|  |
| --- |
| Java code text. Computer software coding vector concept. Coding script ... |
| Mastering the Web and Beyond  A Comprehensive Guide to HTML, CSS, JavaScript, Java, and Python |
| |  |  |  | | --- | --- | --- | | DELL | 1/1/24 | Pai Thant Htoo | |

Contents

[Chapter 1: Introduction to HTML 3](#_Toc176277050)

[1.1 What is HTML? 3](#_Toc176277051)

[1.2 Structure of an HTML Document 4](#_Toc176277052)

[1.3 Common HTML Tags 4](#_Toc176277053)

[1.4 Creating Your First Webpage 5](#_Toc176277054)

[1.5 Forms and Input Elements 6](#_Toc176277055)

[1.6 Semantic HTML and Accessibility 7](#_Toc176277056)

[Chapter 2: Styling with CSS 9](#_Toc176277057)

[2.1 Introduction to CSS 9](#_Toc176277058)

[2.2 CSS Syntax and Selectors 11](#_Toc176277059)

[2.3 Colors in CSS 12](#_Toc176277060)

[2.4 Text Styling and Fonts 12](#_Toc176277061)

[2.5 Box Model 13](#_Toc176277062)

[2.6 Positioning Elements 14](#_Toc176277063)

[2.7 Flexbox and Grid Layout 15](#_Toc176277064)

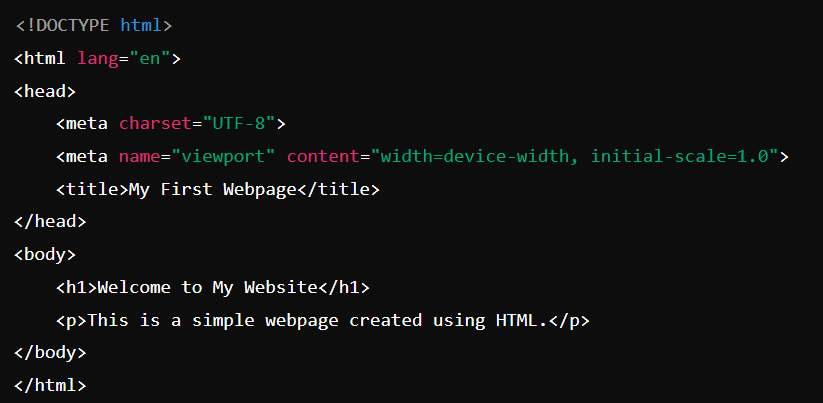
[2.8 Responsive Web Design 17](#_Toc176277065)

About Us

# Chapter 1: Introduction to HTML

## 1.1 What is HTML?

*HTML (Hyper Text Markup Language) is the standard language for creating webpages. It provides the structure of a webpage by defining elements like headings, paragraphs, links, images, and more. HTML is not a programming language; it’s a markup language that uses tags to annotate content.*

**

**In this example**

***<!DOCTYPE html>***: Declares the document type and version of HTML.

***<html>***: The root element of the HTML page.

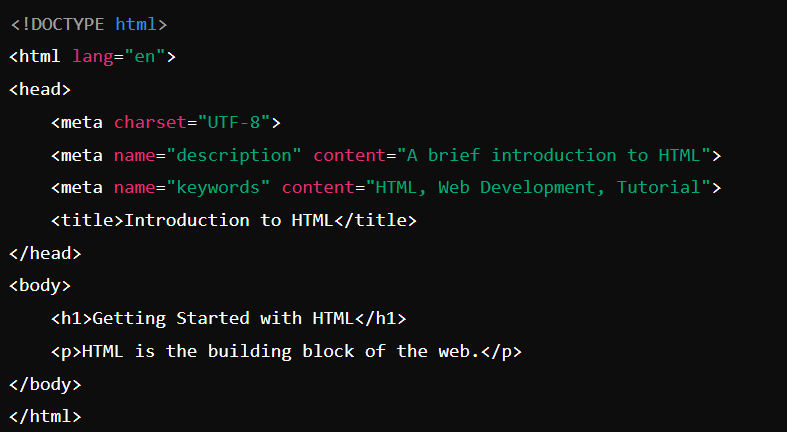
***<head>***: Contains meta-information about the document, like its title and character encoding.

