**Intro to HTML / CSS**

Full Course Notes

Samuel Chau

Contents

[Overview of Web Development 3](#_Toc123786546)

[Intro to HTML 3](#_Toc123786547)

[HTML Document Structure 3](#_Toc123786548)

[Text Elements 4](#_Toc123786549)

[Images and Attributes 5](#_Toc123786550)

[Hyperlinks 5](#_Toc123786551)

[Structuring A Webpage 6](#_Toc123786552)

[Semantic HTML 6](#_Toc123786553)

[Intro to CSS 7](#_Toc123786554)

[Inline, Internal and External CSS 7](#_Toc123786555)

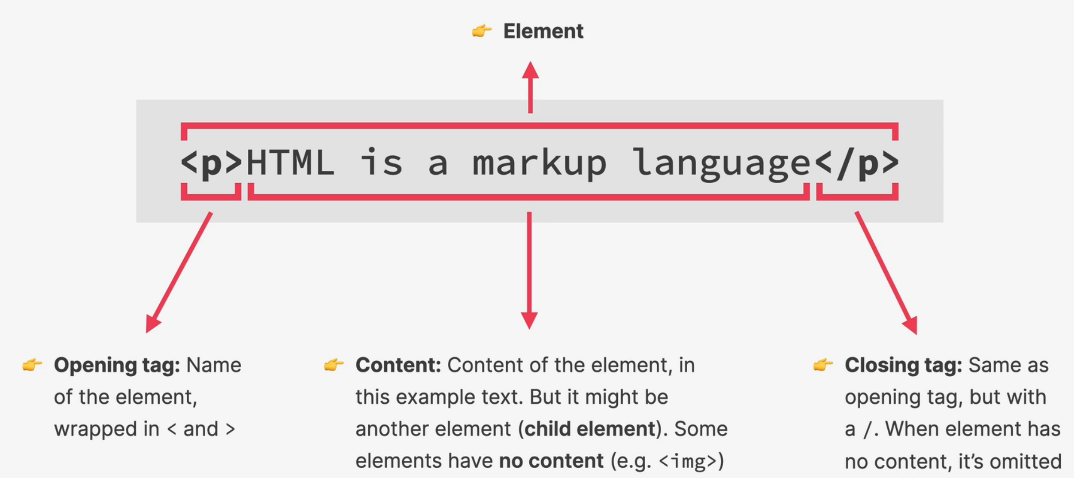
[Styling Text 8](#_Toc123786556)

[Combining Selectors 8](#_Toc123786557)

[Class and ID Selectors 8](#_Toc123786558)

[Working With Colours 9](#_Toc123786559)

Overview of Web Development

* Websites are written using HTML, CSS, and JavaScript. The HTML is the content of the webpage, the CSS defines how the content is displayed and the JS describes what the content can do. Combined, they make up the process of front-end development.
* Websites exist on some sort of server, where they are requested by some web browser and sent in response. These websites can exist statically, where the contents of the website are not changed, or dynamically, where the contents can be changed according to a database or other web application.
* We use back-end languages to define how dynamic websites can be updated, this process is called back-end development.

Intro to HTML

* HTML stands for Hyper Text Markup Language
* Used to structure and describe the content of a webpage
* Is a markup language, not a programming language
* Consists of elements that describe types of content
  + Paragraphs <p></p>
  + Headings <h></h>
  + Links
  + Images
  + Videos
  + Etc
* Browsers understand HTML and render the code as a website

HTML Document Structure

<!DOCTYPE html> // tells the browser how to render the following code

<html>

// The TAB

  <head>

    <title>The Basic Language of the Web: HTML</title> // the tab title

  </head>

// The Actual web page

  <body>

    <h1>The Basic Language of the Web: HTML</h1> // first heading shown on webpage

  </body>

</html>

Text Elements

* h1, h2, h3…. Gets smaller the higher the number

  <body>

    <h1>The Basic Language of the Web: HTML</h1>

    <h2>The Basic Language of the Web: HTML</h2>

    <h3>The Basic Language of the Web: HTML</h3>

    <h4>The Basic Language of the Web: HTML</h4>

    <h5>The Basic Language of the Web: HTML</h5>

    <h6>The Basic Language of the Web: HTML</h6>

  </body>

* <p></p> for paragraphs

    <p>

      In this post, let's focus on HTML. We will learn what HTML is all about,

      and why you too should learn it.

