SOURCE CODE MANAGEMENT

WHAT IS A VERSION CONTROL SYSTEM?

- Collaboration with the team.
- History track.
- Easy rollbacks.
- Non-Distributed.
  - Subversion, CVS.
- Distributed.
  - Git, Mercurial.

NON-DISTRIBUTED VS DISTRIBUTED

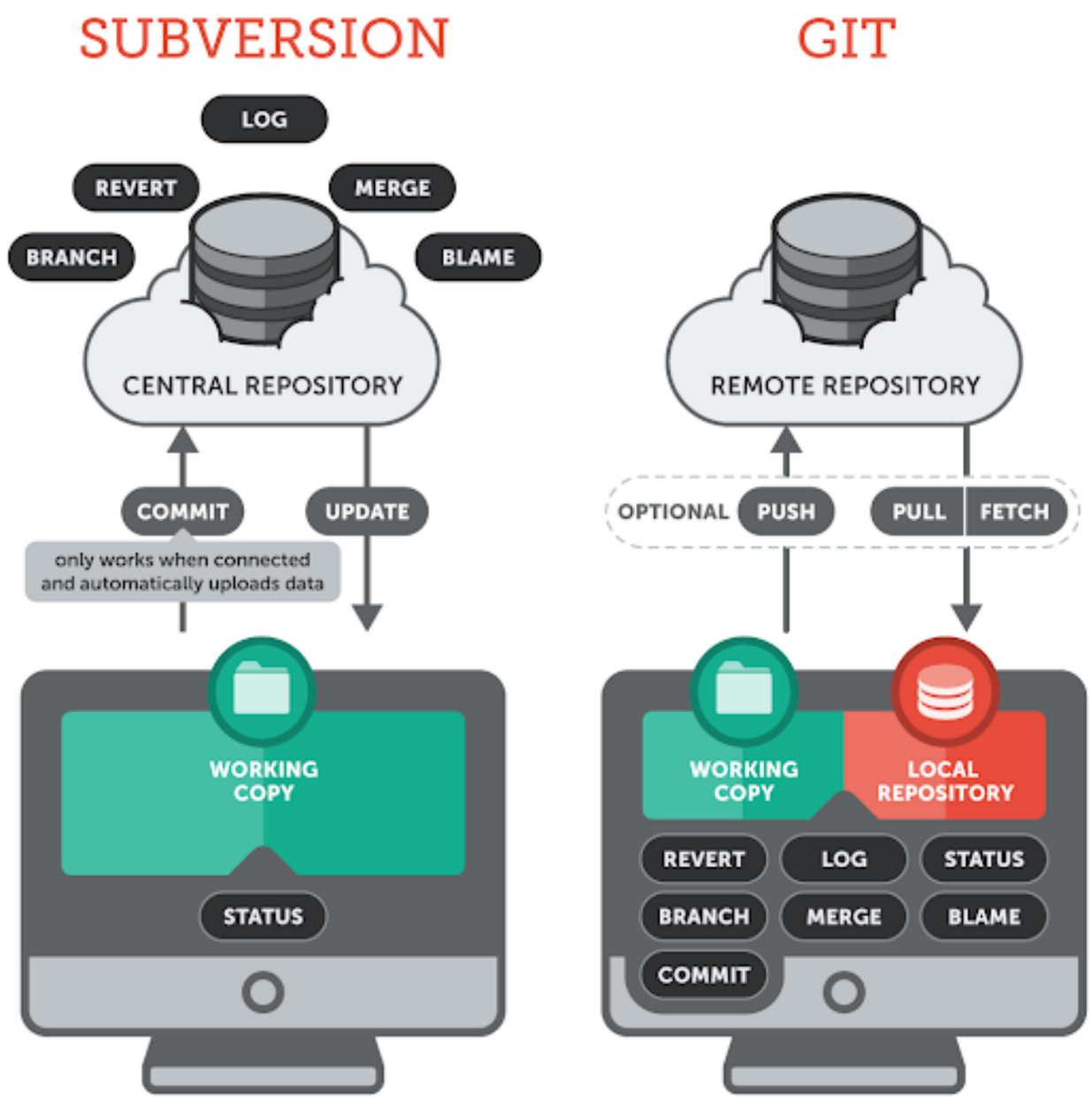


Photo: https://scriptcrunch.com/295/

INDIVIDUALS AND INTERACTIONS

OVER

PROCESSES AND TOOLS

GIT

- Created by Linus Torvalds - 2005.
- Created mainly for programmers but can track any files.
- Free & open source tool.
- Distributed.

