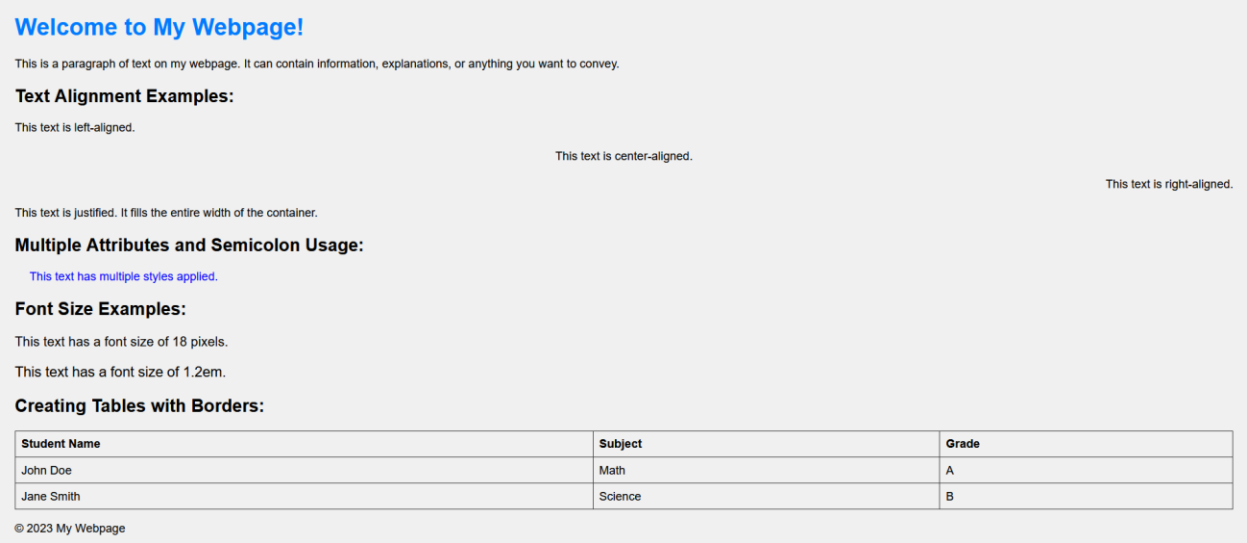
HTML INTRODUCTION

Github Repository : https://github.com/KaunainRizvi/Learning-In-Public-Kaunain_Rizvi

Linkedin : https://www.linkedin.com/in/kaunain-rizvi/

WITH THIS NOTES IN THE END WE WILL BE ALBE TO CREATE WEBPAGES LIKE THIS :



1. HTML Full Form and its Origin: Unveiling the Web's Building Blocks

Hey there! Let's dive into the ABCs of the internet - HTML, which stands for HyperText Markup Language. Sounds fancy, right? But don't worry, it's not as complicated as it sounds.

So, back in the early days of the World Wide Web, a smart guy named Tim Berners-Lee thought, "Hey, we need a way to share information on this internet thing." That's where HTML comes into play. Tim, in the early 1990s, cooked up this simple markup language. Now, what's a markup language? Think of it as a set of instructions that tells your web browser how to display things.

So, why did Tim bother creating HTML? Well, imagine a world where every website looks different, where chaos reigns supreme. Tim wanted order and consistency. He wanted a way for people to share information on the web in a standardized format.

That's why HTML was born. It's the backbone of almost every webpage you visit. When you open a webpage, your browser uses HTML to understand how to structure and display the content - text, images, links, you name it.

In a nutshell, HTML is like the architect's blueprint for a website. It tells your browser where to put each piece of content so that when you open a webpage, it looks just like it's supposed to.

So, the next time you see a webpage, remember, behind the scenes, HTML is quietly doing its job, making sure everything is in its right place. It's like the unsung hero of the internet!

2. HTML: The Language of Web Content Presentation

HTML, which stands for HyperText Markup Language, is not exactly a programming language in the traditional sense. Instead, it's a markup language. Let me break it down for you:

1. What's in a Name: HyperText Markup Language (HTML)

- HTML is all about organizing and presenting content on the web. It's not a programming language like Python or Java, but a markup language designed for creating the structure of a webpage.

2. The Blueprint: Markup vs. Programming

- Unlike programming languages that give instructions for tasks, HTML provides a blueprint for web browsers. It tells the browser how to structure content, like headings, paragraphs, images, and links, through a system of tags.

3. Tags: The Building Blocks of HTML

- Tags are the stars of the show in HTML. They are elements enclosed in angle brackets (< >) that define the structure of your content. For example, `<p>` tags indicate a paragraph, and `<h1>` tags define a top-level heading.

4. Elements, Attributes, and Values: Oh My!

- Elements are made up of tags, and they can have attributes and values. Attributes provide additional information about an element, like setting the source of an image. Values are assigned to attributes to specify their properties.

5. Structure Matters: HTML Document Skeleton

- Every HTML document has a basic structure, starting with `<!DOCTYPE html>` to define the document type. The `<html>` tag contains the entire document, and within it, you'll find `<head>` for metadata and `<body>` for the actual content.

