



tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time. It is more stable and powerful CSS extension language that describes the style of document structurally. This tutorial covers the basics of SASS.

Audience

This tutorial will help both students as well as professionals, who want to make their websites or personal blogs more attractive.

Prerequisites

Before you proceed with this tutorial, we assume that you know:

- Basic word processing using any text editor.
- How to create directories and files.
- How to navigate through different directories.
- Internet browsing using popular browsers like Internet Explorer or Firefox.
- How to develop simple Web Pages using HTML or XHTML.

If you are new to HTML and XHTML, then we would suggest you to go through our HTML Tutorial or XHTML Tutorial first.

Copyright & Disclaimer

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents	ii
1. SASS – Overview	1
What is SASS?	1
Why to Use SASS?.....	1
Features of SASS.....	1
Advantages of SASS	2
Disadvantages of SASS.....	2
2. SASS – Installation	3
Installation of Ruby.....	3
3. SASS – Syntax.....	11
SASS Indented Syntax	11
Syntax Differences of SASS	12
Multiline Selectors.....	12
Comments	14
@import	14
Mixin Directives	14
Deprecated Syntax	14
4. SAAS – Application.....	16
Rack/Rails/Merb Plugin	16
Caching	17
Options	17
Syntax Selection	19
Encodings.....	19
5. SASS – CSS Extensions	20
SASS – Nested Rules	20
SASS – Referencing Parent Selectors.....	23
SASS – Nested Properties	25
SASS – Placeholder Selectors.....	26
6. SASS – Comments	29
Sass – Interpolation in Multiline Comments	31
7. SASS – Script	33
SASS – Interactive Shell	34
SASS – Variables	34
SASS – Data Types	36
Strings.....	36
Lists.....	38
Maps.....	39
Colors.....	39
SASS – Operations	39
SASS – Number Operations	40

SASS – Color Operations	42
SASS – String Operations	44
SASS – Boolean Operations	45
SASS – Parentheses	47
SASS – Functions.....	49
SASS – Interpolation	51
SASS – & in SassScript.....	53
SASS – Variable Defaults.....	55
8. SASS – @-Rules and Directives	57
Sass – Import Directives	57
Sass – Media Directives	61
Sass – Extend Directives	63
Sass – At-root Directives.....	66
Sass – Debug Directives	69
Sass – Warn Directives	70
Sass – Error Directives	71
9. SASS – Control Directives & Expressions	73
SASS – if() Function	73
SASS – @if Directive	75
SASS – @if Directive	75
SASS – @else if Directive	77
SASS – @for Directive.....	79
SASS – through Keyword	79
SASS – to Keyword.....	81
SASS – @each Directive.....	83
SASS – @each Directive.....	84
SASS – @each Multiple Assignment	86
SASS – @each Multiple Assignment with Maps	88
SASS – @while Directive.....	90
10. SASS – Mixin Directives.....	92
Sass – Defining a Mixin	92
Sass – Including a Mixin	94
Sass – Mixin Arguments.....	96
Sass – Passing Content Blocks to a Mixin	99
11. SASS – Function Directives	102
Naming Conventions	103
12. SASS – Output Style	104
:nested.....	104
:expanded.....	104
:compact.....	105
:compressed	105
13. Extending SASS	106
Defining Custom SASS Functions	106
Cache Stores	106
Custom Importers.....	106

1. SASS – Overview

What is SASS?

SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time. It is more stable and powerful CSS extension language that describes the style of a document cleanly and structurally.

It was initially designed by **Hampton Catlin** and developed by **Natalie Weizenbaum** in 2006. Later, **Weizenbaum** and **Chris Eppstein** used its initial version to extend the Sass with SassScript.

Why to Use SASS?

- It is a pre-processing language, which provides indented syntax (its own syntax) for CSS.
- It provides some features, which are used for creating stylesheets that allows writing code more efficiently and is easy to maintain.
- It is a super set of CSS, which means it contains all the features of CSS and is an open source pre-processor, coded in **Ruby**.
- It provides the document style in a good, structured format than flat CSS. It uses re-usable methods, logic statements and some of the built-in functions such as color manipulation, mathematics and parameter lists.