    </p>

* <strong>CONTENT</strong> to bold text

    <p>

      HTML stands for <strong>H</strong>yper<strong>T</strong>ext

      <strong>M</strong>arkup <strong>L</strong>anguage. It's a markup language

      that web developers use to structure and describe the content of a webpage

      (not a programming language).

    </p>

* <em></em> to italicise

    <p>

      All modern websites and web applications are built using three

      <em>fundamental</em>

      technologies: HTML, CSS and JavaScript. These are the languages of the

      web.

    </p>

* <ol></ol> for ordered list

    <ol>

      <li>The opening tag</li>

      <li>The closing tag</li>

      <li>The actual element</li>

    </ol>

* <ul></ul> for unordered list

    <ul>

      <li>To be able to use the fundamental web dev language</li>

      <li>

        To hand-craft beautiful websites instead of relying on tools like

        Worpress or Wix

      </li>

      <li>To build web applications</li>

      <li>To impress friends</li>

      <li>To have fun 😃</li>

    </ul>

* <li></li> for list element, goes within body of list element

Images and Attributes

    <img

      src="post-img.jpg"

      alt="HTML code on a screen"

      width="500"

      height="200"

    />

* Attributes define the characteristics of an image element (and other elements too)
  + src describes the path of the image file
  + alt is the description of the image
  + width, height is used to resize the image. Only define one to scale the image linearly. Defining both will stretch the image if not linearly scaled
* no closing tag needed
* attributes also applied to doctype and header. This is usually done automatically when creating a HTML file

<!DOCTYPE html>

<html lang="en"> // says the language of this web page is english

  <head>

    <meta charset="UTF-8" />

    <title>The Basic Lanaguage of the Web: HTML</title>

  </head>

Hyperlinks

* <a href=”URL” target=”\_blank></a> for link (a stands for anchor)
  + Where href is an attribute that defines the URL
  + Target=”\_blank” will open the URL in a new tab

    <p>

      You can learn more at the

      <a

        href="https://developer.mozilla.org/en-US/docs/Web/HTML"

        target="\_blank"

        >MDN Web Docs</a

      >

    </p>

* Can also include links to other local local html webpages
  + Href is the path of the local HTML file in relation to the index file

    <a href="blog.html">Blog</a>

    <a href="challenges.html">Challenges</a>

    <a href="flexbox.html">Flexbox</a>

    <a href="CSS.html">CSS</a>

    <a href="grid.html">Grid</a>

* Can also use href=”#” to scroll the current page back to the top
* Hyperlinks can also be used with images
  + <a href=”src”> <img src=”src” /> /a>

Structuring A Webpage

* We can group together similar parts of our webpage using elements
  + <nav></nav> used to group links together
  + <header></header>
  + <footer></footer>
  + <article></article>
* We can add groups within groups too

    <header>

      <h1>📘 The Code Magazine</h1>

      <nav>

        <a href="blog.html">Blog</a>

        <a href="challenges.html">Challenges</a>

        <a href="flexbox.html">Flexbox</a>

        <a href="CSS.html">CSS</a>

        <a href="grid.html">Grid</a>

      </nav>

    </header>

Semantic HTML

* Certain elements have a meaning and should be considered, regardless of whether it affects how the webpage looks
* Therefore, we use <nav> instead of <div> for a section of links, etc
* It’s to categorise our code and make it easier to read and understand, not to improve how the webpage looks

Intro to CSS

* Stands for Cascading Style Sheets
* Used to style HTML properties – text, spacing, layout, colour, etc
* Use a selector to pick a HTML element
* Declare a property (size, colour, etc) with a value (10, blue, etc)
  + These properties go within the declaration block contained in curly braces
* Together, a selector and declaration block is called a **CSS Rule**

Diagram

Description automatically generated

Inline, Internal and External CSS

* Inline includes the CSS within the HTML file
  + Should never be used
* Internal includes writing a *style* element within the head.