6. Nesting: Tags Inside Tags

- HTML allows nesting of tags, meaning you can put tags inside other tags. This hierarchy helps create a well-organized structure for your content.

7. Hyperlinks and Navigation: Connecting the Dots

- HTML facilitates linking between pages using `<a>` (anchor) tags. These tags create hyperlinks, allowing users to navigate seamlessly through different web pages.

8. Multimedia Magic: Embedding Images and Videos

- You can enhance your webpage with multimedia content. The `<img>` tag is used for images, while `<video>` and `<audio>` tags handle video and audio content.

9. Forms and User Interaction: Collecting Data

- HTML includes elements like `<form>`, `<input>`, and `<button>` to create interactive forms, enabling users to input data or make selections on a webpage.

10. Evolution and Compatibility: HTML Versions

- HTML has evolved over the years. HTML5 is the latest version, introducing new features and improvements. Web browsers are designed to interpret and display HTML code, ensuring compatibility across different platforms.

In a nutshell, while HTML isn't a programming language, it's the fundamental language of the web, shaping the way content is structured and presented across the internet. Understanding HTML is like having the key to unlock the door to the vast world of web development.

3. HTML vs. CSS vs. JavaScript: Unraveling the Trio of Web Development

In the realm of web development, HTML, CSS, and JavaScript are the dynamic trio, each with its own unique role. Let's explore the differences with real-life examples and use cases:

---

1. HTML: The Architect of Web Content

Definition: HTML, or HyperText Markup Language, is the backbone of web content. It structures and organizes information on a webpage using a system of tags.

Real Life Analogy: Imagine a book. HTML is like the table of contents—it outlines the structure of the content but doesn't dictate how it looks or the colors used.

Example:

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My Webpage</title>
  </head>
  <body>
    <h1>Welcome to My Webpage</h1>
    <p>This is a paragraph of text.</p>
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
    </ul>
  </body>
</html>
```

Real Life Use: Creating the basic structure of a webpage - headings, paragraphs, lists, and links.

---

2. CSS: Styling the Web

Definition: CSS, or Cascading Style Sheets, is all about presentation. It adds style, layout, and visual appeal to HTML elements.

Real Life Analogy: If HTML is the skeleton, CSS is the clothing—it decides how the content looks, from fonts and colors to spacing and positioning.

Example:

```css
body {
  font-family: 'Arial', sans-serif;
  background-color: #f0f0f0;
}

h1 {
  color: #333;
}

p {
  margin-bottom: 20px;
}
```

Real Life Use: Making your webpage visually appealing by setting fonts, colors, spacing, and positioning elements.

---

3. JavaScript: Adding Interactivity

Definition: JavaScript is a scripting language that adds interactivity to web pages. It allows you to manipulate content, respond to user actions, and update the page dynamically.

Real Life Analogy: If HTML is the static content and CSS is the appearance, JavaScript is the magic that happens when you interact with a page—it makes things move, change, and respond.

Example:

```javascript
function greetUser() {
  var userName = prompt('What is your name?');
  alert('Hello, ' + userName + '!');
}
```

Real Life Use: Creating interactive elements like forms, buttons, and dynamic content that responds to user input.

Summing it Up: The Trio in Harmony

- **HTML** provides the structure.
- **CSS** adds the style and design.
- **JavaScript** brings it to life with interactivity.

In a nutshell, HTML, CSS, and JavaScript work together to create the dynamic and visually appealing web pages we interact with every day. They're the building blocks of the internet, collaborating seamlessly to deliver a rich user experience.

4. Creating a Website: The Impact of HTML, CSS, and JavaScript

1. HTML-Only Website: Building the Foundation

HTML's Role: HTML alone structures content on a webpage. It's like constructing a basic framework without worrying about the appearance.

Example:

```
htmlCopy code
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>HTML-Only Website</title>
</head>
<body>
<header>
<h1>Welcome to My Website</h1>
</header>
<section>
<p>This is a simple HTML-only website.</p>
</section>
<footer>
<p>&copy; 2023 My Website</p>
```

```
</footer>
</body>
</html>
```

Result: A straightforward webpage with content but lacking visual appeal.

2. HTML + CSS: Adding Style and Flair

CSS's Role: CSS steps in to style the webpage. It's like giving your HTML structure a makeover—choosing colors, fonts, and layout.

Example:

