values are not strictly the same. The output of the program is false.

<!DOCTYPE html>

<html>

<body>

<p id="demo"></p>

<script>

var x = 10;

document.getElementById("demo").innerHTML = (x === "10");

</script>

</body>

</html>

**Output:**

false

| **=** | **==** | **===** |
| --- | --- | --- |
| = in JavaScript is used for assigning values to a variable. | == in JavaScript is used for comparing two variables, but it ignores the datatype of variable. | === is used for comparing two variables, but this operator also checks datatype and compares two values. |
| It is called as assignment operator | It is called as comparison operator | It is also called as comparison operator |
| The assignment operator can evaluate to the assigned value | Checks the equality of two operands without considering their type. | Compares equality of two operands with their types. |
| It does not return true or false | Return true if the two operands are equal. It will return false if the two operands are not equal. | It returns true only if both values and data types are the same for the two variables. |
| = simply assign one value of variable to another one. | == make type correction based upon values of variables. | === takes type of variable in consideration. |
| == will not compare the value of variables at all. | The == checks for equality only after doing necessary conversations. | If two variable values are not similar, then === will not perform any conversion. |

## **KEY DIFFERENCES:**

* = is used for assigning values to a variable, == is used for comparing two variables, but it ignores the datatype of variable whereas === is used for comparing two variables, but this operator also checks datatype and compares two values.
* = is called as assignment operator, == is called as comparison operator whereas It is also called as comparison operator.
* = does not return true or false, == Return true only if the two operands are equal while === returns true only if both values and data types are the same for the two variables.

**17. Explain what is pop () and push () method in JavaScript?**

**18. Explain try n catch concept in java string using examples.**

**19. Explain error and exception handling with examples.**

**20. Write a program to reverse a string.**

**21. Write a JavaScript program to find the Armstrong numbers of 3 digits.**

**22. Write a JavaScript program to construct the following pattern, using a nested for loop.**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**23. Write a JavaScript program which computes the average mark of a student (5 subjects) and assign corresponding grades.**

**24. What is DOM?**

**25. Design a webpage of your interest with technologies that you have learned.**