# *Coding convention in Ruby:*

1. *Quy tắc đặt tên:*

* Sử dụng cú pháp snake\_cake:

tất cả các chữ cái đều viết thường, và các từ cách nhau bởi dấu gạch dưới.  
Ví dụ: user\_name, this\_is\_the\_name\_that\_follow\_the\_snake\_case

* Tên lớp thì đặt theo PascalCase:

Cú pháp Pascal viết hoa chữ cái đầu tiên của mỗi từ.  
Ví dụ: ProductName, PascalCasing, ThisIsTheNameThatFollowThePascalCase

* Hằng số đặt theo UPPER\_CASE, vd: CLICK\_COUNTER
* Tên biến, tên lớp thường là danh từ, cụm danh từ hoặc tính từ: UserModel, user\_name, download\_counter, downloaded?
* Tên hàm thường bắt đầu bằng động từ: pushToUser
* Tên thì phải có nghĩa, KHÔNG ĐƯỢC đặt tên kiểu viết tắt (dlCounter, uName, idl, a, a1, doFA)
* Tránh đặt những tên quá chung chung, tối nghĩa: top, best, doIncrease, getAll

1. *Quy tắc về số lượng:* **“if small is good, then smaller must be better”**

* Số lượng dòng code trong hàm/lớp, số lượng hàm trong lớp, số lượng lớp trong gói phải giữ ở một giới hạn nhất định nào đó, và nên giữ càng ít càng tốt, ví dụ:
* Hàm không nên quá **30** dòng ([Martin Lippert-rule 30](http://www.amazon.com/Refactoring-Large-Software-Projects-Restructurings/dp/0470858923))
* Lớp không nên vượt quá **500** dòng (Clean Code)
* Một hàm không được vượt quá **5** tham số, (nên giữ <=3)(Clean Code)
* Một hàm chỉ làm duy nhất **1** việc, trong trường hợp chính đáng, làm 2 việc cũng được phép, tuy nhiên tên hàm phải nói rõ điều này, vd: increaseDownloadCounterAndUpdateToDatabase
* Khi khai báo biến, một dòng chỉ chứa một biến
* Một dòng không nên dài quá **80** hoặc **120** (prefered) ký tự ([Oracle](http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-136091.html#313))
* Các câu lệnh lồng nhau tối đa **4** cấp

1. *Quy tắc xuống hàng:*

* Nếu có dấu phẩy thì xuống hàng sau dấu phẩy ,
* Xuống hàng trước toán tử + -…
* Nếu có nhiều cấp lồng nhau, thì xuống hàng theo từng cấp
* Dòng xuống hàng mới thì nên bắt đầu ở cùng cột với đoạn lệnh cùng cấp ở trên.

1. *Comment:*

* Tập code giả

Vd:

#bước 1: get parameter

Coding…

#bước 2: xử lí

Coding…

…

* Dùng comment trong trường hợp viết documentation cho thư viện, thông tin đính kèm cho class…\
* Plugin Ruby Linter hỗ trợ convention giúp mình kiểm tra và sửa lỗi. Đọc thêm tại: http://blog.appconus.com/2016/05/27/ruby-linter-va-cac-linter-plugin-cho-atom-va-sublimetext/