Download: ***https://git-scm.com***

Optional: ***https://www.sourcetreeapp.com***

GIT – BASIC COMMANDS

- Git init
- Git clone -b <branch> <remote>
- Git add
- Git commit -m “commit message”
- Git push -u origin master
- Git pull
- Git merge

GIT – BRANCHES

Git branch <branchName>

Git checkout <branchName>

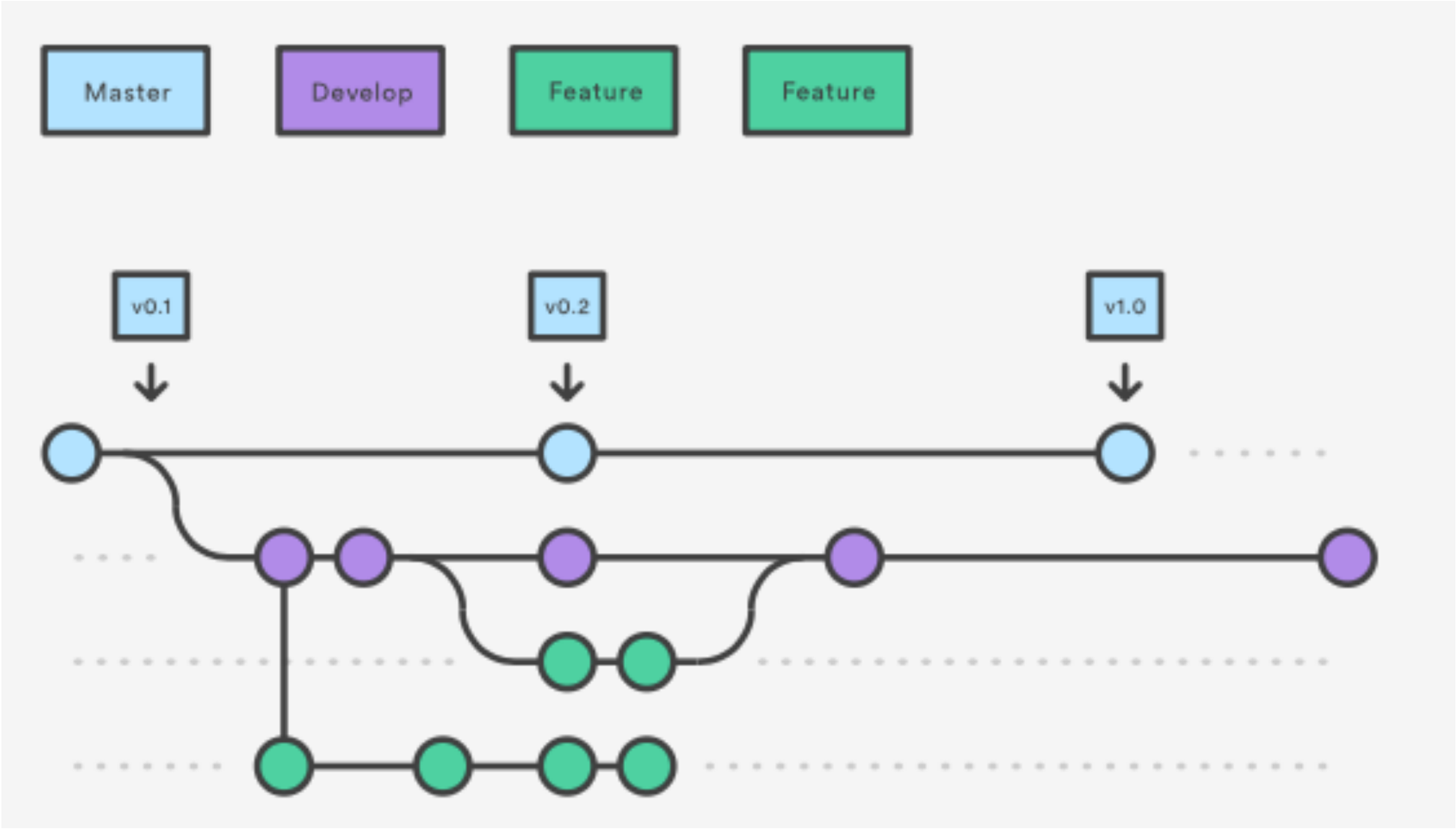
