



Top 40 SASS & LESS Interview Questions & Answers

1) Explain what is Sass? How it can be used?

Sass stands for Syntactically Awesome Stylesheets and was created by Hampton Catlin. It is an extension of CSS3, adding nested rules, mixins, variables, selector inheritance, etc.

Sass can be used in three ways

- As a command line tool
- As a standalone Ruby module
- As a plugin for any Rack-enabled framework

2) List out the key features for Sass?

Key features for Sass include

- Full CSS3-compatible
- Language extensions such as nesting, variables, and mixins
- Many useful functions for manipulating colors and other values
- Advanced features like control directives for libraries
- Well-formatted, customizable output

3) List out the Data Types that SassScript supports?

SassScript supports seven main data types

1. Numbers (eg; 1,5 ,10px)
2. Strings of texts (g., "foo", 'bar', etc.)
3. Colors (blue, #04a3f9)
4. Booleans (true or false)
5. Nulls (e.g; null)
6. List of values, separated by space or commas (g., 1.5em, Arial, Helvetica etc.)
7. Maps from one value to another (g., (key 1: value1, key 2: Value 2))

4) Explain how to define a variable in Sass?

Variables in Sass begin with a (\$) sign and variable assignment is done with a colon(:).

5) Explain what is the difference between Sass and SCSS?

The difference between Sass and SCSS is that,

- Sass is a CSS pre-processor with syntax advancements and an extension of CSS3
- Sass has two syntax
- The first syntax is “SCSS” and it uses the .scss extension
- The other syntax is indented syntax or just “Sass” and it uses the .sass extension
- While Sass has loose syntax with white space and no semicolons, the SCSS resembles more to CSS

Any CSS valid document can be converted to Sass by simply changing the extension from .CSS to .SCSS.



6) What Selector Nesting in Sass is used for?

In Sass, selector nesting offers a way for stylesheet authors to compute long selectors by nesting shorter selectors within each other.

7) Explain what is a @extend function used for in Sass?

In Sass, the @EXTEND directive provides a simple way to allow a selector to inherit the styles of another one. It aims at providing a way for a selector A to extend the styles from a selector B. When doing so, the selector A will be added to selector B so they both share the same declarations. @EXTEND prevents code bloat by grouping selectors that share the same style into one rule.

8) Explain what is the use of the @IMPORT function in Sass?

The @IMPORT function in Sass

- Extends the CSS import rule by enabling import of SCSS and Sass files
- All imported files are merged into a single outputted CSS file
- Can virtually mix and match any file and be certain of all your styles
- @IMPORT takes a filename to import

9) Why Sass is considered better than LESS?

- Sass allows you to write reusable methods and use logic statements, e., loops, and conditionals
- Sass user can access Compass library and use some awesome features like dynamic sprite map generation, legacy browser hacks and cross-browser support for CSS3 features
- Compass also allows you to add an external framework like Blueprint, Foundation or Bootstrap on top
- In LESS, you can write a basic logic statement using a 'guarded mixin', which is equivalent to Sass if statements
- In LESS, you can loop through numeric values using recursive functions while Sass allows you to iterate any kind of data
- In Sass, you can write your own handy functions

10) Explain what is the use of Mixin function in Sass? What is the meaning of DRY-ing out a mixin?

Mixin allows you to define styles that can be re-used throughout the stylesheet without needing to resort to non-semantic classes like .float-left.

DRY-ing out of a mixing means splitting it into dynamic and static parts. The dynamic mixin is the one that the user actually going to call, and the static mixin is the pieces of information that would otherwise get duplicated.

11) Explain what Sass Maps is and what is the use of Sass Maps?

Sass map is a structured data in a hierarchical way and not just a bunch of variables. It can help in organizing the code. Some great use of Sass are

- It is very useful when dealing with layers of elements in your project
- It can be helpful in color management when there is long list of different color and shade
- Use icon map for various social media icons for example: facebook: '\e607' or twitter: '\e602'
- Unlike other programming libraries, Sass map will consist only of code that is going to be used

12) Explain how Sass comments are different from regular CSS?

Syntax for comments in regular CSS starts with `/*` comments...`*/`, while in SASS there are two type of comment, the single line comments `//` and the multiline CSS comments with `/* */`.

13) Does Sass support inline comments?

Single line comments `//` will be removed by the `.scss` pre-processor and won't appear in your `.css` file

While the comment `/* */` are valid CSS, and will be preserved between the translation from `.scss` to your `.css` file

14) How interpolation is used in Sass?

In Sass, you can define an element in a variable and interpolate it inside the Sass code. It is useful when you keep your modules in separate files.

15) Explain when can you use the %placeholders in Sass?

%placeholders in Sass is useful when you want to write styles that were meant to be extended, but you don't want the base styles to be seen in output CSS styles

16) Is it possible to nest variables within variables in Sass?

Interpolation of variables names is not possible currently in Sass. However, you may use interpolation of placeholders.

17) What are Sass cons and pros?

Pros:

- Sass is easy to learn especially for them who has a background of Python, Ruby or Coffescript and place using functions, writing mixins
- CSS can be easily converted to Sass
- Throughout the project, you don't have to repeat similar CSS statements using `@extend` attribute
- It allows to define variables that are usable throughout the entire project
- It keeps your responsive project more organized

18) Explain what is LESS?

LESS is dynamic style sheet producing language. LESS is a CSS pre-processors and extends CSS with dynamic behaviour. It allows for variables, mixins, operations and functions. LESS runs on server side and client side both.

19) Explain how to create LESS file and where to store it and compile it?

Creating or storing LESS file is similar to creating/storing CSS file. A new LESS file can be created with a .less extension, or you can rename existing .css file to .less file. You can write LESS code with existing CSS code.

