About the Tutorial

CoffeeScript is a lightweight language which transcompiles into JavaScript. It provides better syntax avoiding the quirky parts of JavaScript, still retaining the flexibility and beauty of the language.

Audience

This tutorial has been prepared for beginners to help them understand the basic functionality of CoffeeScript to build dynamic webpages and web applications.

Prerequisites

For this tutorial, it is assumed that the readers have a prior knowledge of HTML coding and JavaScript. It would help if the reader has some prior exposure to object-oriented programming concepts and a general idea on creating online applications.

Execute CoffeeScript Online

For most of the examples given in this tutorial, you will find Try it option, so just make use of this option to transcompile your CoffeeScript programs to JavaScript programs on the spot and enjoy your learning.

Try the following example using the Try it option available at the top right corner of the below sample code box −

```javascript
console.log "Hello Welcome to Tutorials point"
```

Copyright & Disclaimer

© Copyright 2016 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
- Audience
- Prerequisites
- Execute CoffeeScript Online
- Copyright & Disclaimer
- Table of Contents

## 1. COFFEESCRIPT – OVERVIEW
   - What is CoffeeScript ?
   - History of CoffeeScript
   - Limitations of CoffeeScript
   - TutorialsPoint’s CoffeeScript IDE

## 2. COFFEESCRIPT – ENVIRONMENT
   - Node.js
   - Installing Node.js
   - Installing CoffeeScript

## 3. COFFEESCRIPT – COMMAND-LINE UTILITY
   - Compiling the CoffeeScript Code
   - Executing the CoffeeScript code
   - Watch and Compile
   - Setting the Output Directory
   - Print the Compiled JavaScript
   - The REPL (Read Evaluate Print Loop)
   - Running CoffeeScript through Browser

## 4. COFFEESCRIPT – SYNTAX
   - CoffeeScript Statements
CoffeeScript

CoffeeScript Variables (No var Keyword) ................................................................. 15
No Parentheses ......................................................................................................... 15
No Curly Braces ....................................................................................................... 16
CoffeeScript Comments ......................................................................................... 16
CoffeeScript Reserved keywords ........................................................................... 17

5. COFFEESCRIPT — DATA TYPES ....................................................................... 18
CoffeeScript Data Types ......................................................................................... 18

6. COFFEESCRIPT — VARIABLES ....................................................................... 19
CoffeeScript Variables ......................................................................................... 19
Variable Scope ....................................................................................................... 19
The Problem with Variables in JavaScript ......................................................... 20
Variable Scope in CoffeeScript ........................................................................... 20
CoffeeScript Variable Names (Literals) ................................................................. 21

7. COFFEESCRIPT — OPERATORS ...................................................................... 22
CoffeeScript Aliases ............................................................................................. 22
Arithmetic Operators .............................................................................................. 22
Comparison Operators .......................................................................................... 25
Logical Operators .................................................................................................. 30
Bitwise Operators .................................................................................................... 34
Assignment Operators ............................................................................................ 37
Equality Operator in CoffeeScript ........................................................................ 40
The existential Operator ......................................................................................... 41
Chained Comparisons ......................................................................................... 42
CoffeeScript Aliases ............................................................................................. 42

8. COFFEESCRIPT — CONDITIONALS ................................................................. 46
if Statement ............................................................................................................ 47
if...else Statement ................................................................................................. 49
### CoffeeScript

