Republic of the Philippines
Sorsogon State University
COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Bulan Campus
Zone-8, Bulan, Sorsogon
Tel. No.; (056) 311 9800; Email Address: cict@sorsu.edu.ph

BAGONG PILIPINAS

IT 224

WEB SYSTEMS AND TECHNOLOGIES

PREPARED BY:
KENNETH L. GISALAN
Instructor I

MODULE 1

INTRODUCTION

The Basics of Web Development

There are three interactive elements on the Internet:

- Websites: a collection of information and files that we access via a computer and a server.
- Servers: computers which store all website data in an extensive network.
- Browsers: software that loads and displays information on your computer.

Two parts of Website

- **Frontend (Client Side) -** The frontend comprises everything the user sees and experiences instantly while visiting a website.
- **Backend (Server Side) -** The backend is behind the scenes that store, send and receive information.

HTML, **CSS**, and **JavaScript** files make up everything you see on a website. As a web developer, these are the most basic tools you'll need. They are the languages that you need to build your websites.

What is HTML?

The basic programming language for web creation is **HTML (HyperText Markup Language).** It contains the essential elements of a website, such as words, titles, and paragraphs. HTML is made up of a set of pre-defined tags that represent various functions and subsequently "translate" into understandable information on the screen. These tags are always written between angle brackets.

What is CSS?

CSS (Cascading Style Sheets) is a style sheet that describes how HTML components appear on a page. CSS is used to manage your website's appearance, style, and formatting, including RGB values, border colors, background pictures, and more. CSS files specify a set of rules for defining a set of properties and their values.

What is JavaScript?

JavaScript is a scripting language that allows you to control the behavior of your website. If you're serious about web programming, you should begin by learning the fundamentals of JavaScript. It is one of the most widely used programming languages in the world, with a low entry barrier and instant results based on your code's success. By manipulating different HTML and CSS elements, JavaScript makes web pages interactive. Using JavaScript, a user may click a button, scroll to the bottom of a page, or see pictures in a rolling carousel.

The Technology Stack

A tech stack is a collection of software, apps, programming languages, and tools that work together to create a website. Different tech stacks will be used by companies and web developers depending on their specific goals and needs.

What are Frameworks?

A framework for your website is similar to a pre-packaged structure of pre-written code that defines how programs should interact. It comes with pre-installed apps, plug-ins, and tools that you may add to your website files. Frameworks help you develop faster, plus they're written by other developers, so you know the code has been well tested.

Front End Frameworks

CSS and JavaScript frameworks are collections of CSS or JS files that share common functionality to execute various tasks. Instead of starting with a blank text page, you begin with a code file that already contains a lot of JavaScript.

Frameworks each have their own set of advantages and disadvantages, making it critical to select the right framework for the kind of website you're creating. Some JS frameworks, for example, are excellent at creating sophisticated user interfaces, while others excel at presenting all of your website's information.

The following are some of the most common frameworks:

- JQuery
- Angular
- React
- Vue
- Next
- Nest

Back End Skills

The back-end layer establishes a dynamic link between the front-end and the database. To get this layer to operate, you must be familiar with at least one programming language, such as Python, Java, PHP, Ruby, and others, as well as server-side frameworks such as NodeJS.

Data and Database

The data layer is a large information warehouse. It includes a database repository that collects and saves data from the front end to the back end. Knowledge of how data is saved, changed, retrieved, and so on is required. Working knowledge of databases such as MySQL and MongoDB is also needed.

Server and Deployment Skills

Servers are computers that house website files as well as other resources such as databases.

Server Setup

A website must be placed on a server in order to be publicly accessible on the internet. It's time to set up the site on the server after you have your domain name and server space. The first step is to point the domain name to the unique IP address of the server. The website files must next be set up, followed by the database and other specifications.

Deployment Tools

A protocol is required to transfer files from your PC to your server. This is a way of sending and receiving files and other data to and from a server. When a change is pushed in Git to the master branch, the deployment tool remembers your FTP/SFTP settings and transfers the files for you. As a result, you don't have to remember which files you altered, which reduces the number of mistakes you make.

Web Browsers

This is where you'll put your code to the test. A competent web developer should be able to work with a variety of browsers, but it's acceptable to start with one and remain with it until you're more comfortable.

The most widely used browsers among web developers are listed below.

