



YAGNI
DRY
KISS

Background

- Dražen Pavetić
- ~ 10 year Python exp.
- Web, scripting, gui*
- Software engineer at VuMedi

Introduction

- Lots of programming principles
- Can be rules (or formal agreements)
- Not to follow blindly*
- Mixing is good
- Project requirements and sane team judgment
- Wikipedia:
https://en.wikipedia.org/wiki/Category:Programming_principles

YAGNI

- You Ain't Gonna Need It
- Prototyping (first exception :)
- Reading and good preparation
- Blueprint(s)
- Live project
- Refactoring

DRY

- Don't Repeat Yourself
- Single real world representation (module, class, function)
- Leads to:
 - Faster code understanding and review
 - Reusable entities and components
 - Lessen fear of change

KISS

- Keep It Short and Simple
 - Variations: | small simple | stupid* simple | ...
- Simple as possible: Basic simplicity
- Not easy at all
- Leads to elegance
- Easy to understand and adjust (with KISS in mind)

KISS

- Code should be in good shape
 - E. g. able to run 400m without panting
- Comments
- Code style
- Use code linting and checking

Real world

- Stuck in implementation
 - Do your best to make it work. Make notes, talk about, search, read and continue to seek better path
- Site is down, database crashed, ...
 - Do not panic :) Pull up sleeves and fix it with any method (copy paste is good). Normalize code after fix.

Zen of Python

- import this

- Thank you for your time