HTML

HTML stands for Hyper Text Markup Language. It used for creating and describes the structure of web pages. Hypertext defines the link between the web pages and markup language defines the text document within the tag. manipulate text, images, and other content, in order to display it in the required format.

An HTML document can be created using any text editor. Save the text file using .html or .htm. Once saved as an HTML document, the file can be opened as a webpage in the browser.

<!DOCTYPE html>

<html>

<head>

<title>Demo Web Page</title>

</head>

<body>

<h1>Anamika </h1>

BackEnd Trainee at Beinex

</body>

</html>

It is the structure of html page.

HTML Elements

Elements are the building blocks of HTML that describe the structure and content of a web page.

It contains a start tag, end tag, and some content enclosed between these two tags.

p Element

The tag stands for paragraph, which is the most common tag used to create lines of text inside an HTML document.

This is a paragraph.

Heading element

There are six heading elements

- <h1>, <h2>, <h3>, <h4>, <h5>, <h6>.

When higher the number of heading, lower the importance.
That is importance of <h1>is higher than <h2>, importance of <h2> is higher than <h3>. <h6> is lesser importance than other headings.

Example

```
<h1>This is main heading.</h1>
<h2>This is subheading.</h2>
<h3>This is subheading.</h3>
<h4>This is subheading.</h4>
<h5>This is subheading.</h5>
<h6>This is subheading.</h6>
```

a Element

The anchor (<a>) element creates a hyperlink to another page or file. In order to link to a different page or file the <a> tag must also contain a href attribute, which indicates the link's destination.

List Elements

There are two main types of lists

1. ordered (<ol>)
2. unordered (<ul>).

All lists must contain one or more list elements (<li>).

Unordered List:

The unordered list starts with <ul> tag and list items start with the <li> tag. In unordered lists all the list items marked with bullets by default.

```
<ul>
  <li>Item</li>
  <li>Item</li>
  <li>Item</li>
</ul>
```

Output:

- Item
- Item
- Item

Ordered List:

The ordered list starts with <ol> tag and list items start with the <li> tag. In ordered lists all the list items are marked with numbers.

```
<ol>
  <li>First Item</li>
  <li>Second Item</li>
```

```
  <li>Third Item</li>
</ol>
```
Output:

1. First Item
2. Second Item
3. Third Item

img Element

A simple HTML <img> element can be included in an HTML document like this:
Note that <img> elements are self-closing, and do not require a closing tag.

nav Element

The <nav> element is intended for major block of navigation links. NOT all links of a document should be inside a <nav> element.
Example

```
<nav class="menu">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>
```

div Element

The tag is a generic container that defines a section in your HTML document. A element is used to group sections of HTML elements together and format them with CSS.
A is a block-level element.
Here is an example of how to display a section in the same color:

```
<div style="color:#ff0000">
  <h3>my heading</h3>
  <p>my paragraph</p>
</div>
```

HTML Formatting Elements

- <b> - Bold text
- - Important text
- <i> - Italic text

- - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

Attributes

An attribute is used to provide extra or additional information about an element. It takes two parameters name and value. The name parameter takes the name of the property we would like to assign to the element and the value takes the properties value or extent of the property names that can be aligned over the element. Every name has some value that must be written within quotes.

Comment Types

There are three types of comments in HTML which are:

Single-line Comment:

The single-line comment is given inside the (<!– comment –>) tag.

Syntax

```
<!-- comment -->
```

Multi-line Comment:

Multiple lines can be given by the syntax (<!– –>), Basically it's the same as we used in single line comment, difference is half part of the comment (" –> "), is appended where the intended comment line ends.

Syntax

```
<!-- Multi
Line
Comment -->
```

Using comment tag

There used to be an HTML **<comment>** tag, but currently it is not supported by any modern browser.

Syntax

```
<comment> Some Comment </comment>
```

CSS

CSS (Cascading Style Sheets) is the code that styles web content. Like HTML, CSS is not a programming language. It's not a markup language either. **CSS is a style sheet language.** CSS is what you use to selectively style HTML elements.

It is used for styling web documents written in HTML or XML. It allows you to control the presentation and layout of a web page. Here are some CSS basics:

There are 3 ways to write CSS in our HTML file.

Inline CSS,Internal CSS and External CSS

Priority order:

Inline > Internal > External

- Inline CSS: Before CSS this was the only way to apply styles. It is not an efficient way to write as it has a lot of redundancy. It is self-contained and uniquely applied on each element

Example:

<h3 style=" color:red"> Have a great day </h3>

 I did this , I did that

- Internal CSS: With the help of style tag, we can apply styles within the HTML file. Redundancy is removed. But the idea of separation of concerns still lost. Uniquely applied on a single document

Example:

```
< style>
        H1{
            Color:red;
           }
 </style>
<h3> Have a great day </h3>
```

- External CSS:With the help of <link> tag in the head tag, we can apply styles. Reference is added and file saved with .css extension. Redundancy is removed and the idea of separation of concerns is maintained. Uniquely applied to each document

Example:

```
<head>

 <link rel=”stylesheet” type=”text/css” href=”name of the Css file”>

</head>

        H1{

              Color:red;        //.css file

            }
```

Selectors: Selectors target HTML elements you want to style. They can be simple, like targeting a specific HTML tag (p for paragraphs), or more complex, using classes (.classname) or IDs (#idname).