The best way of creating it inside ~/content/ or ~/Styles/ folder

20) In what ways LESS can be used?

- Via npm LESS can be used on the command line
- Download as a script file for the browser
- For third party tools, it is used

21) How variable is represented in LESS?



LESS allows variables to be defined. In LESS, the variable is represented as @sing. While, variable assignment is done with a: (colon) sing. The values of the variables are inserted into the CSS output file as well as minified file.

22) Explain how Mixins is useful?

Mixins enable embedding all the properties of a class into another class by including the class name as one of its properties. It is just like variables but for whole classes.

23) Explain how can set code in a watch mode when you run LESS.js in an HTML5 browser?

If you run LESS.js in an HTML5 browser, it will use local storage to cache the generated CSS. However, from the developer point of view they cannot see the changes they made instantly. In order to see your changes instantly, you can load program in development and watch mode by following JavaScript

```
less.env = "development " ;
```

less.watch ();

24) Explain what is the meaning of nesting in LESS programming?

Nesting in LESS is clustering of statements inside other statements, so it forms a group of related code. In other words when we add a code snippet and add another code inside it, then that code snippet is called nesting.

25) Mention what are the color channel functions used in LESS?

- hue
- saturation
- hsvhue
- saturation
- hswalue
- red
- green
- blue
- alpha
- luma
- luminance

26) Explain what is data-uri in LESS?

In CSS, Data URI's is one of the best technique, it allows developers to avoid external image referencing and instead embed them directly into a stylesheet. Data URIs are the excellent way to reduce HTTP requests

27) Explain what “Source Map Less Inline”?

The “ Source Map Less Inline” option indicates that we should include all of the CSS files into the sourcemap. Which means that you only need your map file to get to your original source.

28) Explain what is the use of Extend “all” in LESS?

When you specify all keyword last in an extend argument, it tells LESS to match that selector as part of another selector.

29) Explain what is “StrictImports” in LESS?

The strictImports controls whether the compiler will allow a @import inside of either @media blocks or other selector blocks

30) List out the differences between LESS and Sass?

Each style-sheet language is good in their perspective and use; however there are few differences in their usage.

LESS

- LESS uses JavaScript and processed at client-side
- Variable names are prefaced with the @symbol
- LESS does not inherit multiple selectors with one set of properties
- LESS does not work with “unknown” units neither it returns syntax error notification for incompatible units or maths related syntax error

Sass

- Sass is coded in Ruby and thus processed to server-side
- Variable name are prefaced with \$ symbol
- Sass inherits multiple selectors with one set of properties
- Sass allows you to work with “unknown” units also returns a syntax error notification for incompatible units

31) What are the similarities between LESS and Sass?

Between LESS and Sass the similarities are

- Namespaces
- Color functions
- Mixins and parametric mixins
- Nesting capabilities
- JavaScript evaluations

32) Explain what is the use of &combinator ?

&combinator concatenates nested selector with the parent selector. It is useful for Pseudo classes such as :hover and :focus

33) Explain what is the use of operations in LESS?

Operations can be used for performing functions like

- Simple Mathematical operators: +, -, *, /
- Color functions
- Math functions
- Any size or colour variable can be operated upon

34) Explain what is the use of Escaping?

The use of escaping in LESS

- When you need to output CSS that is not valid CSS syntax
- Proprietary syntax not recognized by LESS
- LESS compiler will throw an error if not used

- Simple prefix with ~ symbol and put in quotes

35) What does LESS elements contains?

Less elements contain commonly used mixins like

- .gradient
- .rounded
- .opacity
- .box-shadow
- .inner-shadow

36) List out alternatives against LESS?

- SASS: Syntactically Awesome Stylesheets
- SCSS: Version.2 of SASS
- Stylus

37) Explain how you can invoke the compiler from the command line?

You can invoke the compiler from the command line in LESS as

```
$ lessc styles.less
```

This will output the compiled CSS to **stdout**; you may then redirect it to a file of your choice

```
$ lessc styles.less > styles.css
```

38) What is the use of e () function?

With the help of e() function you can escape a value so that it passes straight through to the compiled CSS, without being noticed by the LESS compiler.

39) Explain how you can pre-compile LESS into CSS?

To pre-compile LESS into CSS you can use

- Run less.js using Node.js : By using the Node.js JavaScript framework you can run the less.js script outside the browser
- Use lessphp: For the implementation of the LESS compiler written in PHP, lessphp is used
- Use online Compiler: Use online compiler for quick compilation of LESS code without installing a compiler
- Less. app (for Mac users): Less.app is a free tool for Mac users, this tool auto compiles them into CSS files

40) Explain how merge function is used in LESS?

For aggregating values from multiple properties into a space or comma separated list under a single property LESS is used. It is useful for properties such as transform and background

41) How can you create a loop structures in LESS?

A mixin can call itself in LESS. Such recursive mixins, when combined with Pattern matching and Guard Expressions, can be used to create various iterative/loop structures.

42) Why do we need parametric mixins in LESS?

Parametric mixins are same like standard mixins. The only difference is that parametric mixins take parameters like functions in JavaScript. After determining parameters to the mixins, you get more control over mixins.

Cons:

- White space sensitive
- No inline rules

[Guru99](#) Provides [FREE ONLINE TUTORIAL](#) on Various courses like

Java	MIS	MongoDB	BigData	Cassandra
Web Services	SQLite	JSP	Informatica	Accounting
SAP Training	Python	Excel	ASP Net	HBase
Project Management	Test Management	Business Analyst	Ethical Hacking	PMP
Live Project	SoapUI	Photoshop	Manual Testing	Mobile Testing
Selenium	CCNA	AngularJS	NodeJS	PLSQL

**Stay updated with new
courses at Guru99
Join our Newsletter**