KISS

Keep **I**t **S**imple, **S**tupid or **K**eep **I**t **S**imple and **S**tupid.

- Do the simplest thing that could work.

- Rational
  - Simplicity should be a major goal in software development
  - Unnecessary complexity should be avoided
  - The software has already an implicit complexity
  - Add only the amount of complexity that makes things easier (pattern)

YAGNI

You **A**rent't **G**onna **N**eed **I**t or **Y**ou **A**in't **G**onna **N**eed **I**t

- Rational
    - Strongly related to KISS
    - Against speculative generalization and over-engineering
    - Don't produce something for possible future use-cases
    - The unnecessary code wastes time, complicates the code and introduces bugs
    - Refactor your code if necessary

DRY

Don't **R**epeat **Y**ourself or Once And Only Once (OOAO)

- Rational
  - One of the most often (unintentionally or intentionally) violated rule
  - When one piece of code is changed, maybe not the other one
  - "every piece of knowledge must have a single, unambiguous, authoritative representation within the system" (*The Pragmatic Programmer*)

PLA

Principle of **L**east **A**stonishment or **P**rinciple of **L**east **S**urprise

- Rational
  - The user should not be surprised by unexpected behavior or mysterious side effects of the user interface
  - A function should do exactly what its function name implies
  - "Make interfaces easy to use correctly and hard to use incorrectly." (Scott Meyers)

Boy Scout Rule

Always leave the campground cleaner than you found it.

- Rational
  - Whenever you find a piece of code that needs to be improved, you should fix it.
  - The code would not rot
  - Small steps improve the code
    - Renaming bad names
    - Decomposing large functions
    - Deleting the comment by making it self-explanatory
    - Cleaning up complex code
    - Remove duplicates