- Safari - excellent development tools, but not compatible with Windows.
- Opera - free VPS service with identical drawbacks as Internet Explorer.
- Brave - performance-driven, secure, with limited plug-in support.
- Mozilla Firefox - an open-source browser that runs on all platforms and receives frequent upgrades.
- Edge - integrated with Windows, having the same drawbacks as Internet Explorer.
- Google Chrome - DevTools, which makes JavaScript troubleshooting simple, consumes a lot of memory

Content Management System (CMS)

A Content Management System (CMS) is software that makes it easier to edit, create, and publish your work. A content management system (CMS) provides you complete control over your content by combining all of your site's functions into one easy-to-use platform. Many of these applications assist with content management, marketing, and distribution.

A CMS usually makes use of a database (such as MySQL or MariaDB) to hold a collection of applications and tools written in a certain programming language. You may then update and maintain your website without having to return to the code's finer points. The number of users, the size of your staff, and the simplicity of the interface all play a role in selecting a CMS. WordPress is a fantastic CMS for beginners. It's a free, open-source website-building and-publishing tool. It's a popular blogging platform with a low entrance hurdle for newcomers. With such a large network of support, you'll almost certainly be able to discover solutions to your issues as they arise.

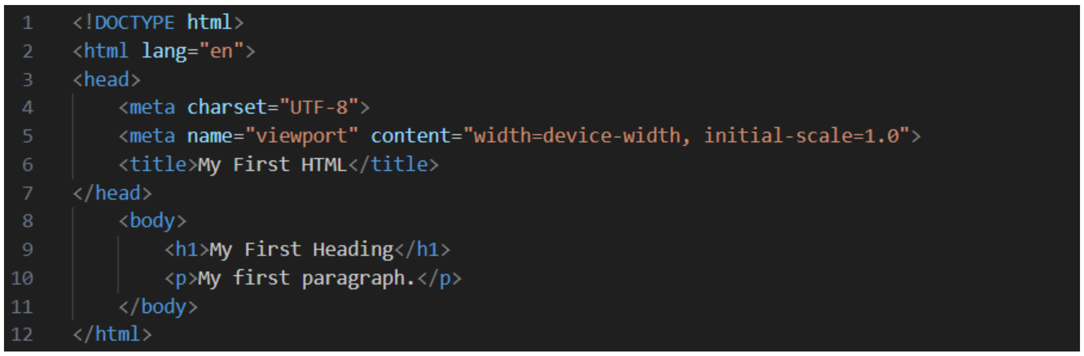
MODULE 2

INTRODUCTION TO HTML

What is HTML?

HTML is the standard markup language for creating Web pages. It describes the structure of a Web page. HTML consists of a series of elements that tells the browser how to display the content.

Simple HTML Document



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My First HTML</title>
</head>
    <body>
        <h1>My First Heading</h1>
        <p>My first paragraph.</p>
    </body>
</html>
```

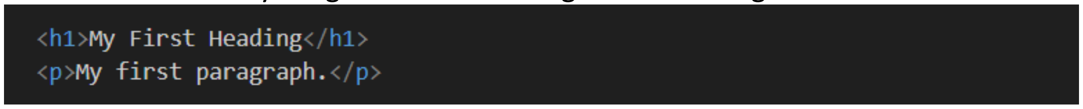
- The **<!DOCTYPE html>** declaration defines that this document is an HTML5 document.
- The **<html>** element is the root element of an HTML page.
- The **<head>** element contains meta information about the HTML page.
- The **<title>** element specifies a title for the HTML page (which is shown in 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.

What is HTML Element?

An HTML element is defined by a start tag, some content, and an end tag:

<tagname> Content goes here... </tagname>

The HTML **element** is everything from the start tag to the end tag:



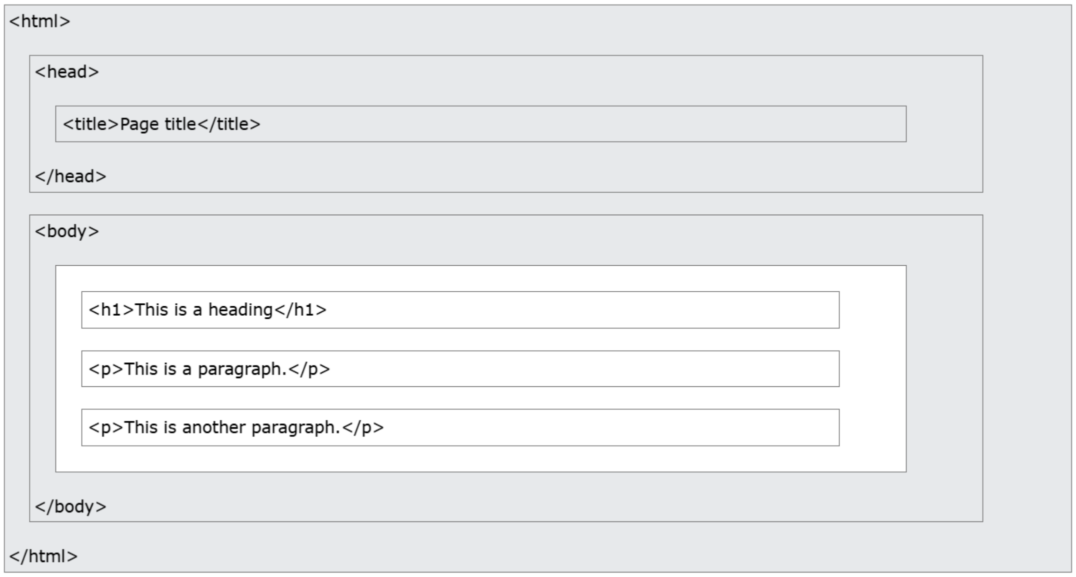
```html
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