***<body>***: Contains the content that will be visible on the webpage.

## 1.2 Structure of an HTML Document

An HTML document typically has two main sections:

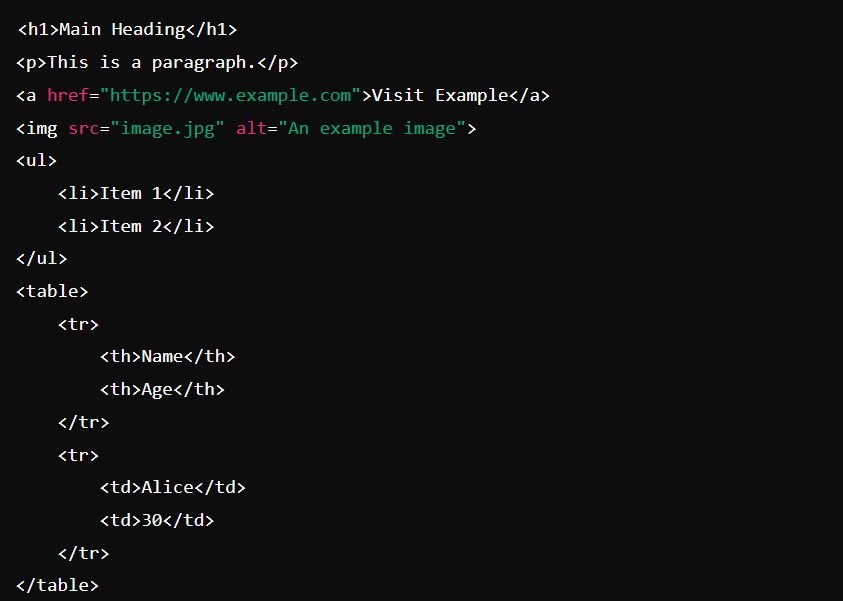
* **Head**: Contains meta-information (title, links to stylesheets, scripts).
* **Body**: Contains the actual content of the webpage (text, images, links).



## 1.3 Common HTML Tags

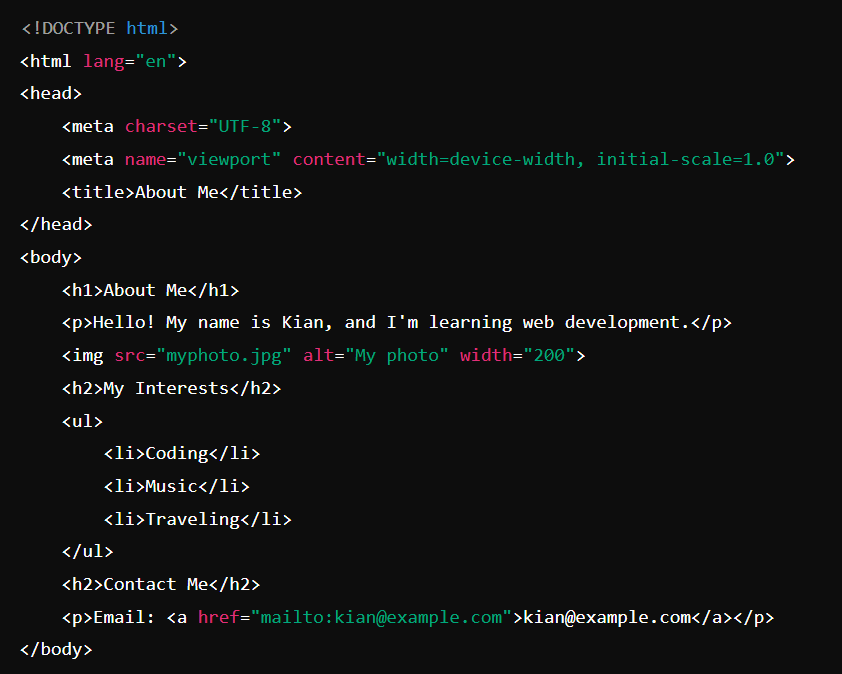
HTML uses various tags to define different elements on a webpage. Here are some of the most common ones:

* **Headings**: ***<h1>*** to ***<h6>*** define headings, with ***<h1>*** being the highest level and ***<h6>*** the lowest.
* **Paragraphs**: ***<p>*** defines a paragraph.
* **Links**: ***<a>*** defines a hyperlink.
* **Images**: ***<img>*** embeds an image.
* **Lists**: ***<ul>*** for unordered lists, ***<ol>*** for ordered lists, and ***<li>*** for list items.
* **Tables**: ***<table>*** creates a table, ***<tr>*** for table rows***, <td>*** for table data cells, and ***<th>*** for table headers.



