

# **The Ruby Style Guide**

Tsoftime Inc.

<b>Source Code Layout</b>	<b>3</b>
<b>Syntax</b>	<b>3</b>
<b>Naming</b>	<b>4</b>
<b>Annotations</b>	<b>5</b>
<b>Classes</b>	<b>5</b>
<b>Exceptions</b>	<b>6</b>
<b>Collections</b>	<b>6</b>
<b>Strings</b>	<b>6</b>
<b>Percent Literals</b>	<b>6</b>
<b>Misc</b>	<b>7</b>

## Source Code Layout

- Use UTF-8 as the source file encoding.
- Use two spaces per indentation level. 2格缩进。
- Use Unix-style line endings.
- Use spaces around operators, after commas, colons and semicolons, around { and before }.

The only exception is when using the exponent operator:

# bad

```
e = M * c ** 2
```

# good

```
e = M * c**2
```

- Indent when as deep as case. when和case缩进相同。
- Use empty lines between defs and to break up a method into logical paragraphs. 函数之间用空行分割。
- Use RDoc and its conventions for API documentation. Don't put an empty line between the comment block and the def. 使用标准的RDoc格式注释。
- Keep lines fewer than 120 characters. 每行不多余120个字符。
- Avoid trailing whitespace. 行尾不要有空格。

## Syntax

- Use def with parentheses when there are arguments. Omit the parentheses when the method doesn't accept any arguments. 使用def声明函数时，如果有参数，则加上括号（），如果没有参数，则省略括号。
- 使用迭代器代替for语句。
- Use &&/|| for boolean expressions, and/or for control flow. (Rule of thumb: If you have to use outer parentheses, you are using the wrong operators.)
- Avoid multi-line ?: (the ternary operator), use if/unless instead. 避免三元运算符（? : ）被分成多行。

- Favor modifier if/unless usage when you have a single-line body. Another good alternative is the usage of control flow `and/or`. 当if/unless只有一行的时候，优先使用修饰符。
- Favor unless over if for negative conditions (or control flow `or`). unless优先于if not.
- Never use unless with else. Rewrite these with the positive case first. unless不要和else一起使用。
- Omit parentheses around parameters for methods that are part of an internal DSL (e.g. Rake, Rails, RSpec), methods that are with "keyword" status in Ruby (e.g. `attr_reader`, `puts`) and attribute access methods. Use parentheses around the arguments of all other method invocations. 函数调用要加括号，除了前面的几种情况。
- Prefer `{...}` over `do...end` for single-line blocks. Avoid using `{...}` for multi-line blocks (multiline chaining is always ugly). Always use `do...end` for "control flow" and "method definitions" (e.g. in Rakefiles and certain DSLs). Avoid `do...end` when chaining. 单行block用`{}`，多行用`do...end`。
- Avoid return where not required. 避免不必要的return。
- Use `||=` freely to initialize variables. 使用`||=`运算来初始化变量。
- Don't use `||=` to initialize boolean variables. 不要用`||=`来初始化布尔变量。
- Never put a space between a method name and the opening parenthesis. 函数名和后面的括号之间严禁添加空格！
- Always run the Ruby interpreter with the `-w` option so it will warn you if you forget either of the rules above!

## Naming

- Use snake\_case for methods and variables. 变量名和函数名用小写加下划线。
- Use CamelCase for classes and modules. 类名和模块名用驼峰格式。
- Use SCREAMING\_SNAKE\_CASE for other constants. 常量用全大写加下划线。

- The names of predicate methods (methods that return a boolean value) should end in a question mark. (i.e. `Array#empty?`).
- The names of potentially "dangerous" methods (i.e. methods that modify self or the arguments, `exit!`, etc.) should end with an exclamation mark.
- When using `reduce` with short blocks, name the arguments `la`, `el` (accumulator, element).
- Prefer `map` over `collect`, `find` over `detect`, `select` over `find_all`, `reduce` over `inject` and `size` over `length`.

## Annotations

- Annotations should usually be written on the line immediately above the relevant code.  
注解在所注解的代码上面
- The annotation keyword is followed by a colon and a space, then a note describing the problem. 注解关键字后面跟随一个冒号和一个空格。
- If multiple lines are required to describe the problem, subsequent lines should be indented two spaces after the `#`.
- Use `TODO` to note missing features or functionality that should be added at a later date.
- Use `FIXME` to note broken code that needs to be fixed.
- Use `OPTIMIZE` to note slow or inefficient code that may cause performance problems.
- Use `HACK` to note code smells where questionable coding practices were used and should be refactored away.
- Use `REVIEW` to note anything that should be looked at to confirm it is working as intended. For example:`REVIEW: Are we sure this is how the client does X currently?`

## Classes

- Always supply a proper `to_s` method for classes that represent domain objects.
- Use the `attr` family of functions to define trivial accessors or mutators. 使用`attr`系列函数来定义成员变量的访问属性。
- Prefer [duck-typing](#) over inheritance. 使用`duck-typing`代替继承。

- Avoid the usage of class (`@@`) variables due to their "nasty" behavior in inheritance. 避免使用类变量。
- Use `def self.method` to define singleton methods. This makes the methods more resistant to refactoring changes.

## Exceptions

- Don't suppress exceptions.
- Don't use exceptions for flow of control. 不要使用异常来控制流程。
- Avoid rescuing the Exception class. 不要捕获Exception。

## Collections

- Prefer `%w` to the literal array syntax when you need an array of strings.
- Use symbols instead of strings as hash keys. 使用symbols作为hash的key。
- Avoid the use of mutable object as hash keys. 避免使用可变的实例作为hash的key。

## Strings

- Prefer string interpolation instead of string concatenation。使用字符串 interpolation来代替字符串连接。
- Prefer single-quoted strings when you don't need string interpolation or special symbols such as `\t`, `\n`, `'`, etc.
- Avoid using `String#+` when you need to construct large data chunks. Instead, use `String#<<`. Concatenation mutates the string instance in-place and is always faster than `String#+`, which creates a bunch of new string objects. 在构造大数据块的时候，使用 `String#<<`代替 `String#+`。

## Percent Literals

- Use `%w` freely.  
`STATES=%w(draft open closed)`

- Use %r only for regular expressions matching more than one '/' character.
- Prefer () as delimiters for all % literals.

## Misc

- Avoid methods longer than 10 LOC (lines of code). Ideally, most methods will be shorter than 5 LOC. Empty lines do not contribute to the relevant LOC.
- Avoid parameter lists longer than three or four parameters. 避免参数超过4个。
- Avoid alias when alias\_method will do.
- Avoid more than three levels of block nesting. 块嵌套不要超过三层。