Note: *Some HTML elements have no content (like the* ***<br>*** *element). These elements are called empty elements. Empty elements do not have an end tag.*
For a complete list of all available HTML tags, visit HTML Tag Reference.

Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly. A browser does not display the HTML tags, but uses them to determine how to display the document.

HTML Page Structure



```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
      <h1>This is a heading</h1>
      <p>This is a paragraph.</p>
      <p>This is another paragraph.</p>
  </body>
</html>
```

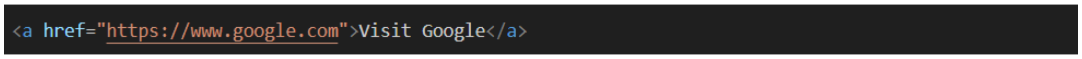
HTML Attributes

HTML attributes provide additional information about HTML elements.

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: ***name="value".***

The *href* Attribute

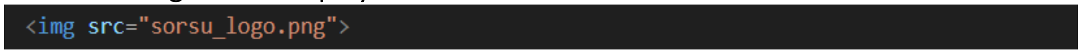
The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to.



```
<a href="https://www.google.com">Visit Google</a>
```

The *src* Attribute

The ***<img>*** tag is used to embed an image in an HTML page. The ***src*** attribute specifies the path to the image to be displayed.



```
<img src="sorsu_logo.png">
```

There are two ways to specify the URL in the ***src*** attribute:

- **Absolute URL** - Links to an external image that is hosted on another website.
  - Example: ***src="https://www.sample.com/images/img_girl.jpg"***.

- **Relative URL** - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page.
    - Example: ***src="img_girl.jpg"***.
    - If the URL begins with a slash, it will be relative to the domain.
    - Example: ***src="/images/img_girl.jpg"***.

Notes: *External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.*

Tip: *It is almost always best to use relative URLs. They will not break if you change domain.*

The *width* and *height* Attributes

The <img> tag should also contain the width and height attributes, which specify the width and height of the image (in pixels).

```html
<img src="img_girl.jpg" width="500" height="600">
```

The *alt* Attribute

The required ***alt*** attribute for the ***<img>*** tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to a slow connection, or an error in the ***src*** attribute, or if the user uses a screen reader.

```html
<img src="sorsu_logo.png" alt="The SorSU Logo">
```

The *style* Attribute

The ***style*** attribute is used to add styles to an element, such as color, font, size, and more.

```html
<p style="color:red;">This is a red paragraph.</p>
```

The *lang* Attribute

You should always include the ***lang*** attribute inside the ***<html>*** tag, to declare the language of the Web page. This is meant to assist search engines and browsers.
The following example specifies English as the language:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    ....
</head>
    <body>
        ....
    </body>
</html>
```

Country codes can also be added to the language code in the lang attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.

The following example specifies English as the language and United States as the country:

```html
<!DOCTYPE html>
<html lang="en-US">
<head>
    ....
</head>
    <body>
        ....
    </body>
</html>
```

You can see all the language code references here: Language Code References.

The title Attribute

The ***title*** attribute defines some extra information about an element. The value of the title attribute will be displayed as a tooltip when you mouse over the element.

```html
<p title="I'm a tooltip">This is a paragraph.</p>
```

***Tip**: Always use lowercase attributes.*

Single or Double Quotes?

Double quotes around attribute values are the most common in HTML, but single quotes can also be used. In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes.