```
cssCopy code
body {
font-family: 'Arial', sans-serif;
background-color: #f0f0f0;
color: #333;
}

header {
text-align: center;
padding: 20px;
background-color: #007BFF;
color: #fff;
}

section {
max-width: 600px;
margin: 0 auto;
}

footer {
text-align: center;
padding: 10px;
background-color: #007BFF;
color: #fff;
}
```

Result: The webpage now looks visually appealing with specified fonts, colors, and layout.

3. HTML + CSS + JavaScript: Adding Interactivity

JavaScript's Role: JavaScript brings life to your website. It allows you to create interactive elements and respond to user actions.

Example:

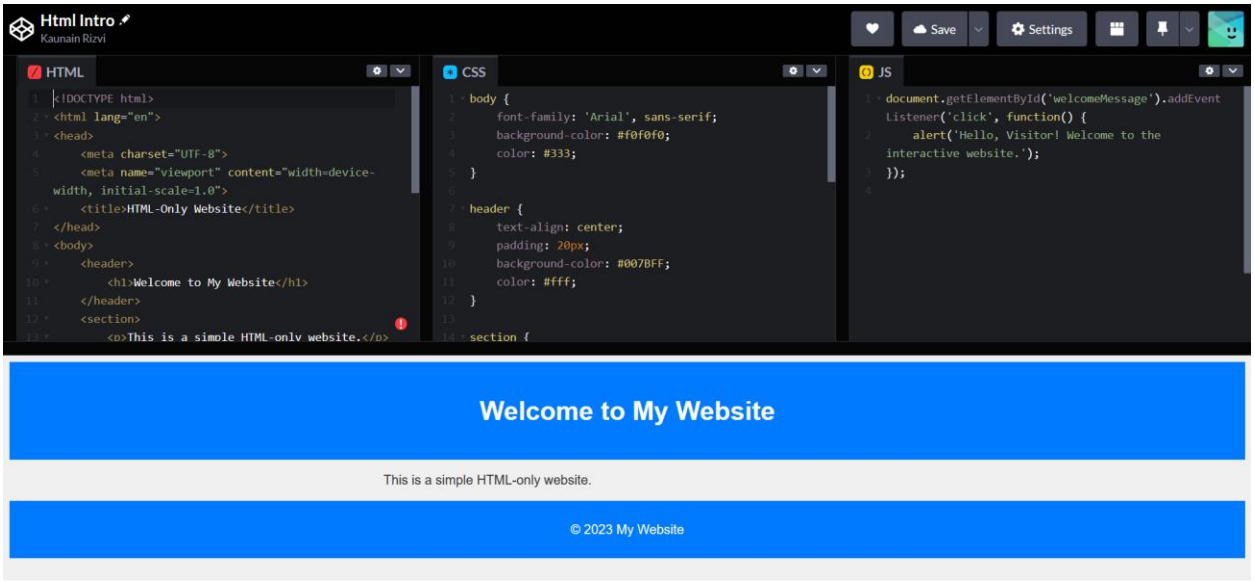
```
javascriptCopy code
document.getElementById('welcomeMessage').addEventListe
ner('click', function() {
alert('Hello, Visitor! Welcome to the interactive
website.');
});
```

Result: By clicking on an element, like a welcome message, the user now triggers an alert, adding an interactive element to the webpage.

Summing it Up: The Evolution of a Website

- **HTML:** Provides structure and content.
- **CSS:** Adds style, making it visually appealing.
- **JavaScript:** Introduces interactivity, making the website dynamic.

With HTML alone, you have a basic webpage. Adding CSS enhances its aesthetics, and integrating JavaScript takes it a step further by making it interactive and engaging. Together, they form a powerful trio that shapes the modern web experience.



You can experiment it : https://codepen.io/KaunainRizvi/pen/GRzmYvy

---

HTML became popular for building web pages because of its simplicity, versatility, and its role in the foundational architecture of the World Wide Web. While tools like Microsoft Word or other document processing software have similarities in terms of handling text and formatting, they serve different purposes. Let's break down why HTML gained fame in the context of the web:

1. **Designed for the Web:**
    - HTML was specifically designed for creating content on the internet. Its structure allows the easy representation of hypertext and hyperlinks, crucial for navigating between web pages. Microsoft Word, on the other hand, is a general-purpose word processing tool, not tailored for the interconnected nature of the web.
2. **Markup Language vs. Document Processing:**
    - HTML is a markup language, not a document processing tool. It provides a way to structure content and define relationships between different parts of a document. This makes it well-suited for creating linked documents on the web. In contrast, document processing tools

like Microsoft Word focus on creating standalone documents rather than interconnected web content.

3. **Open Standards and Accessibility:**
    - HTML is based on open standards, which means that anyone can implement it without restrictions. This openness contributed to the widespread adoption of HTML across different platforms and browsers, fostering a more accessible and inclusive web. In contrast, proprietary document formats may have compatibility issues and are often tied to specific software.
4. **Evolution and Adaptability:**
    - HTML has evolved over time to accommodate the changing needs of the web. The introduction of HTML5 brought new features and capabilities, making it suitable for a wide range of multimedia content and interactive applications. This adaptability has kept HTML at the forefront of web development.
5. **Community and Collaboration:**
    - The web development community has played a crucial role in the popularity of HTML. It is supported by a large and collaborative community that contributes to its development, shares best practices, and creates resources for learning. This collective effort has made HTML accessible to developers worldwide.
6. **Cross-Platform Compatibility:**
    - HTML is designed to work across different platforms and devices seamlessly. It ensures a consistent user experience regardless of the browser or operating system. This cross-platform compatibility has been a key factor in HTML's success.

In summary, HTML gained popularity because it was purpose-built for the web, open and accessible, adaptable to changing technologies, supported by a vibrant community, and ensured cross-platform compatibility. While tools like Microsoft Word serve their own valuable purposes, they are not a replacement for the specific needs of creating content for the interconnected and dynamic environment of the World Wide Web.

---

4. **HTML is used to collect data from users through forms. Let's go through a simple example to illustrate how HTML is employed for data collection:**

HTML Form for User Data Collection

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>User Data Collection</title>
</head>
<body>

<h1>Enter Your Information</h1>

<!-- HTML Form -->
<form action="/submit" method="post">
<!-- Text Input for Name -->
<label for="name">Name:</label>
<input type="text" id="name" name="name" required>

<br>

<!-- Email Input -->
<label for="email">Email:</label>
<input type="email" id="email" name="email" required>

<br>

<!-- Radio Buttons for Gender -->
<label>Gender:</label>
<input type="radio" id="male" name="gender" value="male" checked>
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label>

<br>

<!-- Checkbox for Subscription -->
<input type="checkbox" id="subscribe" name="subscribe" value="yes">
<label for="subscribe">Subscribe to our newsletter</label>
```

```
    <br>

    <!-- Dropdown for Country Selection -->
    <label for="country">Country:</label>
    <select id="country" name="country">
    <option value="usa">USA</option>
    <option value="canada">Canada</option>
    <option value="uk">UK</option>
    <!-- Add more options as needed -->
    </select>

    <br>

    <!-- Textarea for Additional Comments -->
    <label for="comments">Additional Comments:</label>
    <textarea id="comments" name="comments" rows="4"></textarea>

    <br>

    <!-- Submit Button -->
    <input type="submit" value="Submit">
    </form>

    </body>

</html>
```