Properties and Values: CSS properties define the style aspects, and values specify the settings. For example, the color property can have values like red or #FF0000.

Declaration Block: A set of CSS declarations enclosed in curly braces {}. Each declaration consists of a property and a value, separated by a colon.

Comments: You can add comments in CSS using /* comment here */. Comments are ignored by browsers and are useful for explaining your code.

Box Model: Fundamental to layout, it consists of content, padding, border, and margin. The width and height properties control the size of the content box.

Positioning: CSS provides various positioning schemes, including static, relative, absolute, and fixed. Understanding these helps in controlling the layout.

Properties and Values: Color: Use color names (red) or hexadecimal codes (#FF0000).

Font: Specify font-family, font-size, font-weight, and more for text styling.

Margin and Padding: Control spacing around elements.

Border: Define borders using border property with values for width, style, and color.

Flexbox and Grid: These layout models provide more sophisticated ways to design responsive and dynamic layouts. Flexbox is good for one-dimensional layouts, while Grid is suitable for two-dimensional layouts.

Typography: CSS allows you to control font properties like font-family, font-size, font-weight, etc., to define text appearance.

Colors and Backgrounds: You can specify background colors, images, and control foreground text colors using properties like background-color and color.

Transitions and Animations: CSS supports transitions for smooth property changes and animations for more complex effects, enhancing user experience.

JAVASCRIPT

JavaScript is a programming language that adds interactivity to your website. This happens in games, in the behavior of responses when buttons are pressed or with data entry on forms; with dynamic styling; with animation, etc.

JavaScript is a powerful programming language that can add interactivity to a website. It was invented by Brendan Eich. JavaScript is versatile and beginner-friendly.

Applications of JavaScript:

- Web Development
- Web Applications
- Server Applications
- Games
- Machine learning
- Mobile app

JavaScript tools:

- Browser Application Programming Interfaces (APIs) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.

- Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as Twitter or Facebook.
-
- Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications.

JavaScript Variables:

JavaScript Variables are the building blocks of any programming language. In JavaScript, variables can be used to store reusable values. The values of the variables are allocated using the assignment operator("=").

JavaScript Identifiers:

JavaScript variables must have unique names. These names are called Identifiers.

There are some basic rules to declare a variable in JavaScript:

- These are case-sensitive
- Can only begin with a letter, underscore("_") or "$" symbol
- It can contain letters, numbers, underscore, or "$" symbol
- A variable name cannot be a reserved keyword.

JavaScript is a dynamically typed language so the type of variables is decided at runtime. Therefore there is no need to explicitly define the type of a variable. We can declare variables in JavaScript in three ways:

1. JavaScript var keyword
2. JavaScript let keyword
3. JavaScript const keyword

Syntax:

Var geek = "Hii"

Let $ = "Welcome"

Const _example = "new"

Variables are containers that store values. Variables may hold values that have different datatypes

Datatypes:

String: This is a sequence of text known as a string. To signify that the value is a string, enclose it in single or double quote marks

Number: This is a sequence of text known as a string. To signify that the value is a string, enclose it in single or double quote marks.

Boolean: This is a True/False value. The words true and false are special keywords that don't need quote marks.

Array: This is a structure that allows you to store multiple values in a single reference.

Object: This can be anything. Everything in JavaScript is an object and can be stored in a variable. Keep this in mind as you learn.

JS Control Flow Statements:

JavaScript control flow statement is used to control the execution of a program based on a specific condition. If the given condition meets then a particular block will be executed otherwise it will execute another block of action that satisfies that particular condition.

- Return Statement
- Break Statement
- Continue Statement
- Throw Statement
- If…else Statement
- Switch Statement
- Try…catch Statement

JS Loops:

JavaScript provides several types of loops to iterate over a block of code. Loops in JavaScript execute a block of code again and again till the condition is false.

- For Loop
- Do…while Loop
- While Loop
- For…in Loop
- For…of Loop
- labeled Statement
- break Statement
- continue Statement

JS Expression and Operators:

JavaScript expressions are the combinations of values, variables, and operators that can be evaluated to produce a result. Operators are symbols that perform operations on operands.

- Assignment operators
- Comparison operators
- Arithmetic operators
- Bitwise operators
- Logical operators
- BigInt Operators
- String operators
- Ternary operator
- Comma operator
- Unary operators
- Relational operators

Comments:

Comments are snippets of text that can be added along with code. The browser ignores text marked as comments. You can write comments in JavaScript just as you can in CSS:

```
/*
Everything in between is a comment.
*/
```

If your comment contains no line breaks, it's an option to put it behind two slashes like this:

```
// This is a comment
```

Conditionals:

Conditionals are code structures used to test if an expression returns true or not. A very common form of conditionals is the if...else statement.

Functions:

Theya are a way of packaging functionality that you wish to reuse. It's possible to define a body of code as a function that executes when you call the function name in your code.