```html
<p title='John "ShotGun" Nelson'>
```

Summary

- All ***HTML*** elements can have **attributes**
- The ***href*** attribute of ***<a>*** specifies the URL of the page the link goes to
- The ***src*** attribute of ***<img>*** specifies the path to the image to be displayed
- The ***width*** and ***height*** attributes of ***<img>*** provide size information for images
- The ***alt*** attribute of ***<img>*** provides an alternate text for an image
- The ***style*** attribute is used to add styles to an element, such as color, font, size, and more
- The ***lang*** attribute of the ***<html>*** tag declares the language of the Web page
- The ***title*** attribute defines some extra information about an element

Complete list of all attributes for each HTML element are listed here: HTML Attribute Reference.

MODULE 3

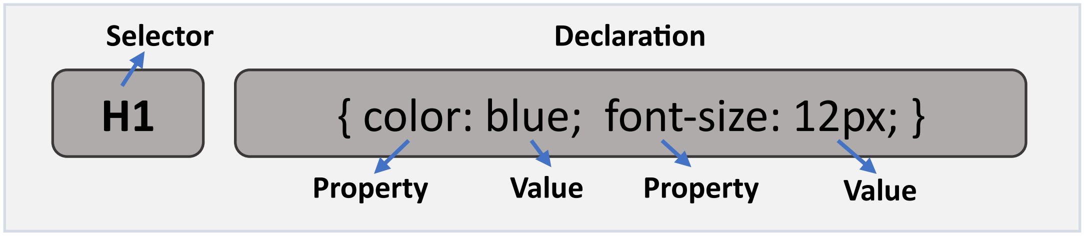
INTRODUCTION TO CSS

What is CSS?

CSS (Cascading Style Sheets) is a language used to define the visual appearance and layout of HTML documents. It allows developers to control design elements such as colors, fonts, margins, spacing, and positioning. CSS enables responsive web design, ensuring that websites look good on different screen sizes and devices. It works by applying styles to HTML elements through selectors and properties, either inline, internally, or externally via separate stylesheets. By separating content from design, CSS improves website maintainability and enhances user experience.

CSS Syntax

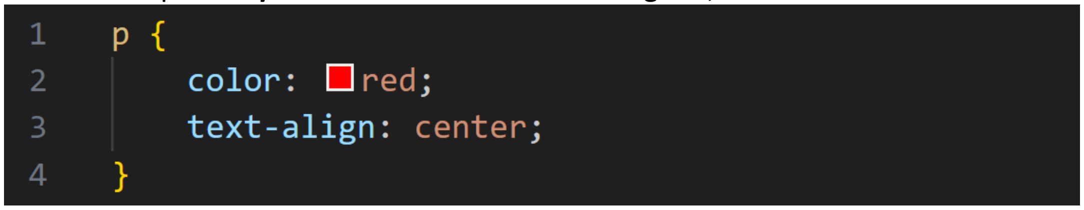
A CSS rule consists of a selector and a declaration block.



- The **selector** points to the HTML element you want to style.
- The **declaration** block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS **property name** and a **value**, separated by a colon.
- Multiple CSS declarations are separated with **semicolons**, and declaration blocks are surrounded by **curly braces**.

Example:

In this example all ***<p>*** elements will be center-aligned, with a red text color:



```
p {
    color: red;
    text-align: center;
}
```

- ***p*** is a selector in CSS (it points to the HTML element you want to style: ***<p>***).
- ***color*** is a property, and ***red*** is the property value
- ***text-align*** is a property, and ***center*** is the property value

CSS Selectors

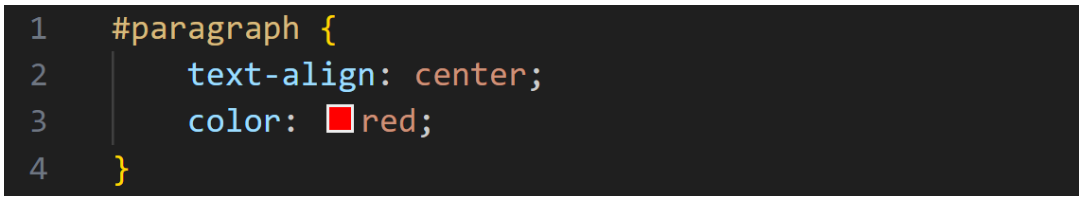
CSS selectors are used to "find" (or select) the HTML elements you want to style.
We can divide CSS selectors into five categories:

- **Simple selectors** (select elements based on name, id, class)

- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.

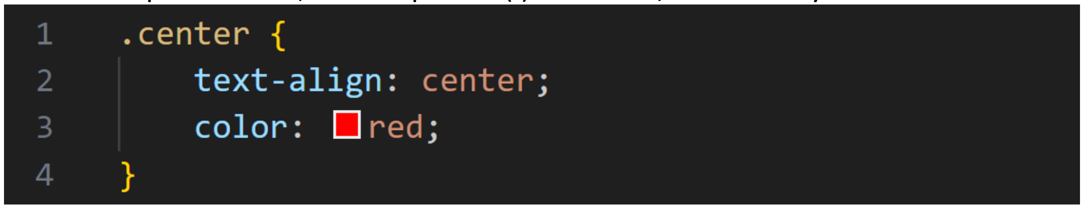


