**1.** Sass (Syntactically Awesome Stylesheets «синтаксически привлекательные таблицы стилей») — препроцессор CSS, который представляет собой метаязык на основе CSS, предназначенный для расширения возможностей и сокращения записи CSS

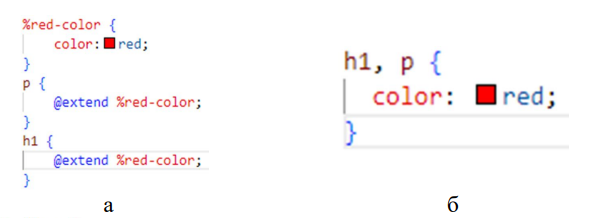
2. Установка Sass

3. Препроцессор  — программа, имеющий собственный синтаксис, который затем компилируется в стандартный CSS-код

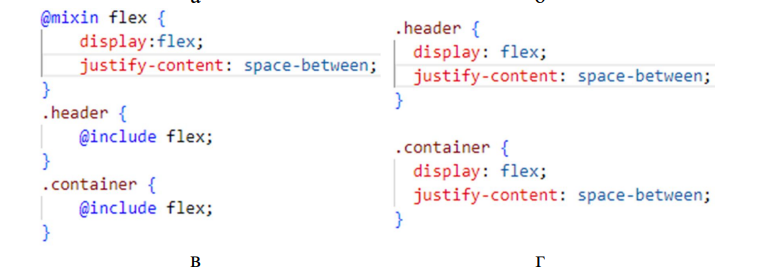
4. $имя переменной: значение; - создание переменной

5. Переменные могут быть глобальными, если создана вне блоков объявлений, и локальными, если будет создана внутри блока объявления. Переменную можно подставить в комментарий, строку, селектор используя вид: #{$имя переменной}

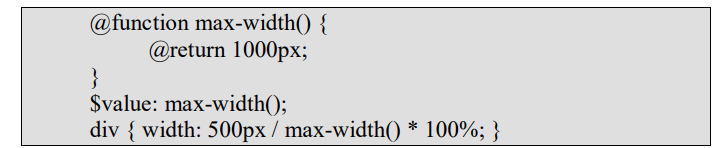
6. Чтобы скомпилировать определенный фрагмент используется директива @import. Все импортированные scss и sass файлы могут быть объединены в одном результирующем css файле.

7. Вместо селектора можно указать %название. Такой вид называется шаблонным селектором или селектором-заполнителем. Чтобы вставить шаблон, нужно воспользоваться директивой @extend. Шаблонный селектор можно указать в составе обычных селекторов. 

8. Миксины позволяют вставить одно или несколько правил стиля вместо директивы @include c названием миксина сколько угодно раз. После директивы указывается название миксина и внутри круглых скобок могут указываться параметры, которые являются локальными переменными.

Миксины в отличие от шаблонов могут содержать параметры, что позволяет настраивать значения стиля или генерировать. 

9. . Для создания миксина используется директива @mixin

10. В Sass можно создавать фрагменты кода, которые можно вызывать неоднократно из любого места программы. Эти фрагменты кода называются функциями. 

11. Операторы логические, арифметические, условные операторы(@if) и цикла(@for).

13. Производить арифметические операции позволяют операторы сложения (+), вычитания (–), унарный минус, умножение (\*), деления (/), остатка от деления (%). Чтобы выполнить деление необходимо либо заключить выражение в круглые скобки, либо сохранить результат в переменной, либо возвратить результат из функции. Приоритет выполнения операции умножения выше, затем деление и сложение. Изменить порядок вычисления выражения можно с помощью круглых скобок. Операции заключенные в них имеют наивысший приоритет.

14. К операторам сравнения относятся равно (==), не равно (!=), меньше (), меньше или равно (<=), больше или равно (=>). Два значения равны, если имеют одинаковый тип данных и значение. Числа равны, если равны их значения и единицы измерения.

15. if for