Features of SASS

- It is more stable, powerful, and compatible with versions of CSS.
- It is a super set of CSS and is based on JavaScript.
- It is known as syntactic sugar for CSS, which means it makes easier way for user to read or express the things more clearly.
- It uses its own syntax and compiles to readable CSS.
- You can easily write CSS in less code within less time.
- It is an open source pre-processor, which is interpreted into CSS.

Advantages of SASS

- It allows writing clean CSS in a programming construct.

- It helps in writing CSS quickly.
- It is a superset of CSS, which helps designers and developers work more efficiently and quickly.
- As Sass is compatible with all versions of CSS, we can use any available CSS libraries.
- It is possible to use nested syntax and useful functions such as color manipulation, mathematics and other values.

Disadvantages of SASS

- It takes time for a developer to learn new features present in this pre-processor.
- If many people are working on the same site, then should use the same preprocessor. Some people use Sass and some people use CSS to edit the files directly. Therefore, it becomes difficult to work on the site.
- There are chances of losing benefits of browser's built-in element inspector.

2. SASS – Installation

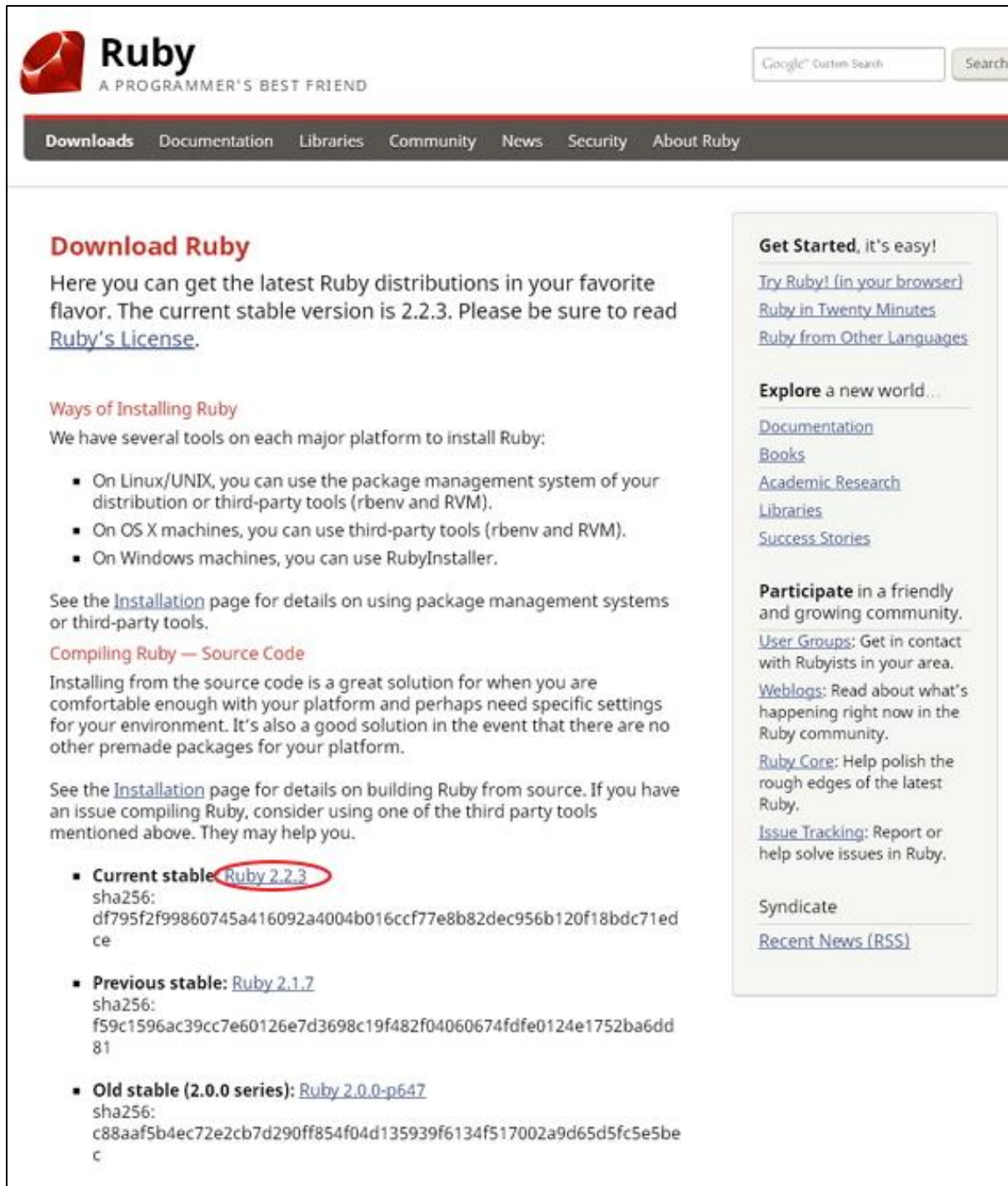
In this chapter, we will learn the step-by-step procedure to install Ruby, which is used for executing the SASS files.