In this example:

- The HTML form is created using the `<form>` element.
- Various input elements such as text input, email input, radio buttons, checkboxes, dropdowns, and textarea are used to collect different types of data.
- The `for` attribute in `<label>` elements is associated with the `id` of the corresponding input, improving accessibility.
- The `required` attribute ensures that certain fields must be filled in before submitting the form.
- The `action` attribute in the `<form>` element specifies where the form data should be sent.

When a user fills out this form and clicks the "Submit" button, the data entered by the user is sent to the specified action (in this case, `/submit`). The server-side code would then handle this data, and further actions can be taken, such as storing it in a database or sending it via email.

You can experiment it : https://codepen.io/KaunainRizvi/pen/qBgmJxq

---

HTML was not primarily created by scientists for document and research sharing; its origins lie in a different context. HTML, which stands for HyperText Markup Language, was developed by Tim Berners-Lee, a British computer scientist, in the early 1990s. Tim Berners-Lee invented HTML as part of his work at CERN (the European Organization for Nuclear Research), but its initial purpose was not specifically for scientific document sharing.

Tim Berners-Lee's primary goal was to facilitate communication and information-sharing among researchers at CERN. He envisioned a system that would allow scientists to easily navigate and access documents related to their research. This vision led to the creation of the World Wide Web.

HTML was designed as a markup language to structure and organize content on the web, making it easier to share and link documents. While the web did have

scientific use cases, the adoption of HTML expanded well beyond the scientific community. Its simplicity, openness, and effectiveness for creating interconnected content contributed to its rapid growth and adoption across various domains.

Over time, as the internet evolved, HTML became the standard language for creating websites of all types, not just for scientific documentation. Today, HTML is a fundamental building block for the entire World Wide Web, serving as the backbone for web development across diverse fields, including business, education, entertainment, and more. The flexibility and adaptability of HTML have contributed to its widespread use in creating websites for a variety of purposes.

Below is a simple HTML boilerplate code for creating a basic webpage. I'll explain each part of the code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   <title>My First Webpage</title>
</head>
<body>

   <header>
      <h1>Welcome to My Webpage</h1>
   </header>

   <section>
      <p>This is a paragraph of text on my webpage. It can contain
information, explanations, or anything you want to convey.</p>
   </section>

   <footer>
      <p>&copy; 2023 My Webpage</p>
   </footer>

</body>
</html>
```

You can experiment it : https://codepen.io/KaunainRizvi/pen/GRzmYdx

---

1. `<html>`: Defines the root of an HTML document.
2. `<head>`: Contains metadata about the HTML document, such as title, character set, etc.
    - `<title>`: Sets the title of the HTML document.
3. `<body>`: Contains the main content of the HTML document.
    - `<p>`: Represents a paragraph of text.
    - `<hr>`: Represents a horizontal rule (line) for separating content.
    - `<br>`: Represents a line break within text.
    - `<ul>`: Represents an unordered list.
        - `<li>`: Represents a list item within an unordered list.
    - `<ol>`: Represents an ordered list.
        - `<li>`: Represents a list item within an ordered list.

So, when you put it all together, a simple HTML document might look like this:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My HTML Document</title>
</head>
```

```
<body>

    <h1>Welcome to My Webpage!</h1>

    <p>This is a paragraph of text in the main content.</p>

    <hr>

    <p>This is another paragraph, separated by a horizontal rule.</p>

    <ul>
        <li>Unordered List Item 1</li>
        <li>Unordered List Item 2</li>
        <li>Unordered List Item 3</li>
    </ul>

    <ol>
        <li>Ordered List Item 1</li>
        <li>Ordered List Item 2</li>
        <li>Ordered List Item 3</li>
    </ol>

    <p>Feel free to add more content!</p>
```

</body>

</html>

You can experiment it : https://codepen.io/KaunainRizvi/pen/BaMRqVO

---

12. Heading Tags:

HTML provides six levels of headings, ranging from `<h1>` (the highest) to `<h6>` (the lowest).