```css
#paragraph {
    text-align: center;
    color: red;
}
```

Note: *An id name cannot start with a number!*

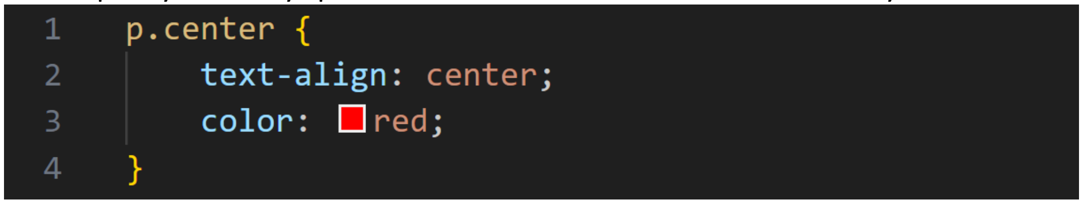
The CSS class Selector

The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.



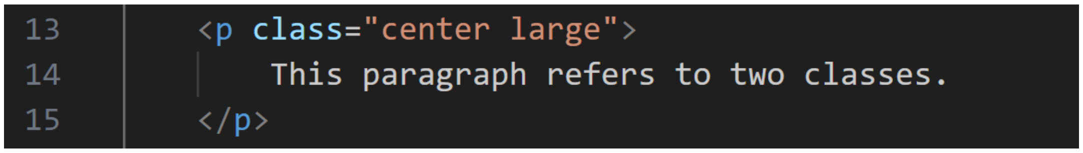
```css
.center {
    text-align: center;
    color: red;
}
```

You can also specify that only specific HTML elements should be affected by a class.



```css
p.center {
    text-align: center;
    color: red;
}
```

HTML elements can also refer to more than one class.

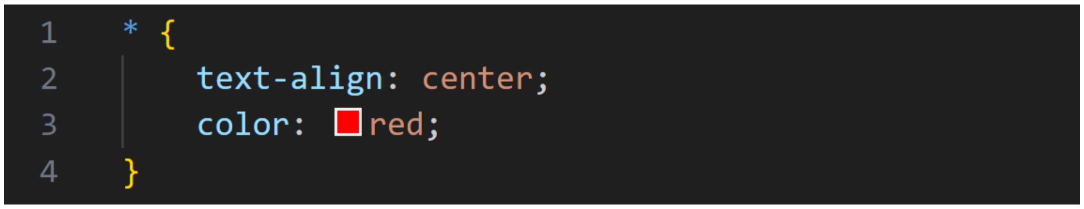


```html
<p class="center large">
    This paragraph refers to two classes.
</p>
```

Note: *A class name cannot start with a number!*

The CSS Universal Selector

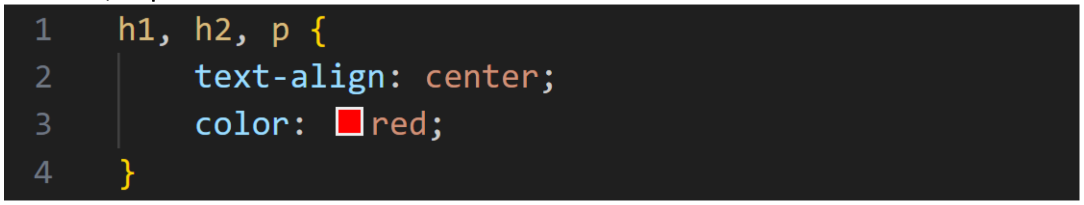
The universal selector (*) selects all HTML elements on the page.



