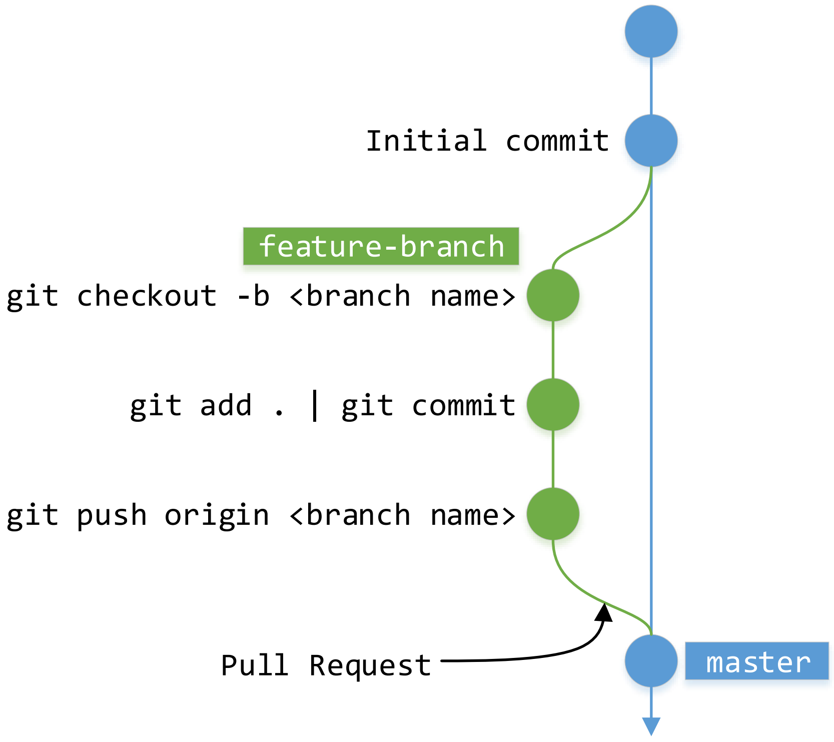
Guidelines for Git

Workflow

1. Create a new branch to work on
    - `git checkout -b <branch name>`
2. Do work and commit
    - `git add .                (gaa)`
    - `git commit               (gc)`
    - In VIM
        1. **i** -> enter input mode
        2. Write message
        3. **Esc** -> Exit input mode
        4. **:wq** -> Write and quit
        5. **:q!** -> Just quit
3. Push commits to remote branch
    - `git push origin <branch name>`
4. Create a pull request to master using website or GUI



Guidelines

- Never work on master branch unless for super small changes
- One branch for each unit of work (feature, document, bug-fix, experiment)
- Don’t push non-functional code to master branch
- Never force a push

Commit Messages

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJ5LKDFJ5DKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Subject

- Limit to 50 characters
  - Easier to view on GitHub, GitLab, Bitbucket, git log etc.
- Capitalize the line
- Use imperative mood
  - Spoken or written as a command or instruction
    - Merge branch ‘thing’
    - Revert ‘commit a’
    - Merge pull request #123
    - Change this
    - Refactor that

Body

- Separate subject line and body with blank line
- Wrap at about 72 characters
- Use the body to explain what was changed and why it changed
  - The how is told by the committed code