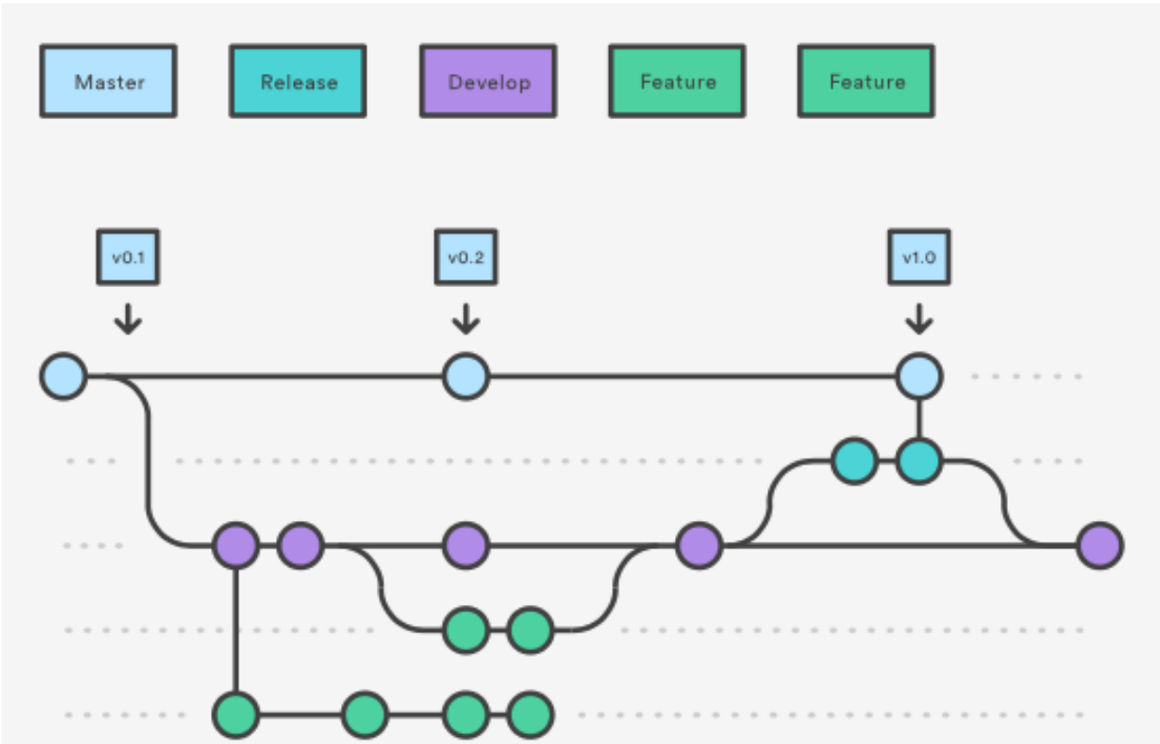


Photo: https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

GIT – BRANCHES: GOOD PRACTICE

- Master - Protected.
  - Production or release branch.
- CI/develop/integration
  - CI branch (used to execute AT and UAT)
- Feature brach
  - Development of each feature or ticket.