```css
* {
    text-align: center;
    color: red;
}
```

The CSS Grouping Selector

To group selectors, separate each selector with a comma.



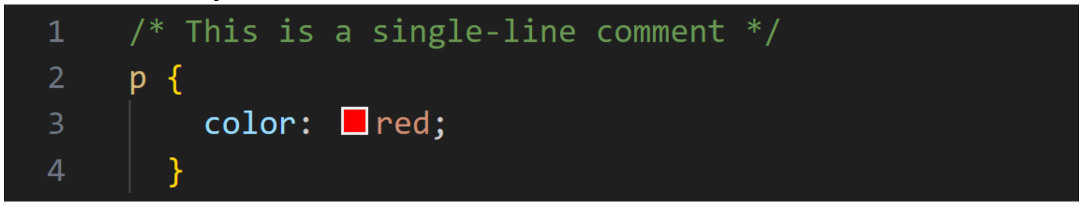
```css
h1, h2, p {
    text-align: center;
    color: red;
}
```

All CSS Simple Selectors

SELECTOR	EXAMPLE	EXAMPLE DESCRIPTION
#id	#firstname	Selects the element with id="firstname"
.class	.intro	Selects all elements with class="intro"
*	*	Selects all elements
element	p	Selects all <p> elements
element, element, element,…	div, p	Selects all <div> elements and all <p> elements

CSS Comments

CSS comments are not displayed in the browser, but they can help document your source code. Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers. It is placed inside the ***<style>*** element, and starts with ***/**** and ends with ****/.***



```css
/* This is a single-line comment */
p {
    color: red;
  }
```

CSS Box Model

The CSS Box Model is a fundamental concept in web design and development, crucial for understanding how elements are rendered on a webpage. It consists of four main components: Content, Padding, Border, and Margin.

Representation of Content, Padding, Border and Margin

```
-------------------------------------
|             Margin                |
|   -----------------------------   |
|   |       Border              |   |
|   |   ---------------------   |   |
|   |   |    Padding        |   |   |
|   |   |   -------------   |   |   |
|   |   |   |  Content  |   |   |   |
|   |   |   -------------   |   |   |
|   |   ---------------------   |   |
|   -----------------------------   |
-------------------------------------
```

Reference: https://programwithjayanth.com/notes/css/css-box-model-lession-4/

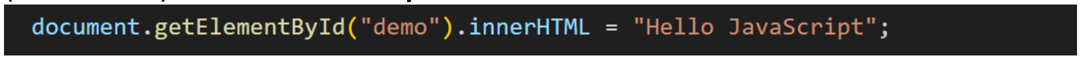
MODULE 4

INTRODUCTION TO JAVASCRIPT

What is JavaScript?

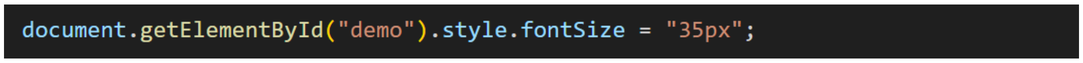
JavaScript is a versatile, high-level scripting language primarily used to create interactive and dynamic web pages. It enables developers to add functionalities such as animations, form validation, and real-time updates. JavaScript can be executed in web browsers, making it essential for front-end development, but it is also widely used on the server side with environments like Node.js. It supports event-driven, functional, and object-oriented programming paradigms. As one of the core technologies of web development, alongside HTML and CSS, JavaScript enhances user experience by enabling interactive and responsive web applications.

- **JavaScript Can Change HTML Content**
    - One of many JavaScript HTML methods is ***getElementById()***. The example below "finds" an HTML element (with ***id="demo"***), and changes the element content (***innerHTML***) to ***"Hello JavaScript"***.



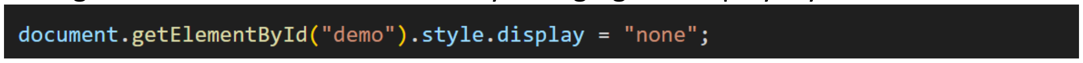
```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

- **JavaScript Can Change HTML Styles (CSS)**
    - Changing the style of an HTML element, is a variant of changing an HTML attribute.



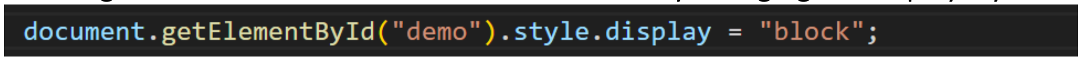
```
document.getElementById("demo").style.fontSize = "35px";
```

- **JavaScript Can Hide HTML Elements**
    - Hiding HTML elements can be done by changing the display style.



```
document.getElementById("demo").style.display = "none";
```

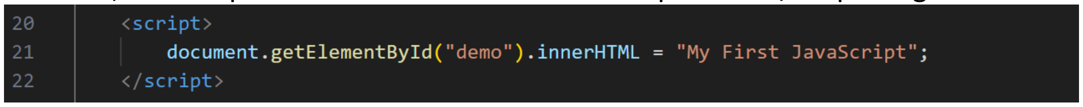
- **JavaScript Can Show HTML Elements**
    - Showing hidden HTML elements can also be done by changing the display style.



```
document.getElementById("demo").style.display = "block";
```

The <script> Tag

In HTML, JavaScript code is inserted between <script> and </script> tags.