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>unless Statement</td>
<td>50</td>
</tr>
<tr>
<td>unless...else Statement</td>
<td>52</td>
</tr>
<tr>
<td>switch Statement</td>
<td>54</td>
</tr>
<tr>
<td>Switch Statement in CoffeeScript</td>
<td>54</td>
</tr>
<tr>
<td>Multiple values for when clause</td>
<td>57</td>
</tr>
<tr>
<td>The then Keyword in CoffeeScript</td>
<td>58</td>
</tr>
<tr>
<td>if...then Statement</td>
<td>59</td>
</tr>
<tr>
<td>if...then...else Statement</td>
<td>60</td>
</tr>
<tr>
<td>unless...then Statement</td>
<td>61</td>
</tr>
<tr>
<td>unless-then...else Statement</td>
<td>62</td>
</tr>
<tr>
<td>postfix if and postfix unless Statements</td>
<td>63</td>
</tr>
<tr>
<td><strong>9. COFFEESCRIPT – LOOPS</strong></td>
<td>65</td>
</tr>
<tr>
<td>The while loop in CoffeeScript</td>
<td>65</td>
</tr>
<tr>
<td>Variants of while</td>
<td>67</td>
</tr>
<tr>
<td>The until Variant of while</td>
<td>67</td>
</tr>
<tr>
<td>The loop Variant of while</td>
<td>69</td>
</tr>
<tr>
<td><strong>10. COFFEESCRIPT – COMPREHENSIONS</strong></td>
<td>72</td>
</tr>
<tr>
<td>for...in Comprehensions</td>
<td>72</td>
</tr>
<tr>
<td>for...of Comprehensions</td>
<td>73</td>
</tr>
<tr>
<td>list Comprehensions</td>
<td>74</td>
</tr>
<tr>
<td>Index of comprehensions</td>
<td>77</td>
</tr>
<tr>
<td>Postfix form of comprehensions</td>
<td>78</td>
</tr>
<tr>
<td>Postfix for...in comprehension</td>
<td>78</td>
</tr>
<tr>
<td>Assigning to a variable</td>
<td>80</td>
</tr>
<tr>
<td>The by keyword</td>
<td>81</td>
</tr>
<tr>
<td><strong>11. COFFEESCRIPT – FUNCTIONS.</strong></td>
<td>83</td>
</tr>
<tr>
<td>Functions in CoffeeScript</td>
<td>83</td>
</tr>
</tbody>
</table>
Defining a Function .............................................................................................................. 83
Multi-line Functions ............................................................................................................. 84
Functions with Arguments .................................................................................................... 85
Invoking a Function ................................................................................................................ 85
Invoking Functions with Arguments ..................................................................................... 86
Default Arguments ................................................................................................................ 87

12. COFFEESCRIPT — STRINGS ...................................................................................... 90
String Concatenation ............................................................................................................ 90
String Interpolation .............................................................................................................. 91
JavaScript String Object ..................................................................................................... 92
charAt method .................................................................................................................... 93
charCodeAt Method ............................................................................................................ 95
concat() Method ................................................................................................................ 96
indexOf() Method ............................................................................................................... 98
lastIndexOf() Method ...................................................................................................... 99
localeCompare() Method ............................................................................................... 100
match() Method ............................................................................................................... 104
search() Method .............................................................................................................. 105
slice() Method ................................................................................................................ 107
split() Method ................................................................................................................ 108
substr() Method ............................................................................................................... 109
toLocaleLowerCase() method ....................................................................................... 111
toLocaleUpperCase() method ....................................................................................... 112
toLowerCase() method ................................................................................................... 113
toUpperCase() method .................................................................................................. 114

13. COFFEESCRIPT — ARRAYS ..................................................................................... 116
New line instead of comma .............................................................................................. 116
## 14. COFFEESCRIPT – OBJECTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indentations instead of curly braces</td>
<td>120</td>
</tr>
<tr>
<td>Nested objects</td>
<td>120</td>
</tr>
<tr>
<td>Comprehensions over objects</td>
<td>121</td>
</tr>
<tr>
<td>Arrays of Objects</td>
<td>122</td>
</tr>
<tr>
<td>Reserved Keywords</td>
<td>124</td>
</tr>
</tbody>
</table>

## 15. COFFEESCRIPT – RANGES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excluding the end Value</td>
<td>127</td>
</tr>
<tr>
<td>Using Ranges with Variables</td>
<td>128</td>
</tr>
<tr>
<td>Ranges with Arrays</td>
<td>129</td>
</tr>
<tr>
<td>Ranges with Strings</td>
<td>131</td>
</tr>
<tr>
<td>Comprehensions over Ranges</td>
<td>132</td>
</tr>
</tbody>
</table>

## 16. COFFEESCRIPT – SPLAT

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calling Functions using Splats</td>
<td>136</td>
</tr>
<tr>
<td>Splats with a Tailing Argument</td>
<td>139</td>
</tr>
<tr>
<td>Comprehensions with Splats</td>
<td>141</td>
</tr>
</tbody>
</table>

