

What makes it easier for devs to simplify and design style sheets?	SASS
What is a preprocessor?	SASS
What lets you create variables nest css rules into other, import other sass files, among other things?	Sass
What are SCSS and Sass?	Two available sass syntaxes
What have the .scss extension for Sassy CSS?	SCSS files
Which uses indentation rather than brackets to indicate nesting of selectors, and newlines rather than semicolons to separate properties?	Sass, .sass files
What uses variables?	Sass
What start with \$ followed by variable name?	Sass variables
What allow nesting of CSS rules?	Sass
What helps organize code by placing child style rules within the respective parent elements?	nesting
What's a group of CSS declarations that can be reused throughout the style sheet?	A mixin
What help with rewriting vendor prefixes over and over?	mixins
<pre>@mixin box-shadow(\$x, \$y, \$blur, \$c){   -webkit-box-shadow: \$x, \$y, \$blur, \$c;   -moz-box-shadow: \$x, \$y, \$blur, \$c;   -ms-box-shadow: \$x, \$y, \$blur, \$c;   box-shadow: \$x, \$y, \$blur, \$c; }</pre>	How to write a mixin
What starts with @mixin followed by a custom name?	mixin
What are \$x, \$y, \$blur, \$c in the above example?	Parameters which are optional

What replaces rewriting all the vendor prefixes anytime you need a box-shadow in the example above?	Single line calling the mixin
How to call a mixin?	<code>@include box-shadow(0px, 0px, 4px, #fff);</code>
When to use the <code>@include</code> directive?	Calling a mix-in
What's a useful directive to test for specific case, and works just like JS if statement?	<code>@if</code>
What just like in JS test for more conditions?	<code>@else if</code> and <code>@else</code>
What directive adds styles in a loop, similar to JS for loop?	<code>@for</code>
What is used in two ways: start through end and start to end?	<code>@for</code>
Which excludes the end number?	Start to end
Which includes the end number?	Start through end
What is the syntax to combine a variable (i) with text to make a string?	<code>#{\$i}</code>
What directive uses a starting and ending value to loop a certain number of times?	<code>@for</code>
What does Sass also offer that loops over each item in a list or map?	<code>@each</code> directive
On each iteration, what does the variable get assigned to from the list or map?	Current value
list:	<pre>@each \$color in blue, red, green {   .#{\$color}-text {color: \$color;} }</pre>
map:	<pre>\$colors: (color1: blue, color2: red, color3: green);  @each \$key, \$color in \$colors {   .#{\$color}-text {color: \$color;} }</pre>
What is needed to reference the keys in the map?	<code>\$key</code> variable

If not using the \$key variable in the example above, what will the compiled CSS have?	Color1, color2, color3...
<pre>.blue-text {   color: blue; }  .red-text {   color: red; }  .green-text {   color: green; }</pre>	What the above @each directives convert to
What is similar to JS while loop that creates CSS rules until a condition is met?	@while directive
<pre>\$x: 1; @while \$x &lt; 13 {   .col-#{ \$x } { width: 100%/12 * \$x;}   \$x: \$x + 1; }</pre>	First define variable \$x and set to 1, then use @while to create a grid while \$x is less than 13; increment \$x to prevent infinite loop
What are separate files that hold segments of CSS code?	partials
What are imported and used in other Sass files?	partials
What starts with the underscore character which tells Sass it is a small segment of CSS and not to convert it into a CSS file?	Names for partials
What is used to bring the code in the partial into another Sass file?	@import directive
For example, if all your mixins are saved in a partial named "_mixins.scss", and they are needed in the "main.scss" file	<pre>// In the main.scss file  @import 'mixins'</pre>
What is not needed in the import statement>?	The underscore
What Sass feature makes it easy to borrow the css rules from one element and build upon them in another?	extend
.big-panel{	Using extend

[illegible]



