

10 Commandments

*for creating good **Ruby** code*

Marty Haught - **Boulder Ruby**

Source

- originally posted on June 4th by Alberto Gutierrez
- www.makinggoodsoftware.com
- made slight modifications
- not intended to be definitive

I. DRY - Don't Repeat Yourself

- similar/identical snippets of code
- constants/enums
- configuration
- list of objects

2. Write Short Methods

- easier to read
- easier to reuse
- easier to test

3. Use descriptive naming

- names of variable, methods and classes should be self-explanatory and accurate.
- usually the hardest thing to get right quickly

4. Assign the right responsibility

- one class, one responsibility
- there should never be more than one reason for a class to change (via SOLID)

5. Keep your code organized

- physical organization - your classes/modules should be put in a place that is easy and intuitive to find. (Rails/Gems do this automatically)
- logical organization - group related classes/modules together. (thinks models, controllers, helpers)

6. Create effective tests

- originally was 'Create lots of unit tests'.
Why not?
- the more tests you have, the better!!
- automated + effective FTW!

7. Code fast, refactor often

- originally was 'Refactor often and sooner'.
- requires tests to know your implementation works
- code fast to get started, avoid mental blocks
- don't worry about ugly/dumb
- refactor to genius

8. Comments are evil

- why are there comments?
- great code should be obvious in what it is doing
- capture scenario in tests/specs to gain understanding
- explaining why is sometimes not possible without comments

9. Code to a contract

- code to an interface, not the implementation
- frees the internal implementation of a class to change

10. Have code reviews

- getting feedback from fellow developers is invaluable for everyone
- pair programming is the best, if possible

Missing: Simplicity

- start with the simplest thing that could possibly work
- yagni - you'll be surprised how often you won't need it!

Missing: Premature Optimization

- “root of all evil” - Donald Knuth
- can affect good design for little or no gain
- wait until you really need it
- you may be surprised where the optimization efforts are needed

Missing: ?

- Anything else?