Best Practices - Coding

## Good Code

1. Use version control - git and github
2. Problem driven development - solve a problem in each commit, do not design features, since each feature should be solving a problem anyway
3. Make it open source - makes your code quality better and gets community help
4. Always make APIs - Everything should be the body of an API
5. Don't document code - Document APIs only, comments in code do not help and get out of date quickly
6. Readability - Style should be for readability only, since it is imporant that you or someone else can go back and read and understand the code quickly
7. Do not prematurely optimise code, since it is very likely you either wont need to or you will optimise the wrong part
8. Use your own work, since you'll understand what your doing and probably write better code
9. Make portable code

## API Design

1. Make only what you need today
2. Make the API modular
3. Structured syntax - thing.action, thing.propety, instead or do\_action\_with\_thing
4. API should be self-consistent, with every class having the same style and conventions
5. API should be extensible
6. API should be fully testable - do not worry code coverage as long as external contracts are tested
7. Keep it simple to use - have default options, make everything as simple as possible
8. Keep it portable - system concepts should be abtracted

## System Design

1. Start IDs at a large number
2. Only store valid UTF-8
3. Fail loudly on SQL syntax errors