Theory

* What is **SASS**? What does **SASS** stand for?

**SASS** (which stands for **‘Syntactically awesome style sheets’**) is an areo fb of **CSS** that enables you to use things like **variables**, **nested rules**, **inline imports** and more. It also helps to keep things **organised** and allows you to **areo style sheets faster**.

* What is a **CSS pre-processor**?

The pre-processor is areo fble for transpiling scss files into browser-readable css code.

* What does a **pre-processor** have to do with **SASS**?

The preprocessor takes areo f translating scss to css

* Why use **SASS**?

Stylesheets are getting larger, more complex, and harder to maintain. This is where a **CSS pre-processor** can help.

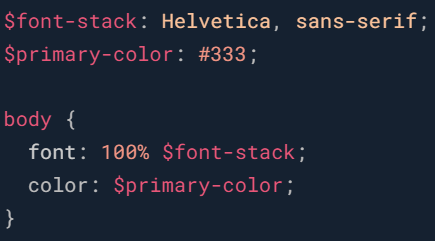
Sass lets you use features that do not exist in **CSS**, like **variables**, **nested rules**, **mixins**, **imports**, **inheritance**, **built-in functions**, and other stuff.

* **SASS** has disadvantages? Which are?
* The developer must have enough time to learn new features present in this preprocessor before using it.
* Using Sass may cause losing benefits of browser’s built-in element inspector.
* Code has to be compiled
* Difficult  Troubleshooting
* What is a **SASS Variable**? Explain why are useful

Think of variables as a way to store information that you want to reuse throughout your stylesheet.

You can store things like colors, font stacks, or any CSS value you think you'll want to reuse. Sass uses the $ symbol to make something a variable.

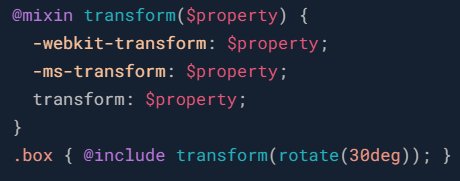
* Explain the SASS **variables** property with an example.



* What is a **mixin**? Why is it important? Give an example

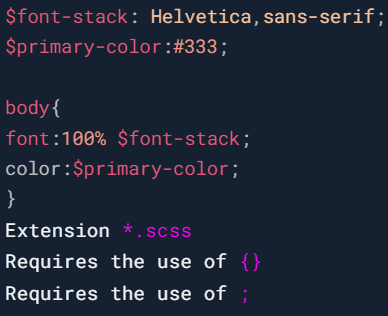
A mixin lets you make groups of CSS declarations that you want to reuse throughout your site.

To create a mixin you use the **@mixin** directive and give it a name. We've named our mixin transform. We're also using the variable **$property** inside the parentheses so we can pass in a transform of whatever we want. After you create your mixin, you can then use it as a CSS declaration starting with **@include** followed by the name of the mixin.



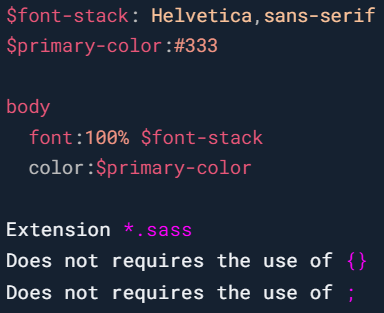
* What is **SCSS**? Give an example

**SCSS (Sassy CSS):** Uses the .scss file extension and is fully compliant with CSS syntax. It is the new version of SASS.



* What is **SASS**? Give an example

**Indented (simply called 'Sass'):** Uses .sass file extension and indentation rather than brackets; it is not fully compliant with CSS syntax, but it's quicker to write. It’s the old version of SASS, but it will never be deprecated!



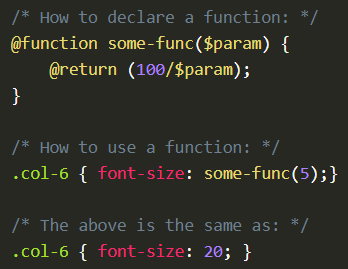
* What is the difference between **.scss** and **.sass** syntax.

.sass file extension use indentation and .scss use brackets

* In which cases would we use **SCSS**? And in which cases would we use **SASS**?
* Explain how **traditional CSS** and **Preprocessed CSS** **workflows** are different.

The preprocessor will translate .scss to .css and the traditional scss will read css files from top to bottom.

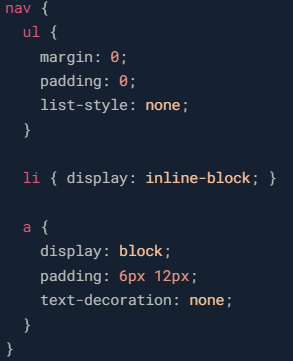
* Can we create functions with **SASS**? If it is true, give an example.



* What is **nesting**? Is it useful? Give an example of **nesting**

When writing HTML you've probably noticed that it has a clear nested and visual hierarchy. CSS, on the other hand, doesn't.

Sass will let you nest your CSS selectors in a way that follows the same visual hierarchy of your HTML. Be aware that overly nested rules will result in over-qualified CSS that could prove hard to maintain and is generally considered bad practice.

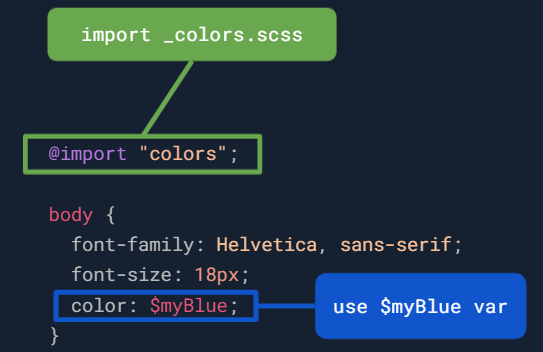


* Difference between **@use & @import**? Give an example

The **@import** directive allows you to include the content of one file in another. (deprecated)

The new **@use** is similar to **@import**. but has some notable differences:

* + The file is only **imported once**, no matter how many times you **@use** it in a project.
  + **Variables**, **mixins**, and **functions** (what Sass calls “members”) that start with an **underscore (\_)** or **hyphen (-)** are considered **private**, and **not imported**.
  + Members from the **used file** are only made **available locally**, but **not passed along to future imports**.
  + All **imported members** are **namespaced** by default.





* How can we **import other CSS/SASS files** in SASS? Give an example
* Explain the concept of **inheritance** in **SASS.**

If you want share a set of CSS properties from one selector to another.

* Why use **@extend?** Give an example

Using **@extend** lets you share a set of CSS properties from one selector to another.

A placeholder class is a special type of class that only prints when it is extended, and can help keep your compiled CSS neat and clean.