GIT – MERGE & MERGE CONFLICTS



index.html styles.css

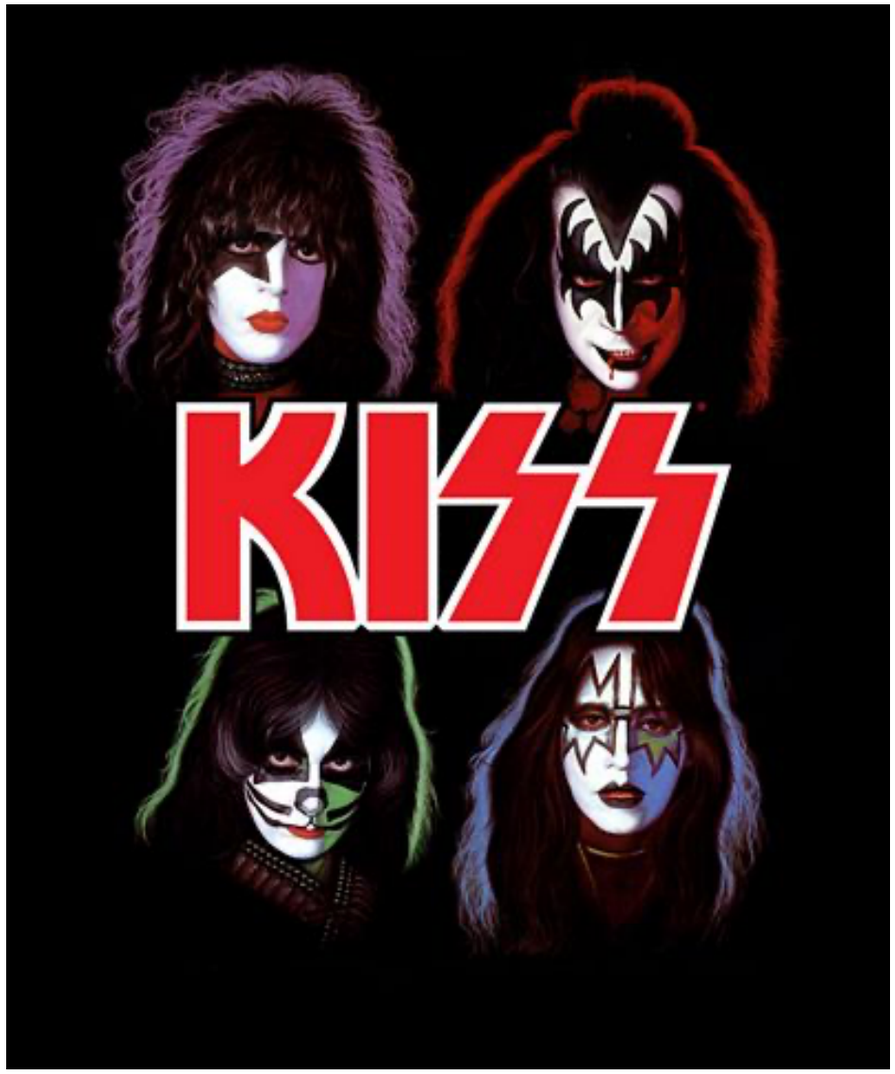
```
<!DOCTYPE html>
<html>
<head>
  <title>GC Merge Demo</title>
  <link href="styles.css" rel="stylesheet" type="text/css" />
</head>
<body>
<<<<<<< HEAD
  <h1>Grand Circus Merge Demo</h1>
=======
  <h1 class="header">Merge Demo</h1>
>>>>>>> 1d46372af5a97f8ef05b9eecb82712382cc5f31c
  <p>
    Demo the merge.
  </p>
  <p class="footer">
    Grand Circus Detroit
  </p>
</body>
</html>
```

EXAMPLE

- Work in couples.
- Create a new project.
- Create different branches.
- Upload some files (in each branch).
- Update same file.
- Merge to develop or integration branch.
- Fix merge conflicts
- Merge to master.