## 17. COFFEESCRIPT – DATE

<table>
<thead>
<tr>
<th>Method</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>getDate() Method</td>
<td>146</td>
</tr>
<tr>
<td>getDay() Method</td>
<td>147</td>
</tr>
<tr>
<td>getFullYear() Method</td>
<td>148</td>
</tr>
<tr>
<td>getHours() Method</td>
<td>149</td>
</tr>
<tr>
<td>getMilliseconds() Method</td>
<td>150</td>
</tr>
<tr>
<td>getMinutes() Method</td>
<td>151</td>
</tr>
<tr>
<td>getMonth() Method</td>
<td>152</td>
</tr>
<tr>
<td>getUTCSeconds() Method</td>
<td>153</td>
</tr>
<tr>
<td>getTime() Method</td>
<td>154</td>
</tr>
</tbody>
</table>
getTimezoneOffset() Method ........................................................................................................ 155
getUTCDate() Method .................................................................................................................. 156
getUTCDay() method .................................................................................................................... 157
getUTCFullYear() Method ............................................................................................................. 158
getUTCHours() Method ................................................................................................................ 159
getUTCMilliseconds() Method .................................................................................................... 160
getUTCMinutes() Method .............................................................................................................. 161
getUTCMonth() Method ................................................................................................................ 162
getUTCSecs() Method ................................................................................................................... 163
getYear() Method ........................................................................................................................ 164
setDate() Method ........................................................................................................................ 165
setFullYear() Method .................................................................................................................. 166
setHours() Method ....................................................................................................................... 167
setMilliseconds() Method ............................................................................................................ 168
setMinutes() Method .................................................................................................................... 169
setMonth() Method ...................................................................................................................... 170
setSeconds() Method .................................................................................................................... 171
setTime() Method ........................................................................................................................ 173
setUTCDate() Method ................................................................................................................... 174
setUTCFullYear() Method ............................................................................................................. 175
setUTCHours() Method .................................................................................................................. 176
setUTCMilliseconds() Method .................................................................................................... 178
setUTCMinutes() Method .............................................................................................................. 179
setUTCMonth() Method ................................................................................................................ 180
setUTCSecs() Method ................................................................................................................... 181
setYear() Method ........................................................................................................................ 183
toDateString() Method .................................................................................................................. 184
toLocaleDateString() Method ....................................................................................................... 185
CoffeeScript

toLocaleString() Method ........................................................................................................... 186
toLocaleTimeString() Method .................................................................................................... 187
toTimeString() Method ............................................................................................................... 188
toTimeString Method .................................................................................................................. 189

18. COFFEESCRIP – MATH........................................................................................................ 191

Math Methods .......................................................................................................................... 193
abs() Method ............................................................................................................................. 194
acos() Method ............................................................................................................................ 196
asin() Method ............................................................................................................................ 197
atan() Method ............................................................................................................................ 198
atan2() Method .......................................................................................................................... 200
ceil() Method ............................................................................................................................ 201
cos() Method ............................................................................................................................... 203
exp() Method ............................................................................................................................... 204
floor() Method ............................................................................................................................ 206
log() Method ................................................................................................................................ 207
max() Method ................................................................................................................................ 209
min() Method ................................................................................................................................ 210
pow() Method ................................................................................................................................ 212
random() Method ....................................................................................................................... 213
round() Method ............................................................................................................................ 215
sin() Method ................................................................................................................................ 216
sqrt() Method ................................................................................................................................ 218
tan() Method ................................................................................................................................ 219

19. COFFEEESCRIP – EXCEPTION HANDLING................................................................. 222

Exceptions in CoffeeScript ............................................................................................................ 222
The finally block ............................................................................................................................ 224
Connecting to Database ................................................................. 256
Creating a Table ........................................................................ 257
Inserting / Creating Data ............................................................... 257
Reading / Retrieving Data ............................................................. 258
Updating Data ............................................................................. 259
Deleting Data .............................................................................. 260
At present, JavaScript is the fastest mainstream dynamic language available, and it is known as the *lingua franca* of the web. It is developed by Brendan Eich in the year of 1995 in 10 days.

Because of its effective features, JavaScript became popular and went global quickly. It was there in lab for a very less time, which was not enough to polish the language. May be for this reason, inspite of its good parts, JavaScript has a bunch of design errors and it bagged a bad reputation of being a quirky language.

**What is CoffeeScript ?**

CoffeeScript is a lightweight language based on Ruby and Python which [transcompiles](https://en.wikipedia.org/wiki/Compiled_language) (compiles from one source language to another) into JavaScript. It provides better syntax avoiding the quirky parts of JavaScript, still retaining the flexibility and beauty of the language.

**Advantages of CoffeeScript**

Following are the advantages of CoffeeScript –

- **Easily understandable** – CoffeeScript is a shorthand form of JavaScript, its syntax is pretty simple compared to JavaScript. Using CoffeeScript, we can write clean, clear, and easily understandable codes.

- **Write less do more** – For a huge code in JavaScript, we need comparatively very less number of lines of CoffeeScript.

- **Reliable** – CoffeeScript is a safe and reliable programming language to write dynamic programs.

- **Readable and maintainable** – CoffeeScript provides aliases for most of the operators which makes the code readable. It is also easy to maintain the programs written in CoffeeScript.