System Requirements for SASS

- **Operating System :** Cross-platform
- **Browser Support :** IE (Internet Explorer 8+), Firefox, Google Chrome, Safari, Opera
- **Programming Language:** Ruby

Installation of Ruby

Step(1): Open the link <https://www.ruby-lang.org/en/downloads/>, you will see a screen as shown below:



The screenshot shows the Ruby website homepage. At the top left is the Ruby logo with the text "A PROGRAMMER'S BEST FRIEND". To the right is a Google Custom Search bar. Below the header is a navigation menu with links: Downloads, Documentation, Libraries, Community, News, Security, and About Ruby. The main content area is titled "Download Ruby" and contains the following text: "Here you can get the latest Ruby distributions in your favorite flavor. The current stable version is 2.2.3. Please be sure to read [Ruby's License](#)." Below this is a section "Ways of Installing Ruby" which states: "We have several tools on each major platform to install Ruby:" followed by a list:

- On Linux/UNIX, you can use the package management system of your distribution or third-party tools (rbenv and RVM).
- On OS X machines, you can use third-party tools (rbenv and RVM).
- On Windows machines, you can use RubyInstaller.

This is followed by a paragraph about the [Installation](#) page and a section "Compiling Ruby — Source Code" which explains installing from source code. Below this is another paragraph about the [Installation](#) page for building from source. The final section lists three stable versions:

- Current stable:** [Ruby 2.2.3](#) (circled in red). It provides sha256 hashes: df795f2f99860745a416092a4004b016ccf77e8b82dec956b120f18bdc71edce.
- Previous stable:** [Ruby 2.1.7](#). It provides sha256 hashes: f59c1596ac39cc7e60126e7d3698c19f482f04060674fdfe0124e1752ba6dd81.
- Old stable (2.0.0 series):** [Ruby 2.0.0-p647](#). It provides sha256 hashes: c88aaf5b4ec72e2cb7d290ff854f04d135939f6134f517002a9d65d5fc5e5bec.

On the right side of the page, there is a sidebar with sections: "Get Started, it's easy!" with links to [Try Ruby! \(in your browser\)](#), [Ruby in Twenty Minutes](#), and [Ruby from Other Languages](#); "Explore a new world..." with links to [Documentation](#), [Books](#), [Academic Research](#), [Libraries](#), and [Success Stories](#); "Participate in a friendly and growing community." with links to [User Groups](#), [Weblogs](#), [Ruby Core](#), and [Issue Tracking](#); and a "Syndicate" section with a link to [Recent News \(RSS\)](#).