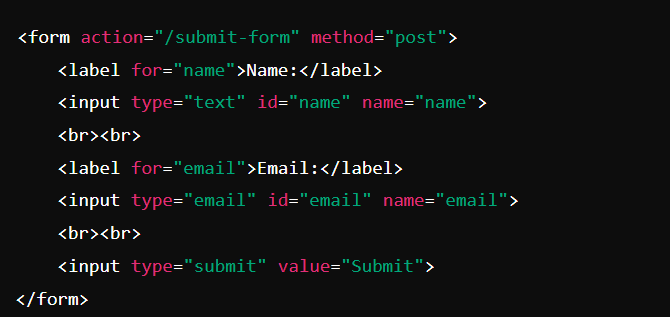
## 1.4 Creating Your First Webpage

Let’s create a simple webpage that introduces yourself:



## 1.5 Forms and Input Elements

Forms in HTML allow users to submit data to a server. Forms contain input elements like text fields, checkboxes, radio buttons, and submit buttons.



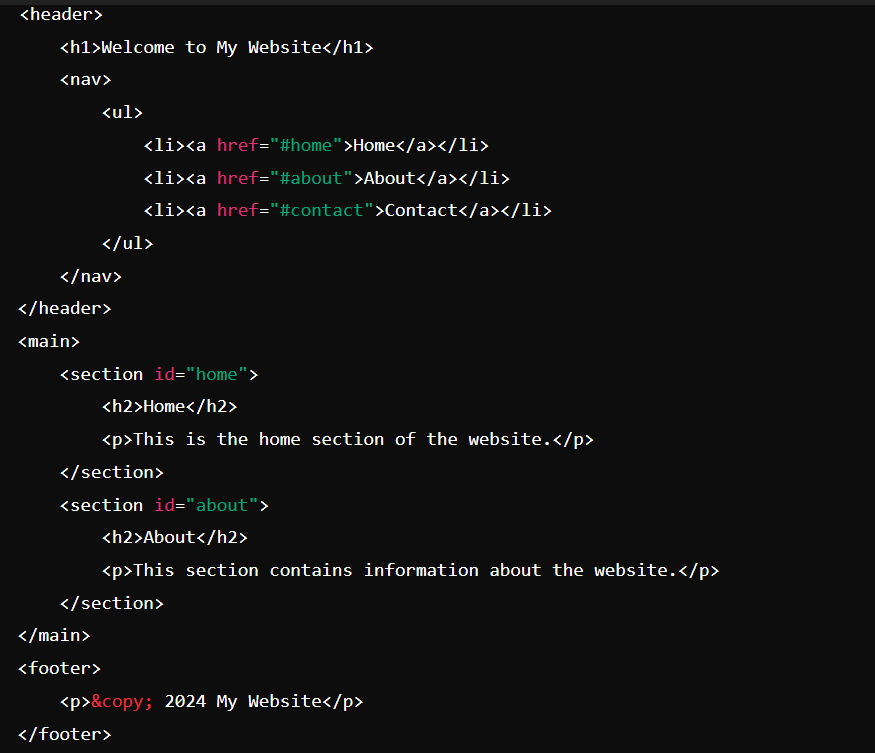
**In this example**

* ***<form>***: The container for the form.
* ***action***: The URL where the form data will be sent.
* ***method***: The HTTP method used to send the data (GET or POST).
* ***<input>:*** Defines input fields.

## 1.6 Semantic HTML and Accessibility

Semantic HTML uses elements that clearly describe their meaning in a human- and machine-readable way. Examples include ***<header>, <footer>, <article>,*** and ***<section>.***

Accessibility is about making your website usable by as many people as possible, including those with disabilities. Using semantic HTML helps improve accessibility.