- **Class-based inheritance** – JavaScript does not have classes. Instead of them, it provides powerful but confusing prototypes. Unlike JavaScript, we can create classes and inherit them in CoffeeScript. In addition to this, it also provides instance and static properties as well as mixins. It uses JavaScript’s native prototype to create classes.

- **No var keyword** – There is no need to use the `var` keyword to create a variable in CoffeeScript, thus we can avoid the accidental or unwanted scope deceleration.

- **Avoids problematic symbols** – There is no need to use the problematic semicolons and parenthesis in CoffeeScript. Instead of curly braces, we can use whitespaces to differentiate the block codes like functions, loops, etc.
• **Extensive library support** – In CoffeeScript, we can use the libraries of JavaScript and vice versa. Therefore, we have access to a rich set of libraries while working with CoffeeScript.

## History of CoffeeScript

- CoffeeScript is developed by Jeremy Ashkenas. It was first committed in Git On December 13, 2009.
- Originally the compiler of the CoffeeScript was written in Ruby language.
- In March 2010, the CoffeeScript compiler was replaced; this time instead of Ruby, they used CoffeeScript itself.
- And in the same year, CoffeeScript 1.0 was released and at the time of release, it was one of the most wanted projects of the Git hub.

## Limitations of CoffeeScript

**Sensitive to whitespaces** – CoffeeScript is very sensitive to whitespaces, so programmers need to be very careful while providing indentations. If we do not maintain proper indentation, the entire code may go wrong.

## TutorialsPoint's CoffeeScript IDE

You can compile CoffeeScript files using Tutorials Point's CoffeeScript compiler provided in our Coding Ground section (http://www.tutorialspoint.com/codingground.htm). Follow the steps given below to use our CoffeeScript compiler.

**Step 1**

Click on the following link [www.tutorialspoint.com](http://www.tutorialspoint.com). You will be directed to the homepage of our website.

**Step 2**

Click on the button named **CODING GROUND** that is located at the top right corner of the homepage as highlighted in the snapshot given below.
Step 3
This will lead to our **CODING GROUND** section which provides online terminals and IDEs for about 135 programming languages. Open CoffeeScript IDE in the Online IDEs section which is shown in the following snapshot.
Step 4
If you paste your CoffeeScript code in **main.coffee** (You can change the file name) and click the **Preview** button, then you can see the compiled JavaScript in the console as shown in the following snapshot.
The Compiler of the latest versions of CoffeeScript is written in CoffeeScript itself. To run CoffeeScript files in your system without a browser, you need a JavaScript runtime.

**Node.js**

Node.js is a JavaScript framework which is used to develop network server applications. It also acts as a bridge between JavaScript and the Operating System.

The command-line version of CoffeeScript is distributed as a Node.js package. Therefore, to install CoffeeScript (command-line) in your system, you first need install node.js.

**Installing Node.js**

Here are the steps to download and install Node.js in your system.

**Step 1**

Visit the nodejs homepage and download its stable version for windows by clicking on the button highlighted in the snapshot given below.
Step 2

On clicking, a .msc file named `node-v5.50-x64` will be downloaded into your system, run the downloaded file to start the Node.js set-up. Here is the snapshot of the Welcome page of No.js set-up wizard.
Step 3
Click on the Next button in the Welcome page of the Node.js set-up wizard which will lead you to the End-user License Agreement page. Accept the license agreement and click on the Next button as shown below.
Step 4
On the next page, you need to set the destination folder to the path where you want to install Node.js. Change the path to the required folder and click on the Next button.

![Node.js Setup](image)

**Step 5**
In the **Custom setup** page, select the Node.js runtime to install node.exe file and click Next.
Step 6
Finally, click on the Install button which will start the Node.js installation.
Click on the Finish button of the Node.js set-up wizard as shown below to complete the Node.js installation.
Installing CoffeeScript

Node.js provides you a command prompt (Node.js command prompt). You can install CoffeeScript globally by entering the following command in it.

```
c:\> npm install -g coffeescript
```

On executing the above command, CoffeeScript will be installed in your system by producing the following output
Verification
You can verify the installation of the CoffeeScript by typing the following command.

```
c:> coffee -v
```

On successful installation, this command gives you the version of CoffeeScript as shown below.
On installing CoffeeScript on Node.js, we can access the coffee-command line utility. In here, the coffee command is the key command. Using various options of this command, we can compile and execute the CoffeeScript files.