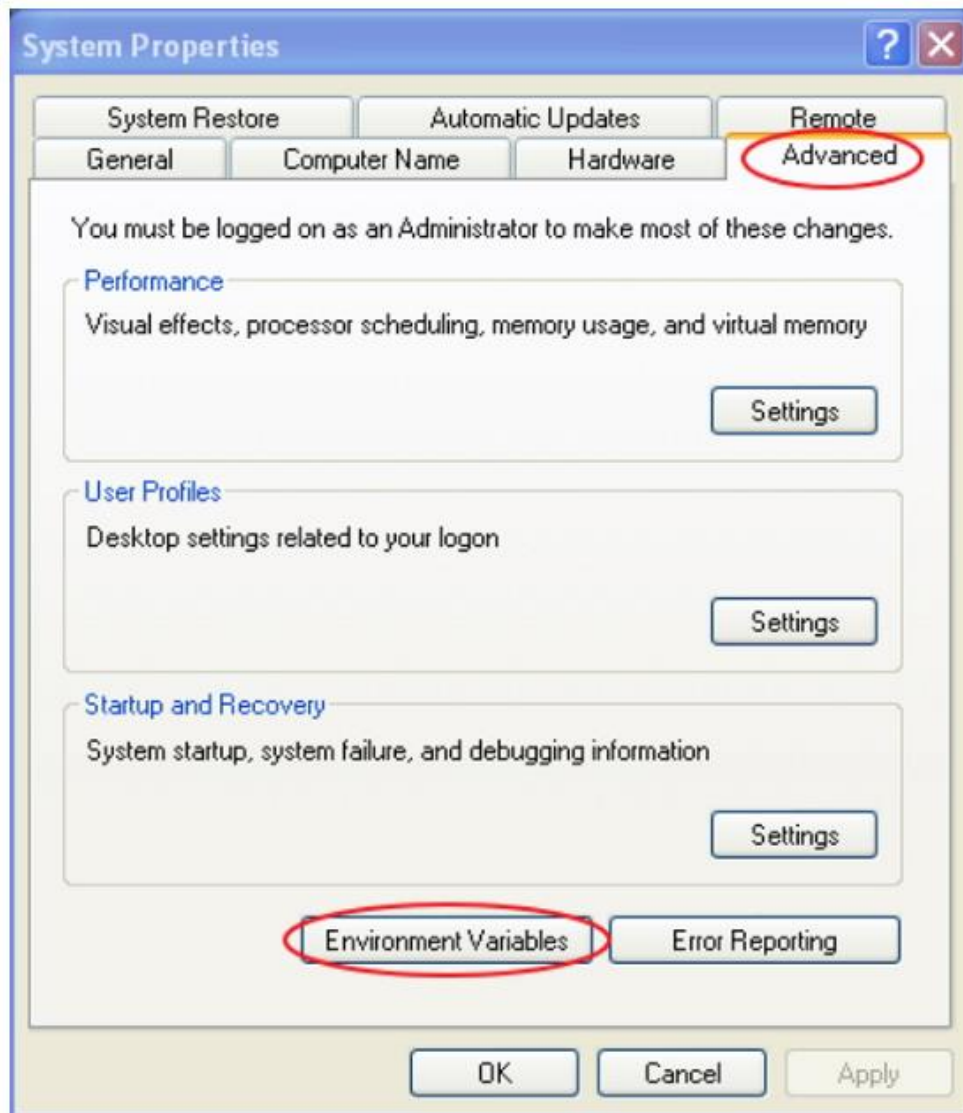
Download the *Current stable* version of the zip file.

Step(2): Next, run the setup to install **Ruby** on the System.