10 Minutes.

SOME BEST PRACTICES IN SOFTWARE DEVELOPMENT



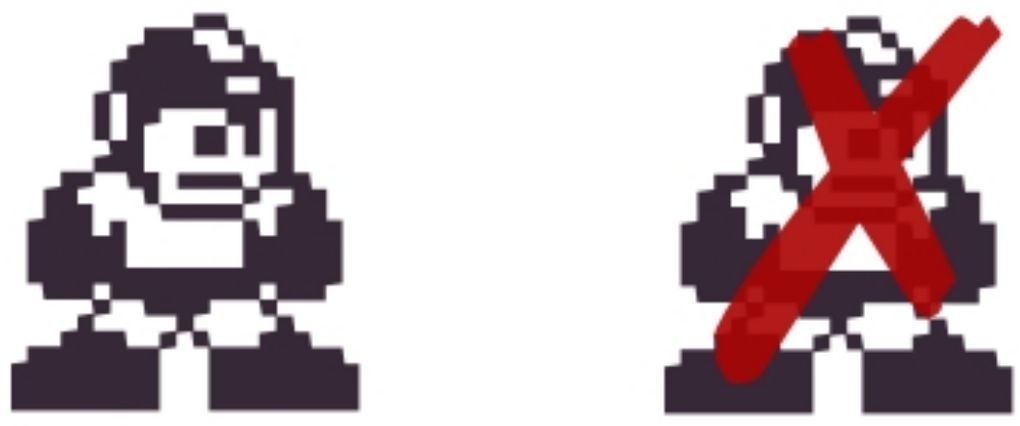
Keep it simple, s....

Y.A.G.N.I

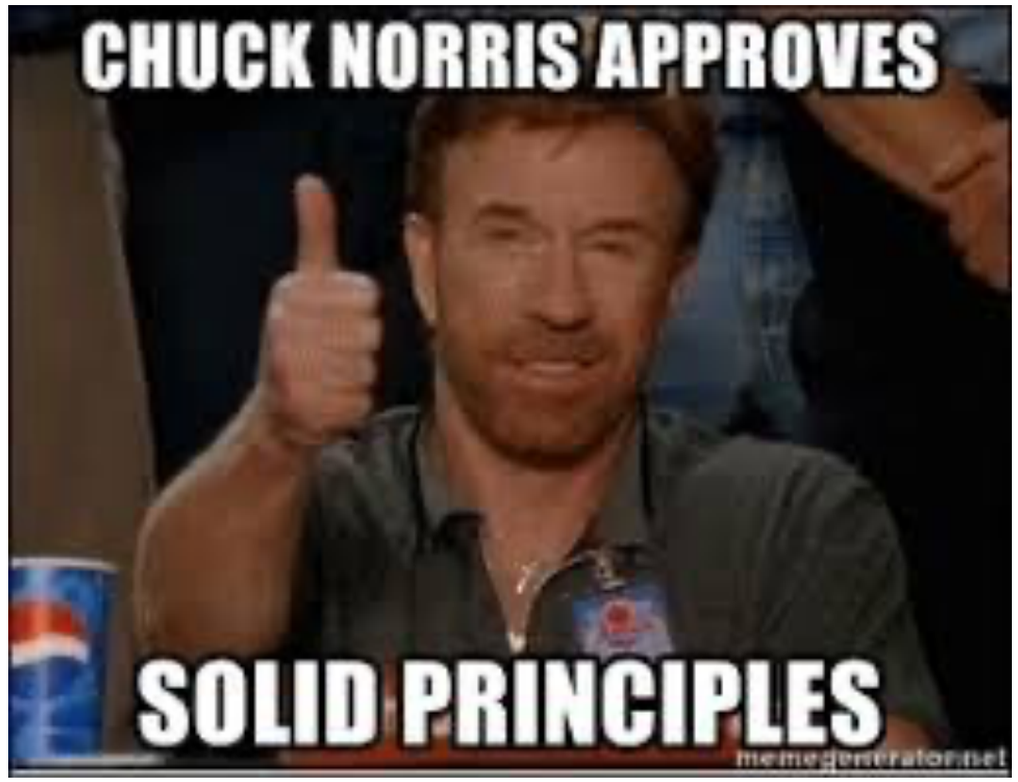
Don't do unnecessary code!