Example:

```
htmlCopy code
<h1>This is a Heading Level 1</h1>
<h2>This is a Heading Level 2</h2>
<!-- ... -->
```

```
<h6>This is a Heading Level 6</h6>
```

Headings are used to structure content hierarchically. Search engines and screen readers use these tags to understand the organization of your content.

13. Self-Closing Tags:

Some HTML tags don't have a closing tag and are self-closed. They end with a forward slash (`/`) before the closing angle bracket.

Example:

```
htmlCopy code
<img src="image.jpg" alt="An example image">
<br>
<input type="text" placeholder="Enter your name">
```

Self-closing tags are used for elements like images, line breaks, and input fields, where there is no content to be placed between opening and closing tags.

14. HTML Attributes:

Attributes provide additional information about HTML elements and are always included in the opening tag. They are made up of a name and a value.

Example:

```
htmlCopy code
<a href="https://www.example.com" target="_blank">Visit
Example.com</a>
<img src="logo.png" alt="Company Logo" width="100"
height="50">
```

In the examples, `href`, `target`, `src`, `alt`, `width`, and `height` are attributes. They provide extra details about the associated elements.

15. Anchor Tag (`<a>`), Href, Target, Input Tag (`<input>`), Placeholder, Type:

Anchor Tag (<a>):

- **Example:**

```
htmlCopy code
<a href="https://www.example.com"
target="_blank">Visit Example.com</a>
```

- **Explanation:**
  - `href`: Specifies the URL of the linked page.
  - `target="_blank"`: Opens the link in a new browser window or tab.

Input Tag (<input>):

- **Example:**

```
htmlCopy code
<input type="text" placeholder="Enter your name">
```

- **Explanation:**
  - `type="text"`: Specifies that the input field is for plain text.
  - `placeholder`: Provides a hint or example text for users.

16. Default Value of Target:

By default, if the `target` attribute is not specified, the link opens in the same window or tab.

Example:

```
htmlCopy code
<a href="https://www.example.com">Visit Example.com</a>
```

This link will open in the same window/tab by default.

Understanding these concepts and examples will give you a solid foundation for working with HTML and creating interactive and well-structured web pages.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML Example</title>
</head>
<body>

    <h1>This is a Heading Level 1</h1>
    <h2>This is a Heading Level 2</h2>

    <p>HTML provides a way to structure content on the web.</p>

    <a href="https://www.example.com" target="_blank">Visit Example.com</a>

    <img src="logo.png" alt="Company Logo" width="100" height="50">

    <br>

    <input type="text" placeholder="Enter your name">

</body>
```

</html>

You can experiment it : https://codepen.io/KaunainRizvi/pen/RwvVeeQ

18. Nested List (Gehri List):

HTML allows you to create nested lists, where one list is contained within another. Here's an example:

```
htmlCopy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Nested List Example</title>
</head>
<body>

    <h2>Programming Languages</h2>

    <ul>
        <li>Frontend
            <ul>
                <li>HTML</li>
                <li>CSS</li>
                <li>JavaScript</li>
```

```
            </ul>
        </li>
        <li>Backend
            <ul>
                <li>Python</li>
                <li>Java</li>
                <li>Node.js</li>
            </ul>
        </li>
    </ul>

</body>
</html>
```

In this example:

- `<ul>` is an unordered list (bullet points).
- `<li>` is a list item.
- Inside the list items of the outer list, there are nested `<ul>` elements to create sublists.

19. Image Tag (`<img>`) Alt Attribute:

The `alt` attribute in the `<img>` tag is used to provide alternative text for an image. This text is displayed if the image cannot be loaded and is also used by screen readers for accessibility.

Example:

```
htmlCopy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Image Alt Attribute Example</title>
</head>
<body>

    <h2>Company Logo</h2>
```

```
    <img src="logo.png" alt="Company Logo">

</body>
</html>
```

Benefits of Using the `alt` Attribute:

1. **Accessibility:** Screen readers use the `alt` text to describe images to users with visual impairments.
2. **SEO (Search Engine Optimization):** Search engines use the `alt` text to understand and index images, improving the overall SEO of the page.
3. **User Experience:** If an image fails to load, the `alt` text provides users with information about the content or purpose of the image.

Including descriptive and relevant `alt` text is considered a best practice for web development, contributing to a more inclusive and user-friendly web experience.

Let's create a comprehensive example using various HTML tags and attributes:

```
htmlCopy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>HTML Tags and Attributes Example</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f0f0f0;
            margin: 20px;
        }
        h1 {
            color: #007BFF;
        }
```

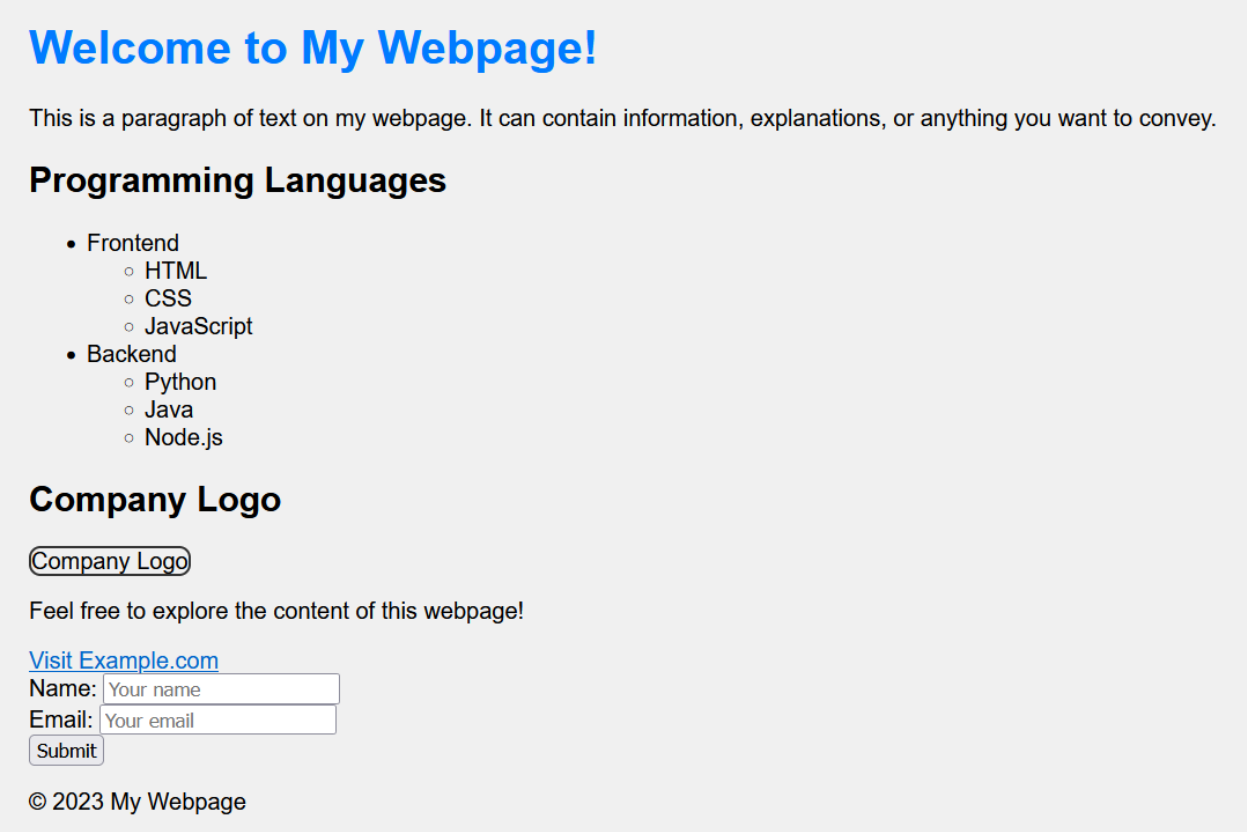
```
        img {
            border: 2px solid #333;
            border-radius: 8px;
        }
    </style>
</head>
<body>

    <h1>Welcome to My Webpage!</h1>

    <p>This is a paragraph of text on my webpage. It can contain information, explanations, or anything you want to convey.</p>

    <h2>Programming Languages</h2>

    <ul>
        <li>Frontend
            <ul>
                <li>HTML</li>
                <li>CSS</li>
                <li>JavaScript</li>
            </ul>
        </li>
        <li>Backend
            <ul>
                <li>Python</li>
                <li>Java</li>
                <li>Node.js</li>
            </ul>
        </li>
    </ul>

    <h2>Company Logo</h2>

    <img src="logo.png" alt="Company Logo" width="150" height="80">

    <p>Feel free to explore the content of this webpage!</p>
```

```
    <a href="https://www.example.com"
target="_blank">Visit Example.com</a>

    <form action="/submit" method="post">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"
placeholder="Your name" required>

        <br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email"
placeholder="Your email" required>