**Exercise 1:** Create your own "About Me" webpage with:

* A heading
* A short paragraph about yourself
* An image of yourself
* A list of your hobbies or interests
* A contact section with your email

**Exercise 2**: Refactor your "About Me" webpage to use semantic HTML elements like ***<header>, <main>, <section>,*** and ***<footer>.***

# Chapter 2: Styling with CSS

*CSS (Cascading Style Sheets) is the language used to style HTML documents. While HTML provides the structure and content of a webpage, CSS is responsible for the visual presentation, including layout, colors, fonts, and more.*

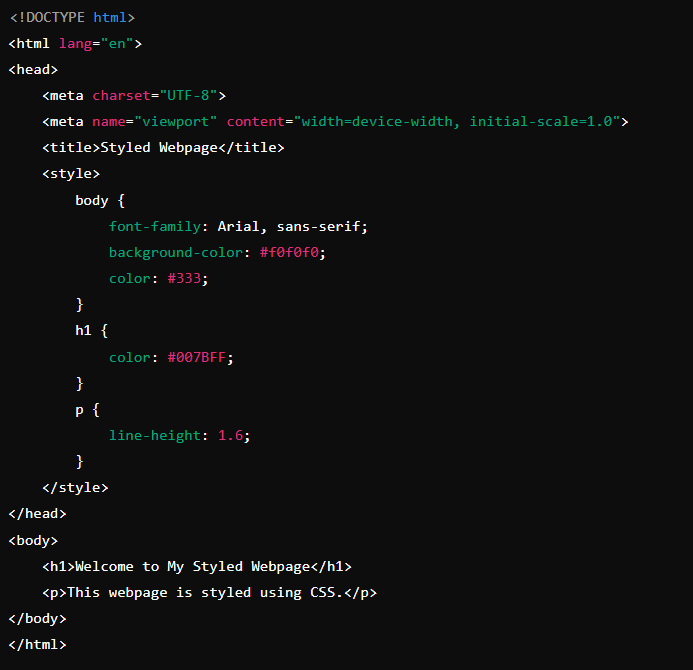
## 2.1 Introduction to CSS

CSS allows you to control the look and feel of your HTML elements. It can be applied in three different ways:

 **Inline CSS:** Applied directly within an HTML element using the ***style*** attribute.

 **Internal CSS:** Defined within a ***<style>*** tag inside the <head> section of an HTML document.

 **External CSS:** Linked in a separate file using the ***<link>*** tag.



**In this example**

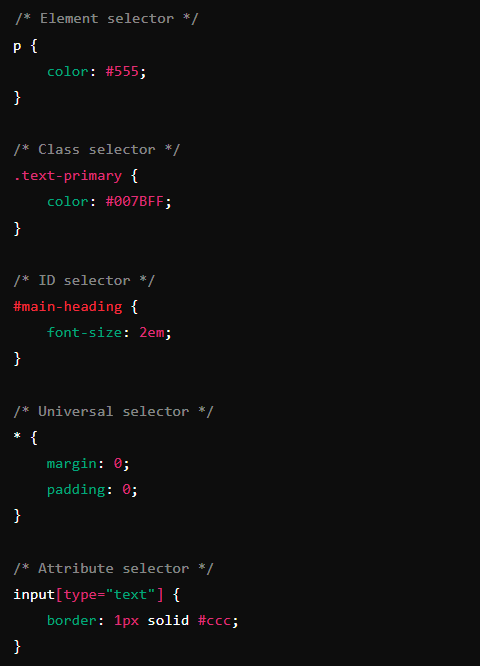
* ***font-family***: Sets the font for the webpage.
* ***background-color***: Sets the background color.
* ***color***: Sets the text color.
* ***line-height***: Controls the spacing between lines of text.

## 2.2 CSS Syntax and Selectors

CSS syntax consists of selectors and declaration blocks. The selector points to the HTML element you want to style, and the declaration block contains one or more declarations separated by semicolons.