D.R.Y

DRY: Don`t Repeat Yourself

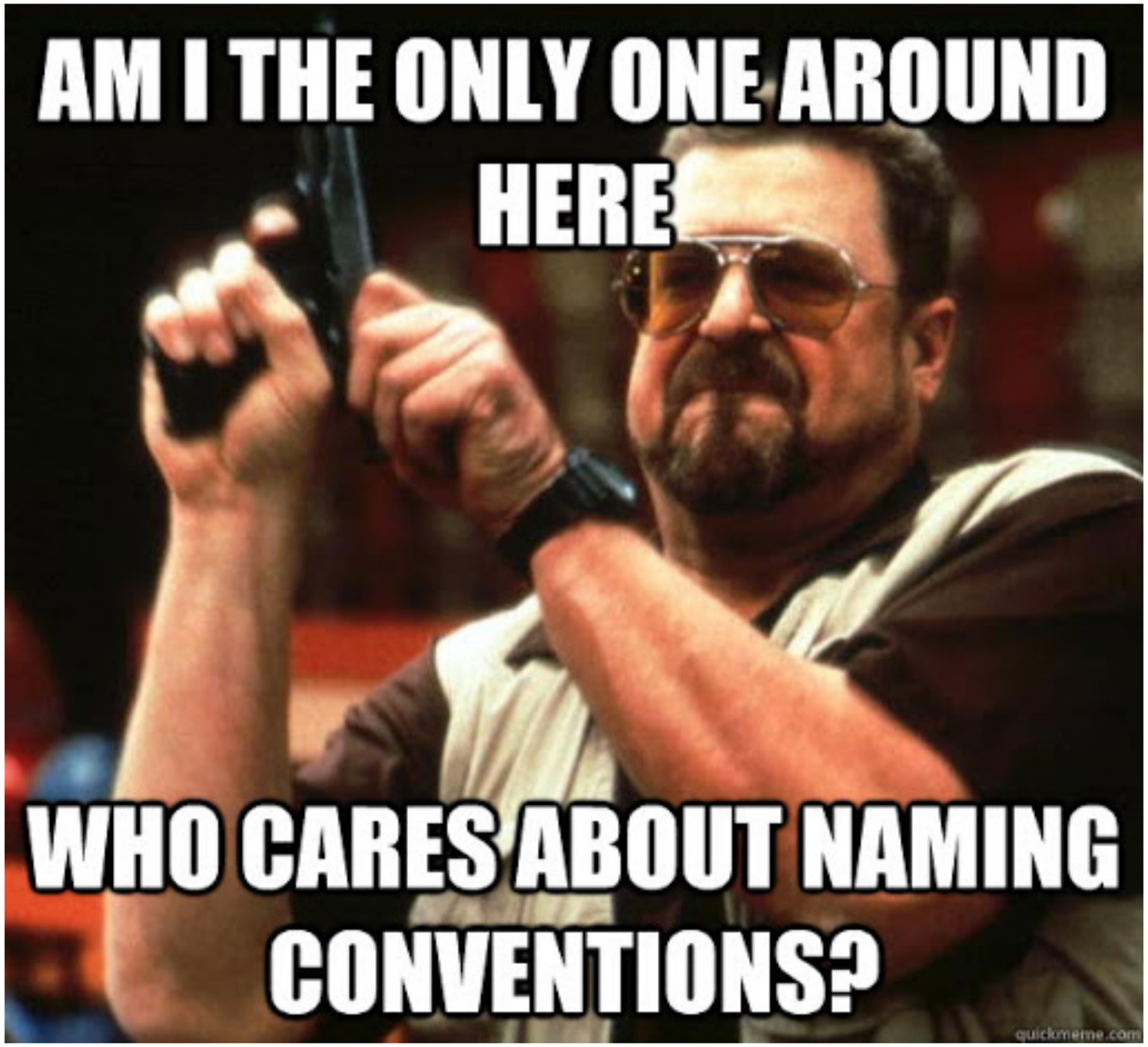


Avoid Duplication!



Use Solid principles.

Meaningful name in variable, methods, classes, etc



Follow a naming convention.

Popular naming conventions

- **Camel Case:**
  - **Eg:** camelCaseName
- **Snake Case:**
  - **Eg**: snake_case_name
- **Kebab Case:**
  - **Eg:** kebab-case-name

LONGEST PLACE NAME IN THE WORLD
Taumatawhakatangihangakoauauotamateaturipukakapikimaungahoronukupokaiwhenuakitanatahu

Name length

Use verbs in methods: Save, close, get, load, etc

Boolean methods: is, has, can.