  <style>

    h1 {

      color: blue;

    }

  </style>

  <head>

    <meta charset="UTF-8" />

    <title>The Basic Lanaguage of the Web: HTML</title>

  </head>

…

* External includes writing the CSS in a separate CSS file
* We link the CSS and HTML file within the HTML head, using the link element
* Index.html

  <head>

    <meta charset="UTF-8" />

    <link href="style.css" rel="stylesheet" />

    <title>The Basic Lanaguage of the Web: HTML</title>

  </head>

* Style.css

h1 {

  color: blue;

}

Styling Text

h1 {

  color: skyblue;

  font-size: 26px;

  font-family: sans-serif;

  text-transform: uppercase;

  font-style: italic;

  text-align: center;

}

* Style <li>’s, not <ol>’s or <ul>’s
* Elements within a stylised element inherit it’s parent style unless it has its own style.

Combining Selectors

* Can apply a property to multiple selectors at once
* Reduces the need to write similar code for every selector

h1, h2, h3, h4, p, li {

  font-family: sans-serif;

}

* Can also use descendant selectors to highlight an element within an element
  + Can be nested, elements within elements within elements, etc

footer p {

  font-size: 16px;

}

Class and ID Selectors

* We can assign an ID to an element and then select than ID in the stylesheet using #ID