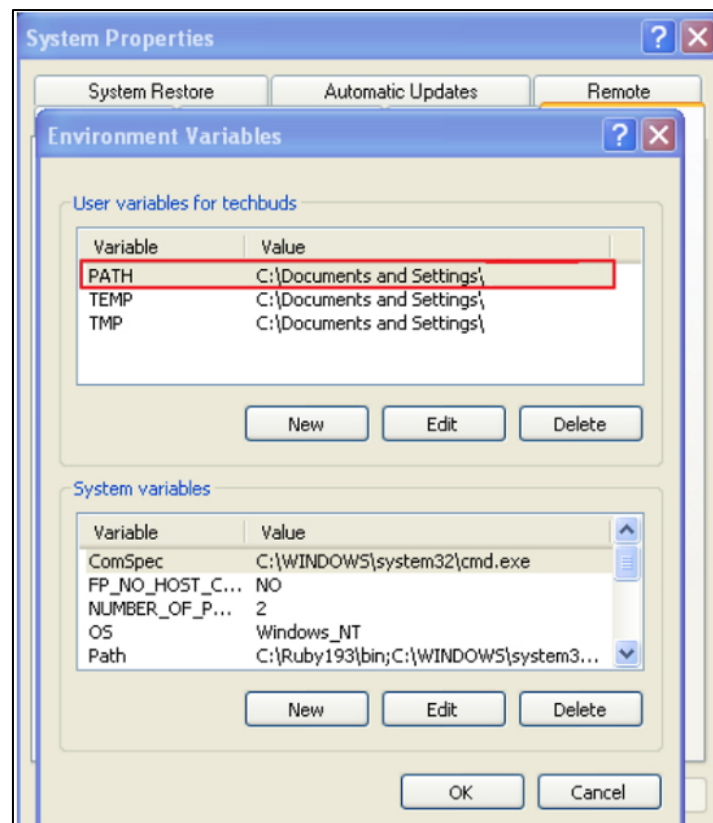
Step(3): Next, add Ruby bin folder to your *PATH User Variable* and *System Variable* to work with gem command.

Path User Variable:

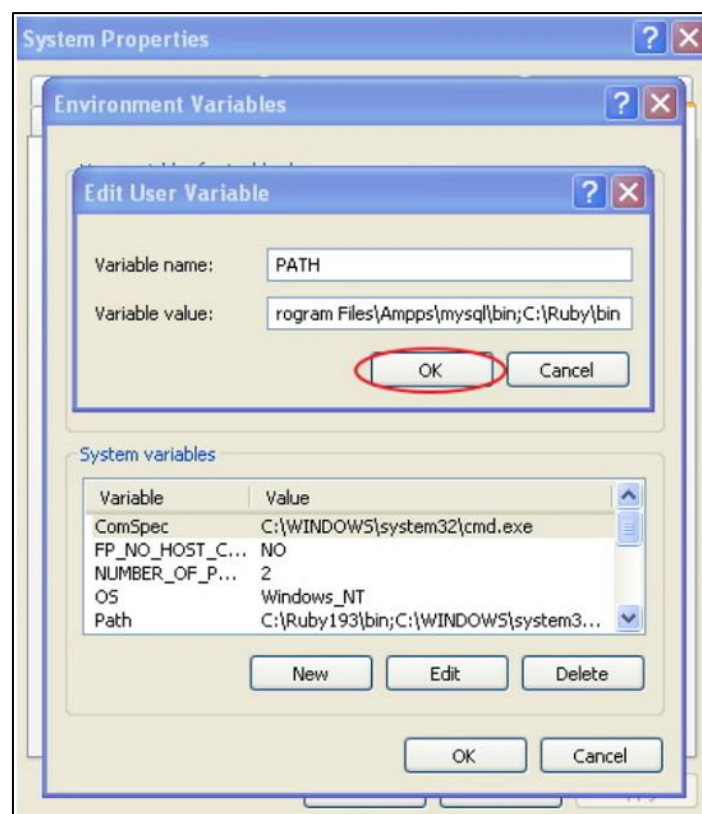
- Right Click the **My Computer** icon.
- Select **Properties**.
- Next, click the **Advanced** tab and click **Environment Variables**.



