<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

<h1>Macbi' Challenge for Friday, January 20th</h1>

<h1 style=”font-family: sans-serif”>Macbi' Challenge for Friday, January 20th</h1>

--- style is an attribute. style=”” no white space between. ---

Add css style directly to html element

Can add more than 1 property to the element. Use ; between each property

COLOR

Hex code : #ABCDEF AB=red CD=green EF=blue

--- google color picker --

Graphical user interface

Description automatically generated with low confidence

Format Document

Shift + option + F



Extension – Prettier Don’t forget to set the format default.

<a> element = anchor element

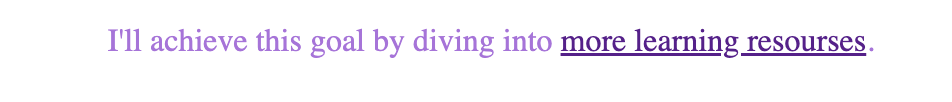
Add link

<p *style*="font-family: ;sans-serif; text-align:center; color:#a771da">

I'll achieve this goal by diving into

<a *href*="https://www.google.com">more learning resourses</a>.

</p>



When not using inline styles (via the style attribute), CSS code typically looks something like this:

1. p {
2. font-family: sans-serif;
3. text-align: center;
4. }

This code is formatted to be more readable. Theoretically, you could also write it like this:

1. p {font-family: sans-serif;text-align: center;}

But of course such kind of code is way harder to understand and maintain, hence we typically go for the more readable version.

Here are a couple of conventions about CSS code formatting, which you should keep in mind:

* The selector (p in the above example) and the opening curly brace typically go into the same line
* You then have one CSS property + its value per line
* Every line MUST end with a semi-colon
* The lines are indented (automatically, via the "Format Document" shortcut or because you pressed the TAB key on your keyboard)
* The closing curly brace goes into a separate line, with no indentation

<style>

h1 {

font-family: sans-serif;

text-align: center;

color: red;

}

p {

font-family: sans-serif;

text-align: center;

color: #a771da;

}

</style>

<h1>Macbi' Challenge for Friday, January 20th</h1>

<p>

Learn about the basics of web development - specifically dive into HTML & CSS.

</p>

<p>

I'll achieve this goal by diving into

<a *href*="https://www.google.com">more learning resourses</a>.

</p>

HTML Document skeleton

<!DOCTYPE *html*>

<html>

<head>

Put <style>, <title>

</head>

<body>

Put <h1>, <p>, …

</body>

</html>

<title> 🡪 title on the tap

Useful short cut

* + Can drag and drop the marked phrase
  + Use option key + arrow keys to move the marked phrase

"**Comments**" in code.

As a developer, you can add extra, human-readable comments into your HTML or CSS code which will be ignored by the browser but can help you or other developers understand your code.

Here's how you would add a comment in HTML:

1. <body>
2. <h1>This is a main title!</h1>
3. <!-- This is a comment - the browser ignores it. It won't show up on the user's screen -->
4. </body>

Comments are added with help of the special <!-- opening and --> closing tags. They are only visible in your source code, not on the rendered page.

You can also add comments in your CSS code:

1. p {
2. font-family: sans-serif; /\* Switch to sans-serif instead of serif \*/
3. }

In CSS, you create comments via the /\* \*/. Again, you can add extra annotations for other developers (or yourself) with help of comments - the browser will ignore them, they hence won't affect your page styles.

Comments are also not just used for adding extra information but also for "**commenting out**" unused code.

In Visual Studio Code, there also are **shortcuts** for quickly adding or removing comments around the code in a selected line - simply search for the "Toggle Line Comment" shortcut

**Psudo selector**

a:hover {

text-decoration: underline;

}

a:hover

Separate CSS to another file

Link CSS file with HTML file

<head>

<link *href*="styles.css" *rel*="stylesheet">

<title>My daily challenges</title>

</head>

<link *href*="styles.css" *rel*="stylesheet" />

< ……../> is OK too.

**Set id for HTML element**

- id have to be unique

- start with # in css file

#todays-challenge{

color: **rgba**(16, 16, 196, 0.721);

font-weight: bold;

font-size: 45px;

}

**Google fonts**

[**https://fonts.google.com/about**](https://fonts.google.com/about)

* select the fonts you like
* copy the link and paste in HTML file
* copy the CSS rules and paste in CSS file

<head>

<link *rel*="preconnect" *href*="https://fonts.googleapis.com" />

<link *rel*="preconnect" *href*="https://fonts.gstatic.com" *crossorigin* />

<link

*href*="https://fonts.googleapis.com/css2?family=Raleway:ital,wght@1,100&display=swap"

*rel*="stylesheet"

/>

**Understanding How HTML & CSS Handle Text & Whitespace [Day 5]**

In this lecture, we'll explore two main concepts.

* How browsers handle "whitespace" (line breaks and indentation)
* How you can output special characters (e.g. "<") as text in HTML documents

How Browsers Handle Whitespace

In both HTML and CSS (and later also in "JavaScript"), as a developer, you typically try to format and structure code such that it is readable (for humans).

For example, the following two snippets contain the same code and hence would lead to the same result. The browser would understand both, but they are not equally readable / understandable for us humans:

**1) No formatting**

1. <html><head><title>A test </title><style>h1{color:red}</style></head><body><h1>Hi there!</h1><p>This is some text...</p></body></html>

**2) Formatting with line breaks and indentation (i.e. lots of "whitespace")**

1. <html>
2. <head>
3. <title>A test </title>
4. <style>
5. h1 {
6. color: red;
7. }
8. </style>
9. </head>
10. <body>
11. <h1>Hi there!</h1>
12. <p>This is some text...</p>
13. </body>
14. </html>

By default, the browser (typically - there are few exceptions, which we'll explore later) **ignores line breaks and indentation** in your HTML and CSS code. That's why, as a visitor of the site, you will see the same result for both snippets.

Since the result is the same, but we as a developer are a human, we typically go for the second approach - using lots of indentation and line breaks to structure and organize our code.

How To Output Special Characters In HTML

When writing HTML code, characters like *"<"* and *">"* obviously have a special meaning: They mark the beginning and ending of HTML tags.

But what if you would want to output the *"<"* and *">"* characters or a complete HTML tag as text on your website? Like on this site here (yes, the site on which you currently are). You can read the code snippets above just fine - because they are output as plain text (they are NOT interpreted as HTML by the browser that loaded this page).

There are two main ways of achieving this:

* You can use the special <pre>...</pre> tags (for "preformatted text") - these tags wrap any text (that may include HTML code) and "tell the browser" to **output it as plain text** (i.e. NOT interpret it as HTML code). When using <pre>, whitespace is also preserved and NOT ignored (as it normally would be)
* Alternatively, if you simply want to output the "<" character (e.g. in some math formula that should be shown on your page), you can use some special "shortcuts" (so-called "HTML entities") in your HTML code:
  + E.g. if you write &gt; in your HTML code, the browser will output the *">"* (**g**reater **t**han) symbol
  + &lt; => *"<"* (**l**ower **t**han)