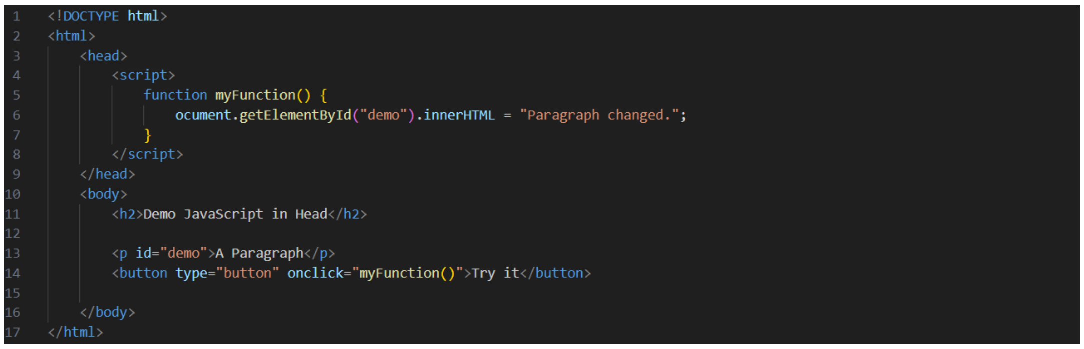
```
<script>
    document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

JavaScript in <head> or <body>

You can place any number of scripts in an HTML document. Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

JavaScript in <head>

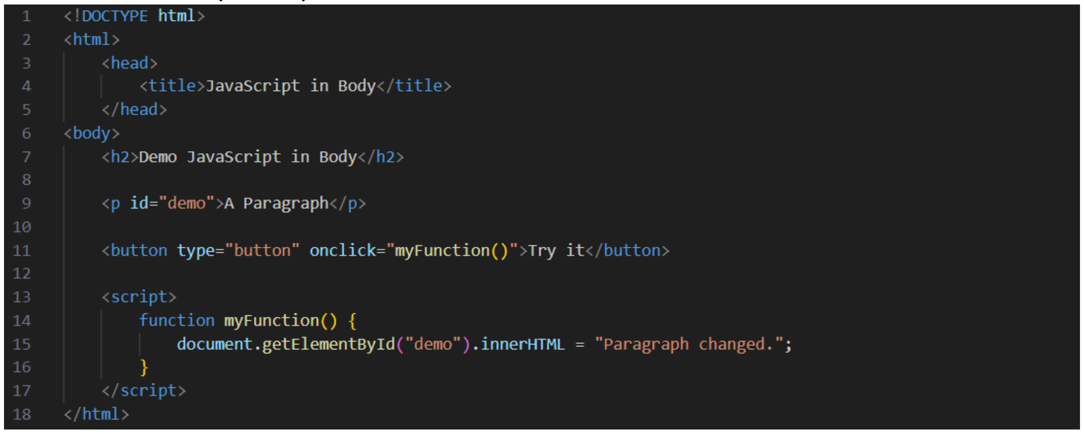
In this example, a JavaScript function is placed in the <head> section of an HTML page. The function is invoked (called) when a button is clicked.



```
<!DOCTYPE html>
<html>
    <head>
        <script>
            function myFunction() {
                ocument.getElementById("demo").innerHTML = "Paragraph changed.";
            }
        </script>
    </head>
    <body>
        <h2>Demo JavaScript in Head</h2>

        <p id="demo">A Paragraph</p>
        <button type="button" onclick="myFunction()">Try it</button>

    </body>
</html>
```

JavaScript in <body>

In this example, a JavaScript function is placed in the <body> section of an HTML page. The function is invoked (called) when a button is clicked.

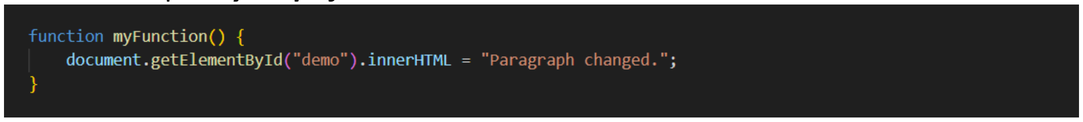