In the *Environment Variables* window, double click the *PATH* as shown in the screenshot given below-



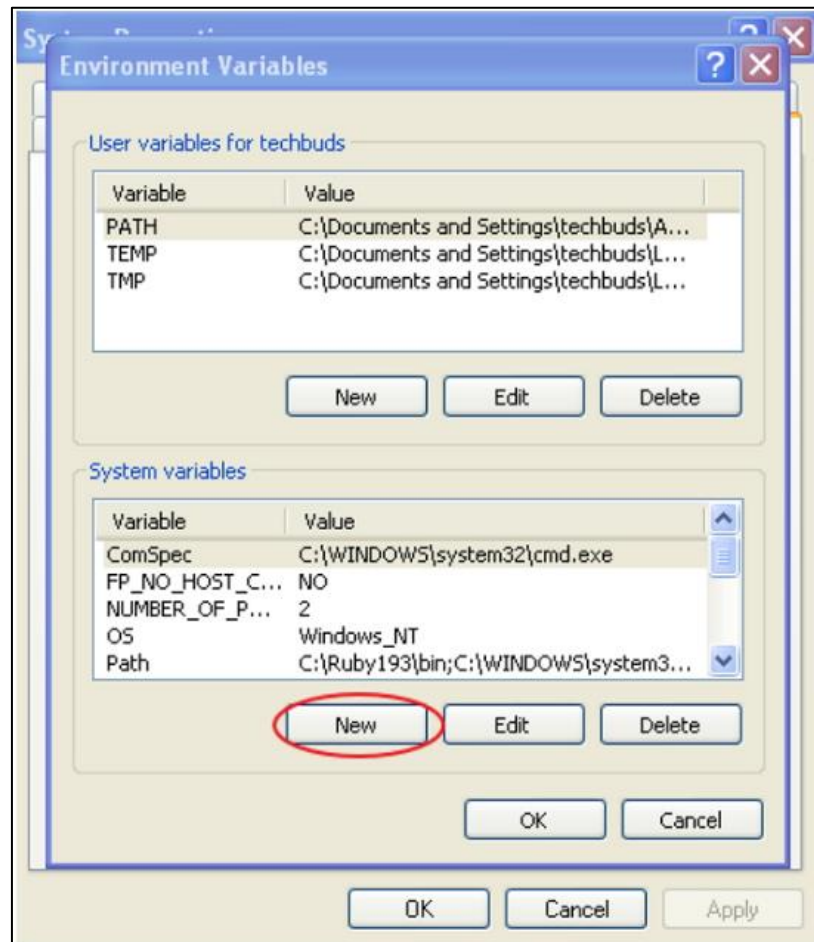
You will get an *Edit User Variable* box as shown. Add ruby bin folder path in the *Variable value* field as **C:\Ruby\bin**. If path is already set for other files, then put semicolon after that and add the Ruby folder path as shown below.



