The Complete Sass & SCSS Course – Quizzes

Quizzes should be included directly after the lecture mentioned above it, e.g. 20.10.preprocessors.

The symbol 🡨 signifies the correct answer.

**20.10.preprocessors**

Formularbeginn

Question 1:

**What are some of the advantages of a CSS preprocessor like Sass?**

* 

Additional capabilities like variables and nesting.

* 

More reusable and flexible code.

* 

Automatic vendor prefixing.

* 

All of the above. 🡨

Formularbeginn

Question 2:

**Can you use frameworks like Bootstrap from Sass as well?**

* 

Sure! 🡨

* 

Nope!

Formularende

**20.20.sass-vs-scss**

Formularbeginn

Question 1:

**What's a fundamental difference between the Sass syntax and the SCSS syntax?**

* 

SCSS uses curly braces and semi-colons to structure the code. 🡨

* 

Sass uses curly braces and semi-colons to structure the code.

* 

There is no difference in the syntax.

Formularende

Formularbeginn

Question 2:

**Which syntax is closer to CSS?**

* 

Sass

* 

SCSS 🡨

* 

Both are equally similar to CSS.

Formularende

**40.10.variables**

Formularbeginn

Question 1:

**Which of the following is an advantage of variables?**

* 

Declarations are in one place and not spread throughout the code.

* 

The code is more modifiable and flexible; you can just adjust the variable values to change a design.

* 

The code becomes more readable when using expressive variable names.

* 

All of the above. 🡨

Formularende

Formularbeginn

Question 2:

**How can you create a variable using SCSS syntax?**

* 

awesome-color: #541F14;

* 

$awesome-color: #541F14; 🡨

* 

$awesome-color = #541F14;

Formularende

**50.10.partials**

Formularbeginn

Question 1:

**What are partials?**

* 

SCSS or Sass files that are not translated to CSS files 🡨

* 

SCSS or Sass files that cannot be imported in other files

* 

Partials are unfinished SCSS or Sass files.

Formularende

Formularbeginn

Question 2:

**What can we use partials for?**

* 

To define all our variables in one or more separate files.

* 

To define a reset file that can be imported as the basis for the design.

* 

To define any SCSS code which should not be translated to a CSS file on its own.

* 

All of the above. 🡨

Formularende

Formularbeginn

Question 3:

**How can we create a partial to hold colors?**

* 

By creating a file called colors.scss in the scss directory.

* 

By creating a file called \_colors.scss in the scss directory. 🡨

* 

By creating a file called colors-partial.scss in the scss directory.

Formularende

Formularbeginn

Question 4:

**What's the best way to import a \_colors.scss partial from another file in the same directory?**

* 

@import "\_colors.scss";

* 

@import "colors.scss";

* 

@import "colors"; 🡨

Formularende

**60.10.mixins**

Formularbeginn

Question 1:

**What are mixins?**

* 

Reusable chunks of code that you can then include throughout your stylesheet. 🡨

* 

Mixins are separate SCSS/Sass files which are not translated to CSS files.

* 

Mixins are styles that are applied directly where they are defined.

Formularende

Formularbeginn

Question 2:

**How can you define a mixin?**

* 

clearfix @mixin { ... }

* 

mixin clearfix { ... }

* 

@mixin clearfix { ... } 🡨

Formularende

Formularbeginn

Question 3:

**How can you include a mixin in another part of your stylesheet?**

* 

Using the @import directive.

* 

Using the @include directive. 🡨

* 

Using the @mixin directive.

Formularende

**60.20.mixins-with-arguments**

Formularbeginn

Question 1:

**How can you create a mixin with two arguments?**

* 

@mixin awesome($color) { ... }

* 

@mixin awesome(color, font-size) { ... }

* 

@mixin awesome($color, $font-size) { ... } 🡨

Formularende

Formularbeginn

Question 2:

**How can you create a mixin with one argument and a default value?**

* 

@mixin tiny(size: 8px) { ... }

* 

@mixin tiny($size: 8px) { ... } 🡨

* 

@mixin tiny($size = 8px) { ... }

Formularende

Formularbeginn

Question 3:

**How can you include a mixin defined like this: @mixin fancy($color: #572820) { ... }**

* 

@include fancy;

* 

@include fancy(#672195);

* 

@include fancy($color: #672195);

* 

All the above are correct. 🡨

Formularende

Formularbeginn

Question 4:

**How can you define a mixin with a vararg (variable argument)?**

* 

@mixin add-backgrounds($backgrounds...) { ... } 🡨

* 

@mixin add-backgrounds($backgrounds) { ... }

* 

@mixin add-backgrounds(vararg $backgrounds) { ... }

Formularende

**60.30.mixins-content-blocks**

Formularbeginn

Question 1:

**How can we include a mixin called ie7-only and change the style of the body tag?**

* 

@mixin ie7-only { body { ... } }

* 

@include ie7-only { body { ... } } 🡨

* 

include ie7-only { body { ... } }

Formularende

Formularbeginn

Question 2:

**How can you create a mixin that uses content passing?**

* 

@mixin ie7-only { ... content; ... }

* 

@mixin ie7-only($content) { ... }

* 

@mixin ie7-only { ... @content; ... } 🡨

Formularende

**70.10.imports**

Formularbeginn

Question 1:

**How can you import an SCSS file called "base.scss" from a subdirectory "vendor"?**

* 

@import "base.scss";

* 

@import "vendor/base"; 🡨

* 

import "vendor/base";

Formularende

Formularbeginn

Question 2:

**Which of the following Sass imports will be translated to a CSS import?**

* 

@import url(https://fonts.googleapis.com/css?family=Roboto);

* 

@import "normalize.css";

* 

@import "print-style" print;

* 

All of the above. 🡨

Formularende

**80.10.nested-media-queries**

Formularbeginn

Question 1:

**Any CSS media query is also a valid media query in SCSS.**

* 

Yes! 🡨

* 

No!

Formularende

Formularbeginn

Question 2:

**In Sass, you can add media queries at any nesting level.**

* 

Yes we can! 🡨

* 

No!

Formularende

**90.20.color-arithmetic**

Formularbeginn

Question 1:

**In Sass, you can simply calculate dynamic sizes and colors right inside your stylesheets. *(Note: calculations in stylesheets work the same as in the Sass interactive shell).***

* 

Yes! 🡨

* 

No!

Formularende

Formularbeginn

Question 2:

**In Sass, *font-size: 1em + 3px;* will increase the font size of the selected element by 3px.**

* 

Sure!

* 

No way! 🡨

Formularende

Formularbeginn

Question 3:

**To manipulate colors in Sass, you must always use basic operators like +, -, \*, and /.**

* 

Yes, there is no other way.

* 

No, Sass is not that stupid. 🡨

Formularende

**100.30.creating-functions**

Formularbeginn

Question 1:

**Functions are great to encapsulate recurring logic in a reusable way.**

* 

Absolutely! 🡨

* 

Not really.

Formularende

Formularbeginn

Question 2:

**Sass provides built-in functions like darken($color, $amount) and opacify($color, $amount) to make your day easier.**

* 

Yes! 🡨

* 

No!

Formularende

Formularbeginn

Question 3:

**How can you create a function in Sass?**

* 

@function mix-colors = ($color1, $color2) { ... }

* 

@function mix-colors($color1, $color2) { ... } 🡨

* 

function mix-colors($color1, $color2) { ... }

Formularende

Formularbeginn

Question 4:

**How can you return a value from a function?**

* 

return some-value;

* 

@return: some-value;

* 

@return some-value; 🡨

Formularende

**110.30.extend-vs-mixins**

Formularbeginn

Question 1:

**What can you do with inheritance in Sass?**

* 

Encapsulate functionality to reuse later.

* 

Extend the styles of an existing selectors with additional styles. 🡨

* 

Store values to reuse in other places.

Formularende

Formularbeginn

Question 2:

**Which directive is used for inheritance in Sass?**

* 

@extend 🡨

* 

@inherit

* 

@extends

Formularende

Formularbeginn

Question 3:

**How does Sass handle extend-only selectors?**

* 

It will compress the CSS code generated from it.

* 

It will generate a separate CSS file for these.

* 

It will not translate these to CSS (which saves file size). 🡨

Formularende

Formularbeginn

Question 4:

**How can you define an extend-only selector called base-box?**

* 

.base-box

* 

#base-box

* 

%base-box 🡨

Formularende

Formularbeginn

Question 5:

**Which rule do you have to keep in mind when using @extend?**

* 

You can only extend class selectors.

* 

You cannot extend more than one base selector.

* 

Inside a media query, you cannot extend a selector defined outside that media query. 🡨

Formularende

**120.30.mixin-with-if**

Formularbeginn

Question 1:

**What can you do with conditional directives?**

* 

Generate different CSS code depending on some self-defined conditions. 🡨

* 

Repeatedly output similar pieces of code.

* 

Encapsulate some styles to reuse later.

Formularende

Formularbeginn

Question 2:

**Which are the directives you need for conditions in Sass?**

* 

@if and @when

* 

@if, @elsif, and @else

* 

@if and @else (plus @else if) 🡨

Formularende

Formularbeginn

Question 3:

**Which of the following is a valid condition (assuming that there is a *$client*variable).**

* 

$client = my-favorite-client

* 

$client is my-favorite-client

* 

$client == my-favorite-client 🡨

Formularende

**130.30.while-loops**

Formularbeginn

Question 1:

**What are loops used for in Sass?**

* 

To output different code depending on some condition.

* 

To repeatedly output parts of the code (with slight changes in each iteration). 🡨

* 

To define parts of the code that should not be translated to CSS.

Formularende

Formularbeginn

Question 2:

**What are the three loop directives available in Sass?**

* 

@for, @for-each, and @while

* 

@loop, @each, and @while

* 

@for, @each, and @while 🡨

Formularende

Formularbeginn

Question 3:

**What can you use @each loops for?**

* 

To iterate over lists and maps. 🡨

* 

Only to iterate over ranges of numbers such as 1, 2, 3, 4, 5.

* 

To loop as long as a certain condition is true.

Formularende