Use nouns and nouns combinations for classes.

SOLID PRINCIPLES

WHAT IS SOLID?

- Robert C. Martin (Uncle Bob).
- Acronym for 5 principles using OOP.
- Purpose:
    - Understandable.
    - Easier to maintain
    - Easier to extend.

SOLID ACRONYM

S Ingle Responsibility Principle

O pen/Closed Principle

L iskov Substitution Principle

I interface Segregation Principle

D ependency Inversion Principle

S Ingle Responsibility Principle

- Only one single responsibility.
- “Do one thing and do it well”.
- “Reason to change”- Robert C. Martin.

Example

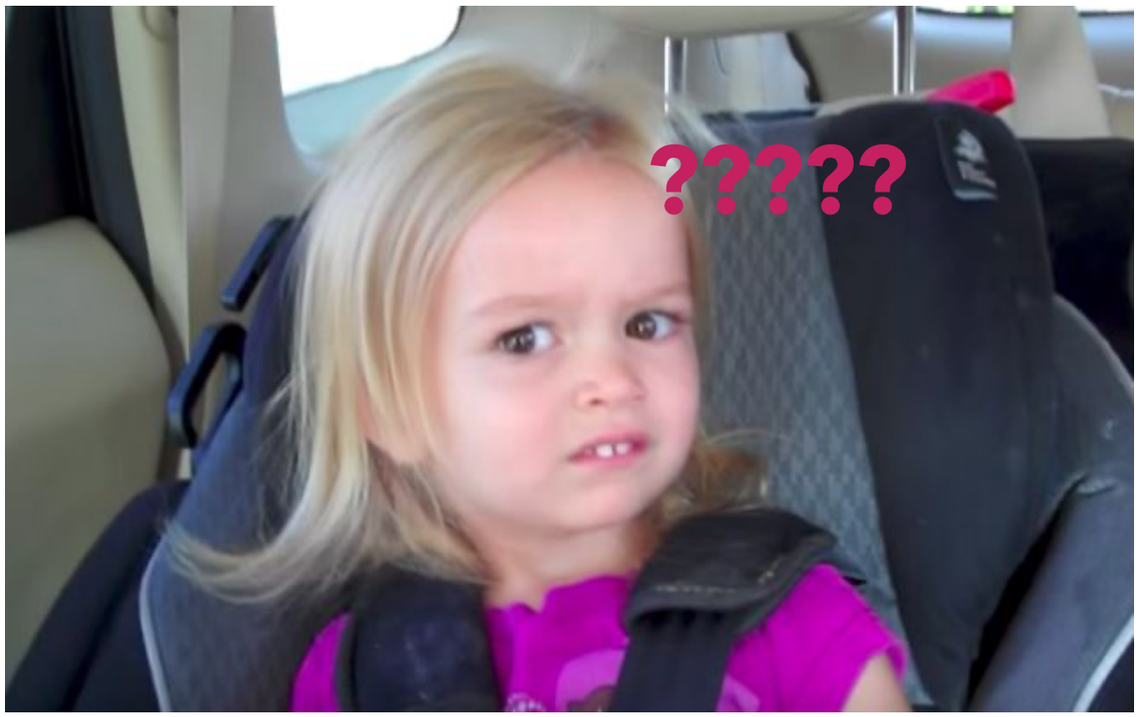
O pen/Closed Principle

- Open for extension BUT close to modification.
- Use inheritance and/or implementing.

Example

L iskov Substitution Principle

- If **"x"** is a subtype of **"y"** that means that the objects of type **"y"** may be replaced with objects of type **"x"**.



L iskov Substitution Principle

"A subclass should behave in such a way that it will not cause problems when used instead of the superclass."

Example

interface Segregation Principle

- Interfaces.
- “No one should be forced to depende on methods it does not use”
- Y.A.G.N.I.
- Create multiple interfaces.

Example

D ependency Inversion Principle

- High level modules should not depend on low-level modules. Both should depend on abstractions.
- Abstractions should not depende on details. Details should depend on abstractions.

Example

SOLID
Software Development is not a Jenga game

THANK YOU.