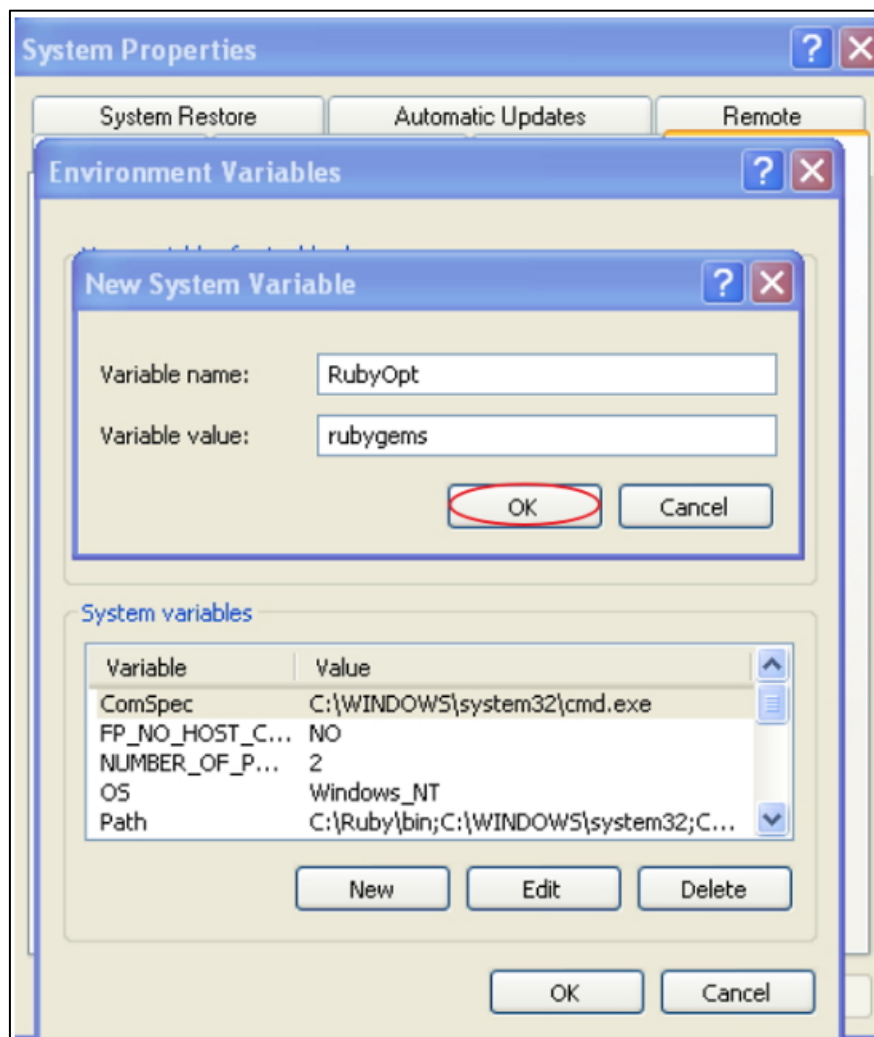
- Click the **OK** button.

System Variable:

- Click the **New** button.



Next, the **New System Variable** block is displayed as shown below.

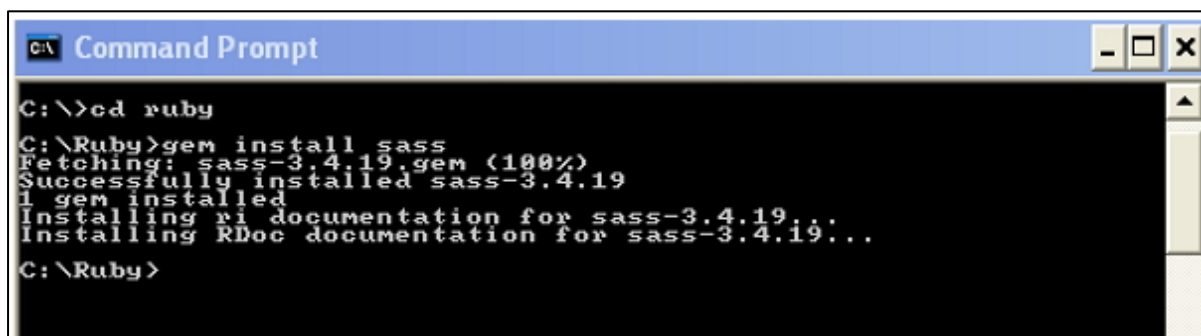


- Enter **RubyOpt** in the *Variable name* field and **rubygems** in the *Variable value* field. After writing the *Variable name* and *value*, click the **OK** button.

Step(4): Open the command prompt in your system and enter the following line-

```
gem install sass
```

Step(5): Next, you will see the following screen after installing SASS successfully.