        <br>

        <input type="submit" value="Submit">
    </form>

    <footer>
        <p>&copy; 2023 My Webpage</p>
    </footer>

</body>
</html>
```

In this example:

- Styles are added within the `<style>` tag in the head for basic page styling.
- The `<form>` tag is used to create a simple form with text input fields and a submit button.
- Various attributes like `id`, `name`, `placeholder`, `width`, and `height` are used within the tags.
- The `target="_blank"` attribute is used in the anchor tag to open the link in a new tab or window.

This example combines different HTML tags and attributes to create a more complex webpage. Feel free to modify and experiment with this code to understand how these elements work together.



You can experiment it : https://codepen.io/KaunainRizvi/pen/dyaWgaV

Css(cascading styling sheet) Introductions:

20. CSS Introduction:

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation of a document written in HTML or XML. It allows you to control the layout, style, and appearance of multiple elements on a webpage. CSS works by selecting HTML elements and applying various styles to them, such as colors, fonts, spacing, and positioning.

Here's a basic example of how CSS looks:

```
cssCopy code
/* This is a CSS comment */
```

```
body {
    font-family: 'Arial', sans-serif;
    background-color: #f0f0f0;
    color: #333;
}

h1 {
    color: #007BFF;
}

p {
    margin-bottom: 20px;
}
```

In this example:

- `body`, `h1`, and `p` are selectors targeting HTML elements.
- `font-family`, `background-color`, `color`, and `margin-bottom` are properties.
- `'Arial', sans-serif` and `#f0f0f0` are property values.

21. CSS Use and Benefits:

Using CSS provides several benefits:

- **Separation of Concerns:** CSS allows separation of content (HTML) and presentation (CSS), making code maintenance and updates more manageable.
- **Consistency:** Apply consistent styles across multiple pages by linking to a single CSS file.
- **Responsive Design:** Adjust layouts for different screen sizes, enhancing the user experience on various devices.
- **Flexibility:** Easily modify the appearance of a website without changing the HTML structure.
- **Readability:** Improve code readability with well-organized styles.

22. Adding Color with CSS Using Different Methods:

a. Inline Styles:

Inline styles are applied directly to individual HTML elements using the `style` attribute.

Example:

```
htmlCopy code
<p style="color: red;">This text is red.</p>
```

b. Using Class Selector:

Define a CSS class and apply it to HTML elements.

CSS:

```
cssCopy code
.red-text {
    color: red;
}
```

HTML:

```
htmlCopy code
<p class="red-text">This text is red.</p>
```

c. Using ID Selector:

Similar to classes, but IDs are unique within a page.

CSS:

```
cssCopy code
#green-text {
    color: green;
}
```

HTML:

```
htmlCopy code
<p id="green-text">This text is green.</p>
```

d. Using Color Names:

Some basic color names can be used directly.

CSS:

```
cssCopy code
.blue-text {
    color: blue;
}
```

HTML:

```
htmlCopy code
<p class="blue-text">This text is blue.</p>
```

e. Using Color Codes:

Specify colors using hexadecimal or RGB color codes.

CSS:

```
cssCopy code
#ff6600 {
    color: #ff6600; /* Hexadecimal */
}

.rgb-color {
    color: rgb(255, 0, 0); /* RGB */
}
```

HTML:

```
htmlCopy code
<p id="ff6600">This text has a custom color.</p>
<p class="rgb-color">This text uses an RGB color.</p>
```

Choose the method that best suits your needs and the structure of your HTML. Generally, external or internal stylesheets are preferred over inline styles for better organization and maintainability.

23. Text Align in Inline Styles:

a. Left Align:

```
htmlCopy code
<p style="text-align: left;">This text is left-
aligned.</p>
```

b. Center Align:

```
htmlCopy code
<p style="text-align: center;">This text is center-
aligned.</p>
```

c. Right Align:

```
htmlCopy code
<p style="text-align: right;">This text is right-
aligned.</p>
```

d. Justify:

```
htmlCopy code
<p style="text-align: justify;">This text is justified.
It fills the entire width of the container.</p>
```

24. Semicolon Usage in HTML and Multiple Attributes:

In HTML, semicolons are primarily used in inline styles. Multiple attributes can be added to a single inline style by separating them with semicolons.

Example:

```
htmlCopy code
<p style="color: blue; font-size: 16px; margin-left:
20px;">This text has multiple styles applied.</p>
```

Here, `color: blue`, `font-size: 16px`, and `margin-left: 20px` are three different style attributes separated by semicolons.

25. Font Size in HTML:

a. Using Pixels:

```
htmlCopy code
<p style="font-size: 18px;">This text has a font size
of 18 pixels.</p>
```

b. Using em Units:

```
htmlCopy code
<p style="font-size: 1.2em;">This text has a font size
of 1.2em.</p>
```

26. Creating Tables in HTML with Borders:

```
htmlCopy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>HTML Table Example</title>
    <style>
        table {
            border-collapse: collapse;
            width: 100%;
        }
        th, td {
            border: 1px solid #333;
            padding: 8px;
            text-align: left;
        }
    </style>
</head>
<body>

    <h2>Student Grades</h2>

    <table>
        <tr>
            <th>Student Name</th>
            <th>Subject</th>
            <th>Grade</th>
        </tr>
        <tr>
            <td>John Doe</td>
            <td>Math</td>
```

```
            <td>A</td>
        </tr>
        <tr>
            <td>Jane Smith</td>
            <td>Science</td>
            <td>B</td>
        </tr>
    </table>