    <footer>

      <p id="copyright">Copyright &copy; 2027 by the Code Magazine</p>

    </footer>

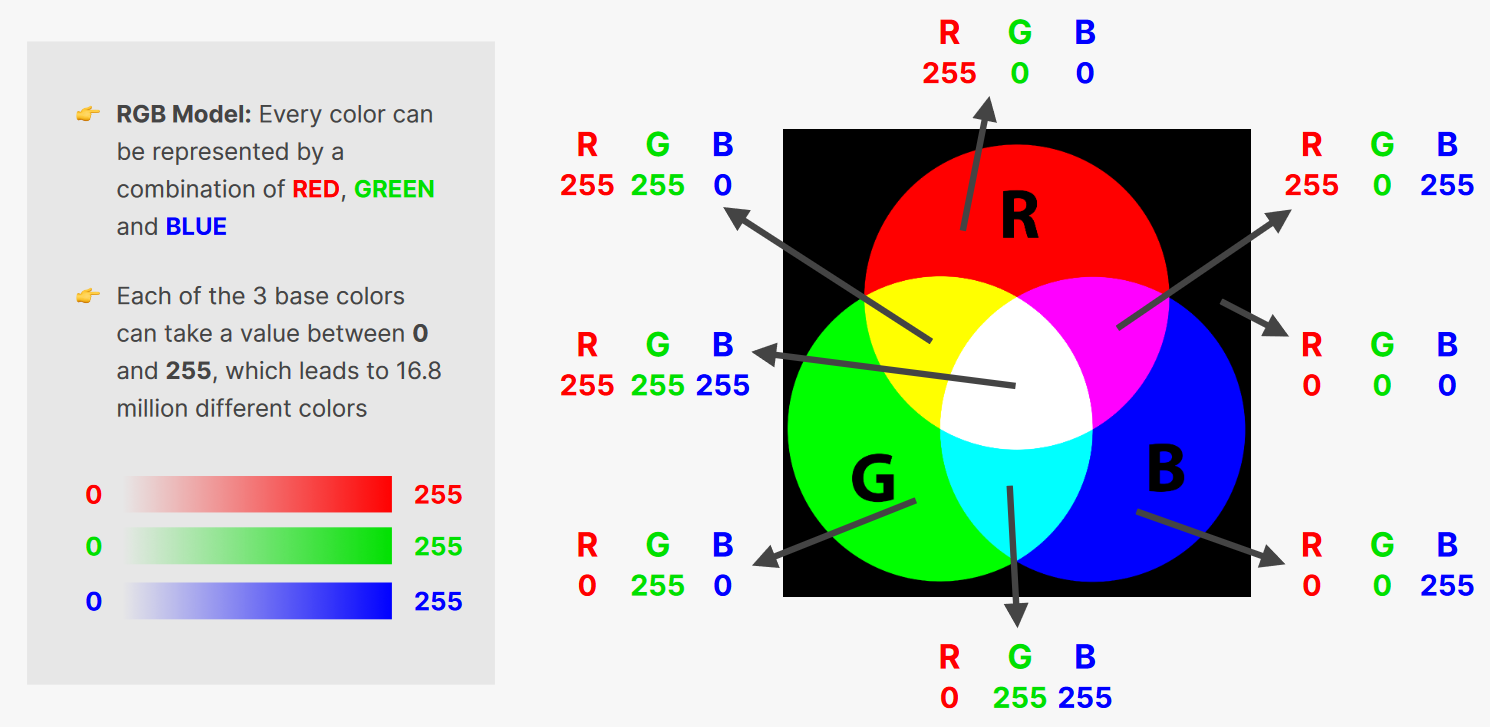
#copyright {

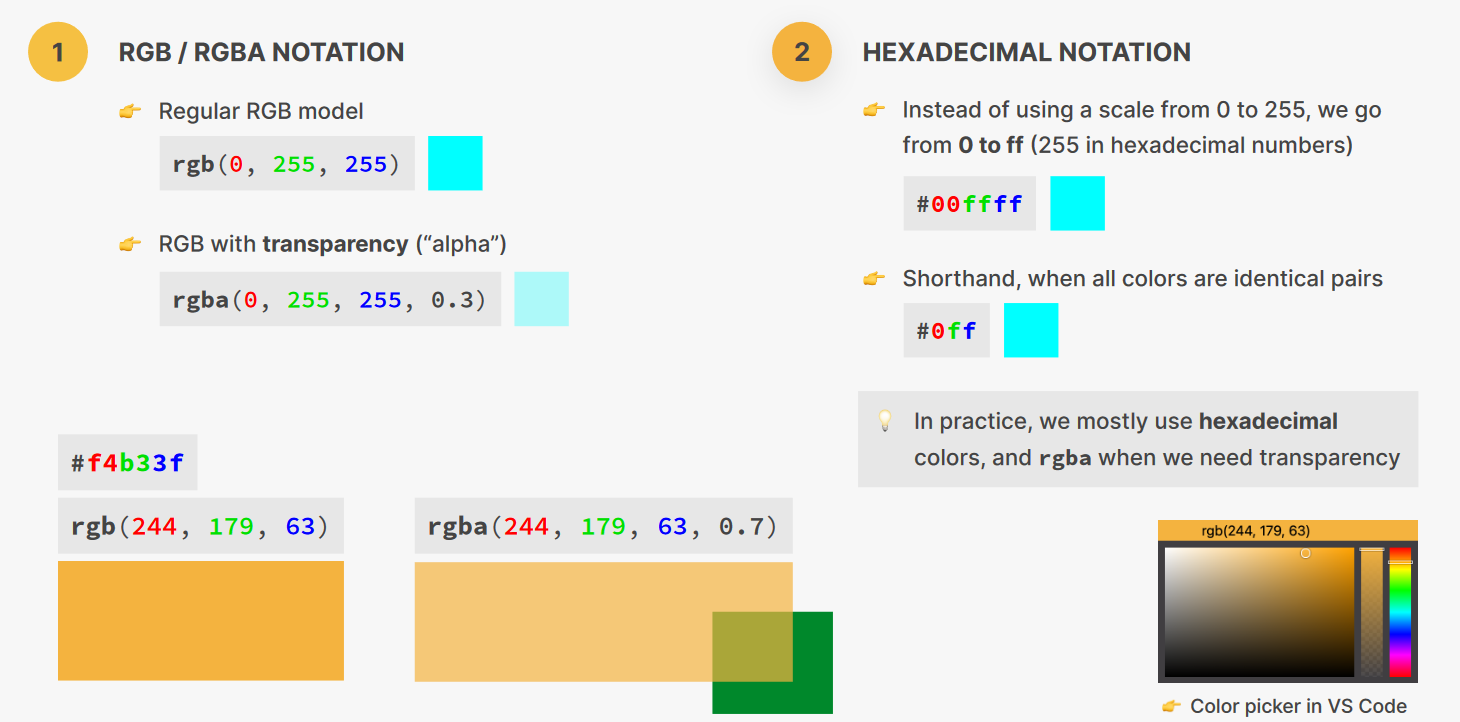
  font-size: 16px;