Example

Following is a simple example of SASS.

```
<html>
<head>
  <title> Import example of sass</title>
  <link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
  <h1>Simple Example</h1>
  <h3>Welcome to Tutorialspoint</h3>
</body>
</html>
```

Now, we will create file as *style.scss*, which is quite similar to CSS and the only one difference is that it will be saved with .scss extension. Both, .htm and .scss files should be created inside the folder **ruby**. You can save your .scss file in the folder **ruby\lib\sass** (Before this process, create a folder as **sass** in lib directory).

```
h1{
  color: #AF80ED;
}
h3{
  color: #DE5E85;
}
```

You can tell SASS to watch the file and update the CSS whenever SASS file changes by using the following command:

```
sass --watch C:\ruby\lib\sass\style.scss:style.css
```

```
C:\>cd ruby
C:\Ruby>sass --watch C:\ruby\lib\sass\style.scss:style.css
>>> Sass is watching for changes. Press Ctrl-C to stop.
      write style.css
      write style.css.map
```

When you run the above command, it will create the *style.css* file automatically. Whenever you change the SCSS file, the *style.css* file will be updated automatically.

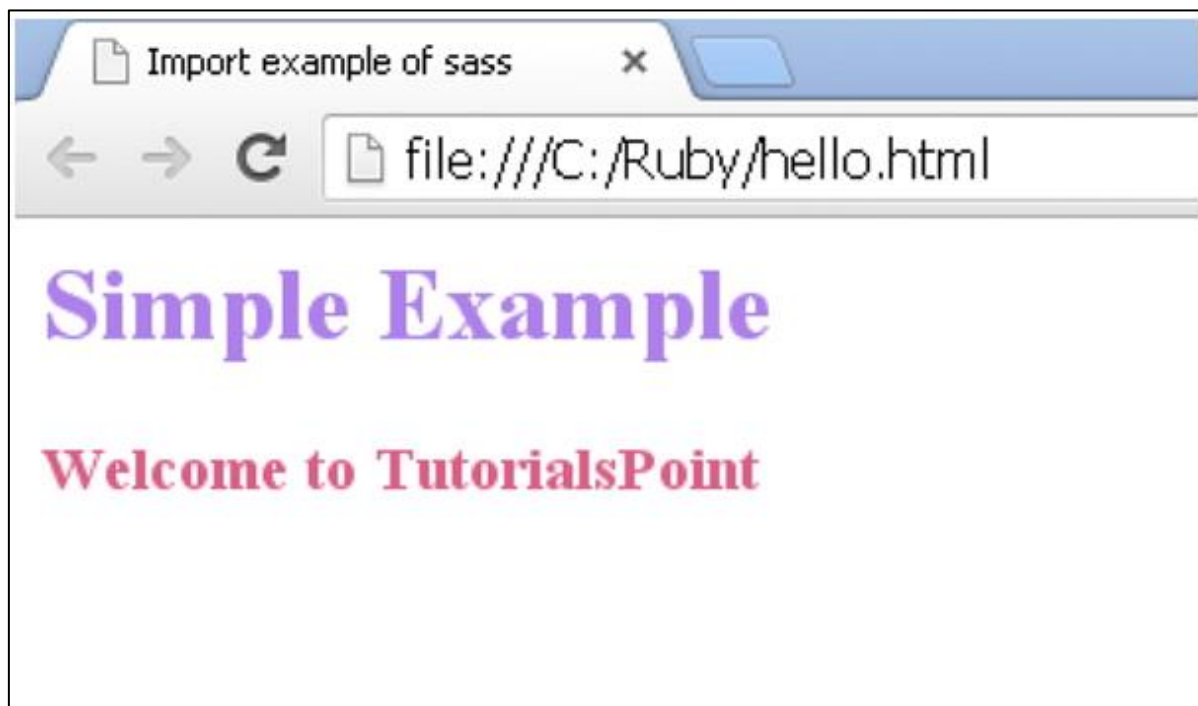
The *style.css* file will have the following code when you run the above given command:

style.css

```
h1 {  
  color: #AF80ED;  
}  
h3 {  
  color: #DE5E85;  
}
```

Let us carry out the following steps to see how the above given code works:

- Save the above given code in **hello.html** file.
- Open this HTML file in a browser.



End of ebook preview

If you liked what you saw...

Buy it from our store @ **<https://store.tutorialspoint.com>**