**Common Selectors:**

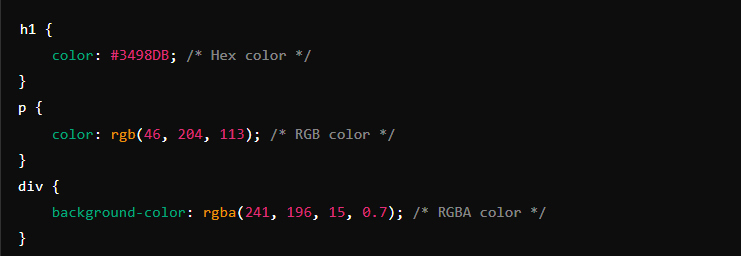
* **Element Selector:** Targets HTML elements by their tag name.
* **Class Selector:** Targets elements by their class attribute, prefixed with a . (dot).
* **ID Selector:** Targets elements by their id attribute, prefixed with a # (hash).
* **Universal Selector:** Targets all elements, represented by \*.
* **Attribute Selector:** Targets elements based on their attributes.



## 2.3 Colors in CSS

Colors can be defined using several methods:

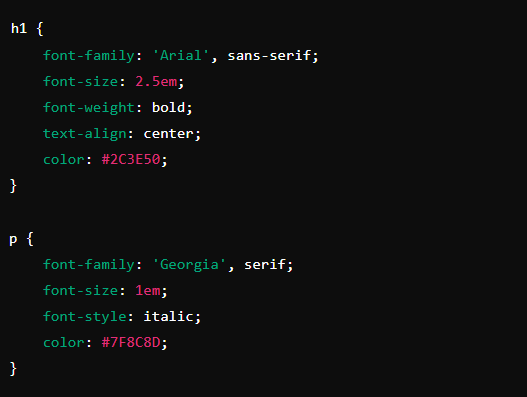
* **Named Colors:** Common names like red, blue, green.
* **Hexadecimal:** Six-digit codes like #FF5733.
* **RGB:** Values like rgb(255, 87, 51).
* **RGBA:** Adds an alpha (transparency) value like rgba(255, 87, 51, 0.5).



## 2.4 Text Styling and Fonts

CSS provides various properties to control the appearance of text:

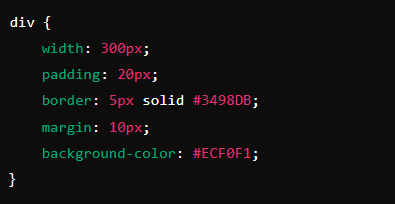
* **font-family:** Specifies the font.
* **font-size:** Defines the size of the text.
* **font-weight:** Sets the weight (boldness) of the text.
* **font-style:** Specifies whether the text is normal, italic, or oblique.
* **text-align:** Aligns the text (left, right, center, justify).



## 2.5 Box Model

The CSS box model is essential for understanding layout and design. It describes the rectangular boxes that are generated for elements in the document tree and consists of:

* **Content:** The actual content of the box, where text and images appear.
* **Padding:** Clears an area around the content inside the box.
* **Border:** Surrounds the padding and content.
* **Margin:** Clears an area outside the border, separating the element from others.



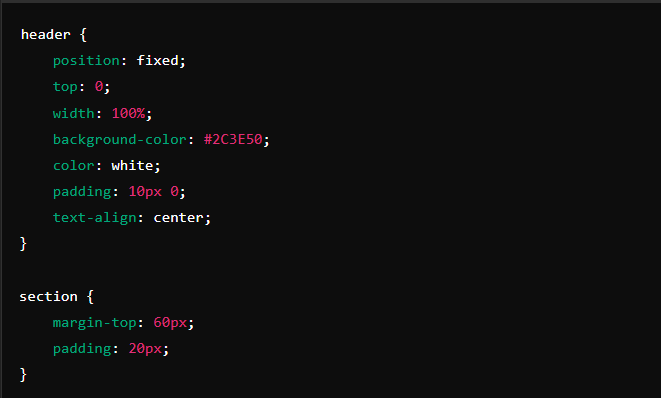
**In this example**

* The div has a content area of 300px wide.
* It has 20px padding, a 5px border, and 10px margin.