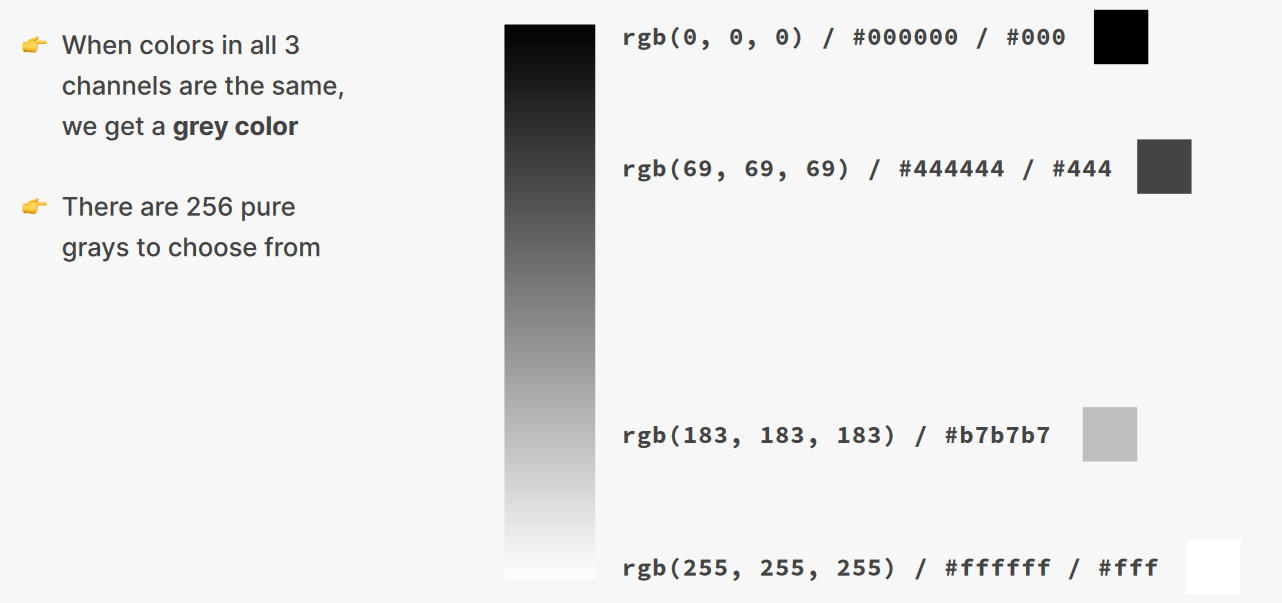
}

* When we want to assign the same ID to multiple elements, we instead assign a *class*
* However, we should always use classes over IDs in every scenario to future proof

Working With Colours

* The RGB Model





Pseudo Classes

* We can target nested elements and assign variables to them

li:first-child {

  font-weight: bold;

}

* Target all first elements from lists

li:last-child {

  font-style: italic;

}

* Target all last elements from lists

li:nth-child(2) {

  color: red;

}

* Target the second element from lists

li:nth-child(odd) {

  color: red;

}

* Target all odd elements from lists

Styling Hyperlinks

* Should only style links, not entire anchor element

a:link {

  color: #1098ad;

  text-decoration: none;

}

* Also style clicked links, should be the same as original colour

a:visited {

  color: #1098ad;

}

* Also important to style the hover attribute of links to show the user it is a link

a:hover {

  color: red;

  font-weight: bold;

  text-decoration: underline dotted;

}

* Can also style the active attribute (the link is being clicked)

a:active {

  background-color: black;

}

* Should always be styled in this order – link, visited, hover and active