```
<!DOCTYPE html>
<html>
    <head>
        <title>JavaScript in Body</title>
    </head>
<body>
    <h2>Demo JavaScript in Body</h2>

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

    <button type="button" onclick="myFunction()">Try it</button>

    <script>
        function myFunction() {
            document.getElementById("demo").innerHTML = "Paragraph changed.";
        }
    </script>
</html>
```

External JavaScript

Scripts can also be placed in external files.

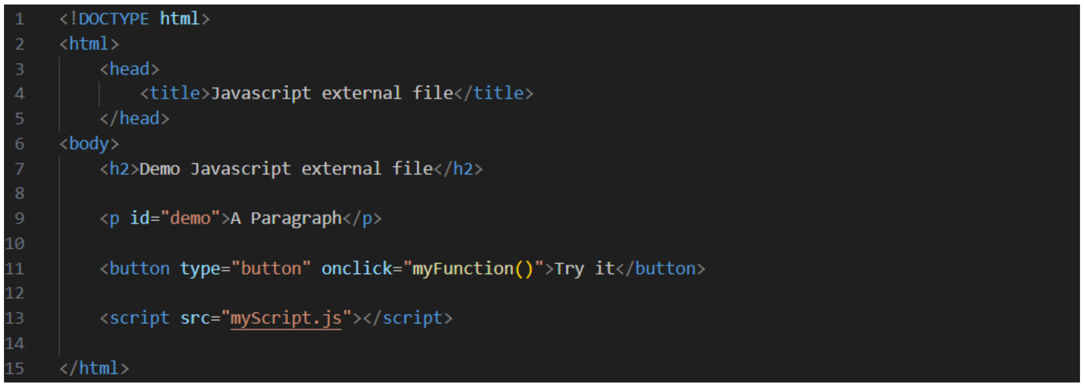
External Script: ***myScript.js***



```
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

- External scripts are practical when the same code is used in many different web pages.
- JavaScript files have the file extension ***.js***.

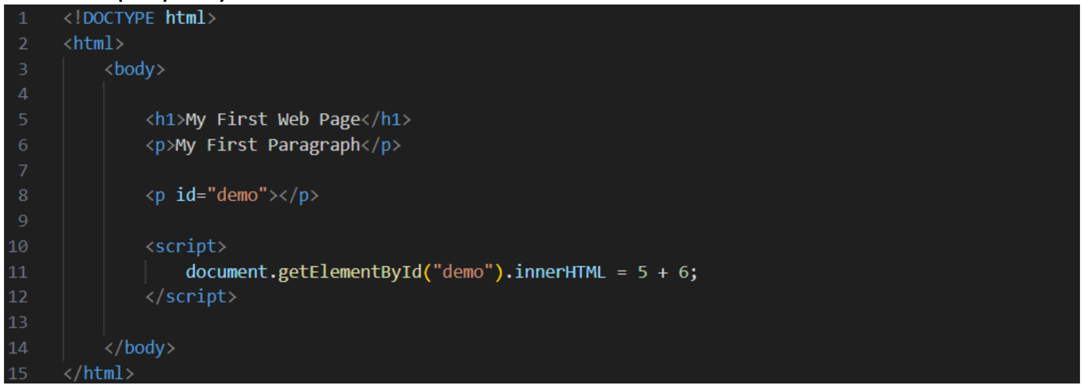
To use an external script, put the name of the script file in the ***src*** (source) attribute of a ***<script>*** tag. You can place an external script reference in ***<head>*** or ***<body>*** as you like. The script will behave as if it was located exactly where the <script> tag is located.



```html
<!DOCTYPE html>
<html>
    <head>
        <title>Javascript external file</title>
    </head>
<body>
    <h2>Demo Javascript external file</h2>

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

    <button type="button" onclick="myFunction()">Try it</button>

    <script src="myScript.js"></script>

</html>
```

Display Ssing *innerHTML*

To access an HTML element, JavaScript can use the ***document.getElementById(id)*** method. The ***id attribute*** defines the HTML element. The ***innerHTML*** property defines the HTML content.



```html
<!DOCTYPE html>
<html>
    <body>

        <h1>My First Web Page</h1>
        <p>My First Paragraph</p>

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

        <script>
            document.getElementById("demo").innerHTML = 5 + 6;
        </script>

    </body>
</html>
```

Display Using *document.write()*