## 2.6 Positioning Elements

CSS positioning allows you to control the layout of elements in a more precise manner:

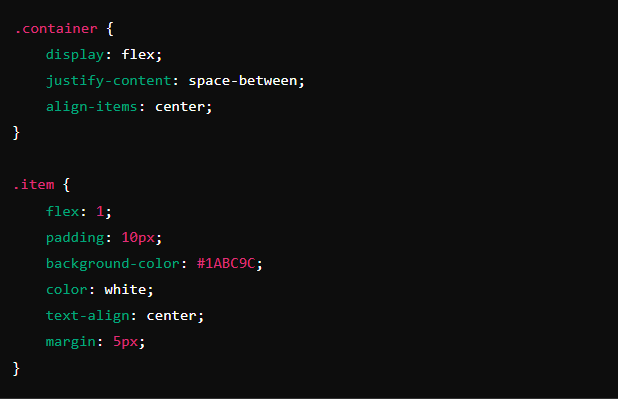
* **Static:** Default position (elements flow in the normal page layout).
* **Relative:** Position relative to its normal position.
* **Absolute:** Positioned relative to its nearest positioned ancestor.
* **Fixed:** Positioned relative to the viewport (it stays in place when the page is scrolled).
* **Sticky:** A hybrid of relative and fixed, depending on the scroll position.



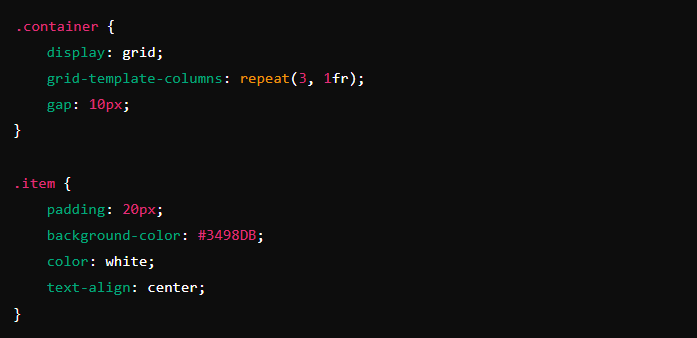
## 2.7 Flexbox and Grid Layout

CSS Flexbox and Grid are powerful layout systems that allow you to create complex designs with ease.

**Flexbox:** A one-dimensional layout system used to distribute space along a single row or column.

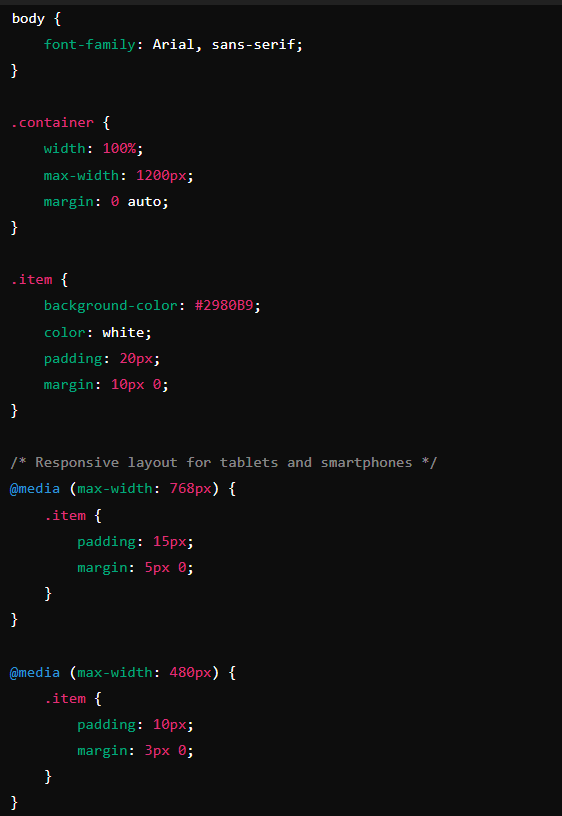


**Grid:** A two-dimensional layout system used to define both rows and columns.



## 2.8 Responsive Web Design

Responsive web design ensures that your webpage looks good on all devices, whether it's a desktop, tablet, or smartphone. Media queries are used to apply different styles for different screen sizes.



**Exercise 1:** Create a simple webpage and use different selectors to style various elements.

**Exercise 2:** Experiment with different color methods to style your webpage.

**Exercise 3:** Use different fonts, sizes, and alignments to style a block of text on your webpage.

**Exercise 4:** Create a layout using the box model, adjusting padding, borders, and margins.

**Exercise 5:** Create a fixed navigation bar that remains at the top of the page while scrolling.

**Exercise 6:** Create a webpage layout using Flexbox or Grid, with multiple items arranged in rows and columns.

**Exercise 7:** Make your webpage responsive by applying media queries to adjust the layout and design based on different screen sizes.