</body>
</html>
```

In this example:

- The `<table>` element defines the table structure.
- `border-collapse: collapse;` ensures that borders are collapsed into a single border.
- `border: 1px solid #333;` provides a border for table cells.
- `padding: 8px;` adds spacing inside cells.
- `text-align: left;` aligns text to the left within cells.

This creates a table with borders around each cell. Adjust styles according to your design preferences.

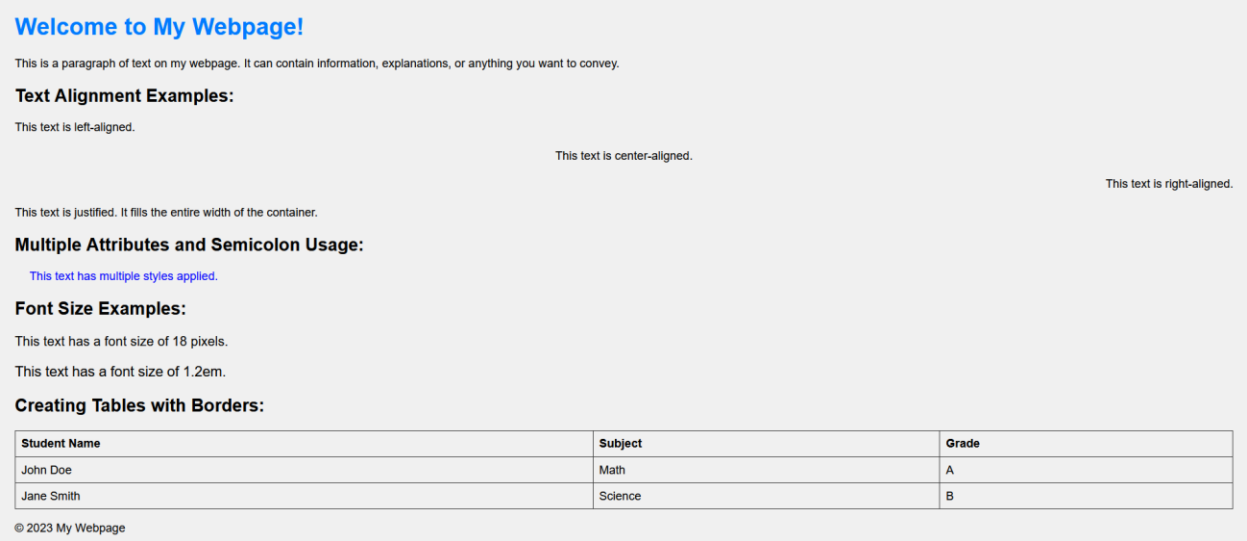
```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Complete HTML Example</title>

   <style>

      body {
```

```
    font-family: 'Arial', sans-serif;
    background-color: #f0f0f0;
    margin: 20px;
}
h1 {
    color: #007BFF;
}
p {
    margin-bottom: 20px;
}
.blue-text {
    color: blue;
}
table {
    border-collapse: collapse;
    width: 100%;
}
th, td {
    border: 1px solid #333;
    padding: 8px;
    text-align: left;
```

```
        }
    </style>
</head>
<body>

    <h1>Welcome to My Webpage!</h1>

    <p>This is a paragraph of text on my webpage. It can contain information, explanations, or anything you want to convey.</p>

    <h2>Text Alignment Examples:</h2>

    <p style="text-align: left;">This text is left-aligned.</p>
    <p style="text-align: center;">This text is center-aligned.</p>
    <p style="text-align: right;">This text is right-aligned.</p>
    <p style="text-align: justify;">This text is justified. It fills the entire width of the container.</p>

    <h2>Multiple Attributes and Semicolon Usage:</h2>

    <p style="color: blue; font-size: 16px; margin-left: 20px;">This text has multiple styles applied.</p>
```

```
<h2>Font Size Examples:</h2>

<p style="font-size: 18px;">This text has a font size of 18 pixels.</p>
<p style="font-size: 1.2em;">This text has a font size of 1.2em.</p>

<h2>Creating Tables with Borders:</h2>

<table>
    <tr>
        <th>Student Name</th>
        <th>Subject</th>
        <th>Grade</th>
    </tr>
    <tr>
        <td>John Doe</td>
        <td>Math</td>
        <td>A</td>
    </tr>
    <tr>
        <td>Jane Smith</td>
        <td>Science</td>
```

```
            <td>B</td>
        </tr>
    </table>

    <footer>
        <p>&copy; 2023 My Webpage</p>
    </footer>

</body>
</html>
```



You can experiment it : https://codepen.io/KaunainRizvi/pen/poGPxXo

Github Repository : https://github.com/KaunainRizvi/Learning-In-Public-Kaunain_Rizvi

Linkedin : https://www.linkedin.com/in/kaunain-rizvi/

Github Repository : https://github.com/KaunainRizvi/Learning-In-Public-Kaunain_Rizvi

Linkedin : https://www.linkedin.com/in/kaunain-rizvi/