You can see the list of options of the coffee command using its -h or --help option. Open the Node.js command prompt and execute the following command in it.

```
c:\>coffee -help
```

This command gives you the list of various options of the coffee, along with the description of the operation performed by each of them as shown below.
Compiling the CoffeeScript Code

The CoffeeScript files are saved with the extension `.coffee`. You can compile these files using the `-c` or `--compile` option of the coffee command as shown below.

```
c:\>coffee -c filename.coffee
```

Example

Suppose there is a file in your system with the following CoffeeScript code which prints a message on the console.

```
name = "Raju"
console.log "Hello"+name+" Welcome to Tutorialspoint"
```

**Note** – The `console.log()` function prints the given string on the console.

To compile the above code, save it in a file with the name `sample.coffee`. Open the Node.js command prompt. Browse through the path where you have saved the file and compile it using the `-c` option of the coffee command of the `coffee command-line utility` as shown below.

```
c:\> coffee -c sample.coffee
```

On executing the above command, the CoffeeScript compiler compiles the given file (`sample.coffee`) and saves it in the current location with a name `sample.js` as shown below.

If you open the `sample.js` file, you can observe the generated JavaScript as shown below.

```
// Generated by CoffeeScript 1.10.0
```
CoffeeScript

(function() {
    var name;
    name = "Raju";
    console.log("Hello " + name + " Welcome to Tutorialspoint");
}).call(this);

**Executing the CoffeeScript code**

You can execute a CoffeeScript file by simply passing the file name to the coffee command in the Node.js command prompt as follows.

```
c:\> coffee sample.coffee
```

**Example**

For example, let us execute the sample.coffee file. For this, open the Node.js command prompt. Browse through the path where you have saved the file and execute the file by directly passing its name to the coffee command as shown below.

![Node.js command prompt](image)

**Watch and Compile**

In some scenarios, there is a chance that we do a lot of changes to our scripts. Using the `-w` option of the coffee command, you watch your scripts for changes.

You can watch and compile a file simultaneously using the `-wc` option as shown below. When we use this option, the file will be recompiled each time you make changes in your script.

```
c:\>coffee -wc file_name
```

**Example**

Suppose we have compiled a file named `sample.coffee` using the `-wc` option and we modified the script thrice. Each time we change the script, the `.coffee` file is recompiled leaving the Node.js command prompt as shown below.
Setting the Output Directory

Using the `-o` option, we can set the output directory to place the compiled JavaScript files as shown below.

```
c:\>coffee -o "Required path where we want our .js files" file_name
```

Example

Let us save the JavaScript code of the sample.coffee file in a folder named `data` in the E drive using the `-o` option by executing the following command in the command prompt.

```
c:\>coffee -o E:\data sample.coffee
```

Following is the snapshot of the given folder after executing the above command. Here you can observe the JavaScript file of the sample.coffee
Print the Compiled JavaScript

If we want to print the compiled javascript on the console itself, we have to use the `-p` option of the coffee command as shown below.

```bash
c:\>coffee -p file_name
```

Example

For example, you can print the compiled JavaScript code of the `sample.coffee` file on the console using the `-p` option as shown below.

![Sample CoffeeScript Code](image)

The REPL (Read Evaluate Print Loop)

CoffeeScript provides you an REPL-interactive shell. This shell is used to evaluate the CoffeeScript expressions. You can type any CoffeeScript code in this shell and get the result immediately. You can open REPL by executing the `coffee` command without any options as shown below.

![CoffeeScript REPL](image)

Using this shell, we can assign values to variables, create functions, and evaluate results. As shown in the following screenshot, if we call functions in REPL, it prints the value of the
function. If we give an expression to it, it evaluates and prints the result of the expression. And if we simply type the statements in it, it prints the value of the last statement.

In RPEL, you can access multiple line mode by pressing `ctrl+v` where you can evaluate the code with multiple lines (like functions) and you can get back to REPL mode from it by pressing `ctrl+v` again. Here is an example usage of the multi line mode.

**Running CoffeeScript through Browser**

We can run CoffeeScript using the `<script>` tag of the HTML just like JavaScript as shown below.

```html
<script src="http://jashkenas.github.com/coffee-script/extras/coffee-script.js" type="text/javascript" charset="utf-8"></script>

<script type="text/coffeescript">
  # Some CoffeeScript
</script>
```
But for this, we have to import the library in each application and the CoffeeScript code will be interpreted line by line before the output is shown. This will slow down your applications, therefore this approach is not recommended.

Therefore, to use CoffeeScript in your applications, you need to pre-compile them using the Coffee command-line utility and then you can use the generated JavaScript in your applications.
End of ebook preview
If you liked what you saw...
Buy it from our store @ https://store.tutorialspoint.com