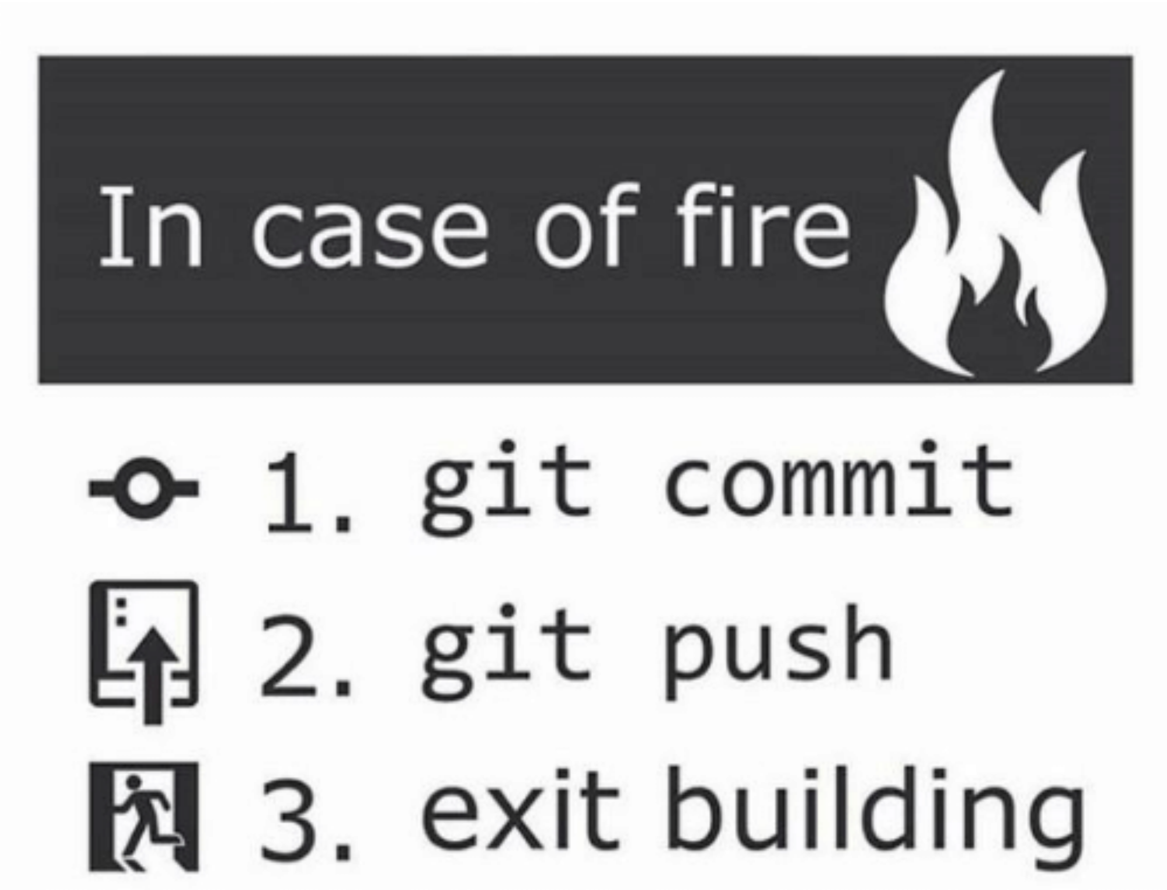
CISC/CMPE 327 Software Quality Assurance

Queen's University, 2020-fall

QA Talk

Git-GitHub: add, commit, repository, working directory, staging, checkout, clone, tag

Fake! Kidding! Not true!



In case of fire

1. git commit
2. git push
3. exit building

Git > Your life

Git v.s. GitHub

- Git
  - File version control system

- Github
  - repository site

Installation

- Just Google It: git download
  - https://git-scm.com/downloads
  - Verify
  - Git –help
  - git config --global user.email "you@example.com"
  - git config --global user.name "Your Name"

Create a local repository

- Clone a repository:
  - git clone

- Alternative (just keep things local first):
- Init a repos:
  - git init

  (so everything will be tracked for any modification)

Add a file

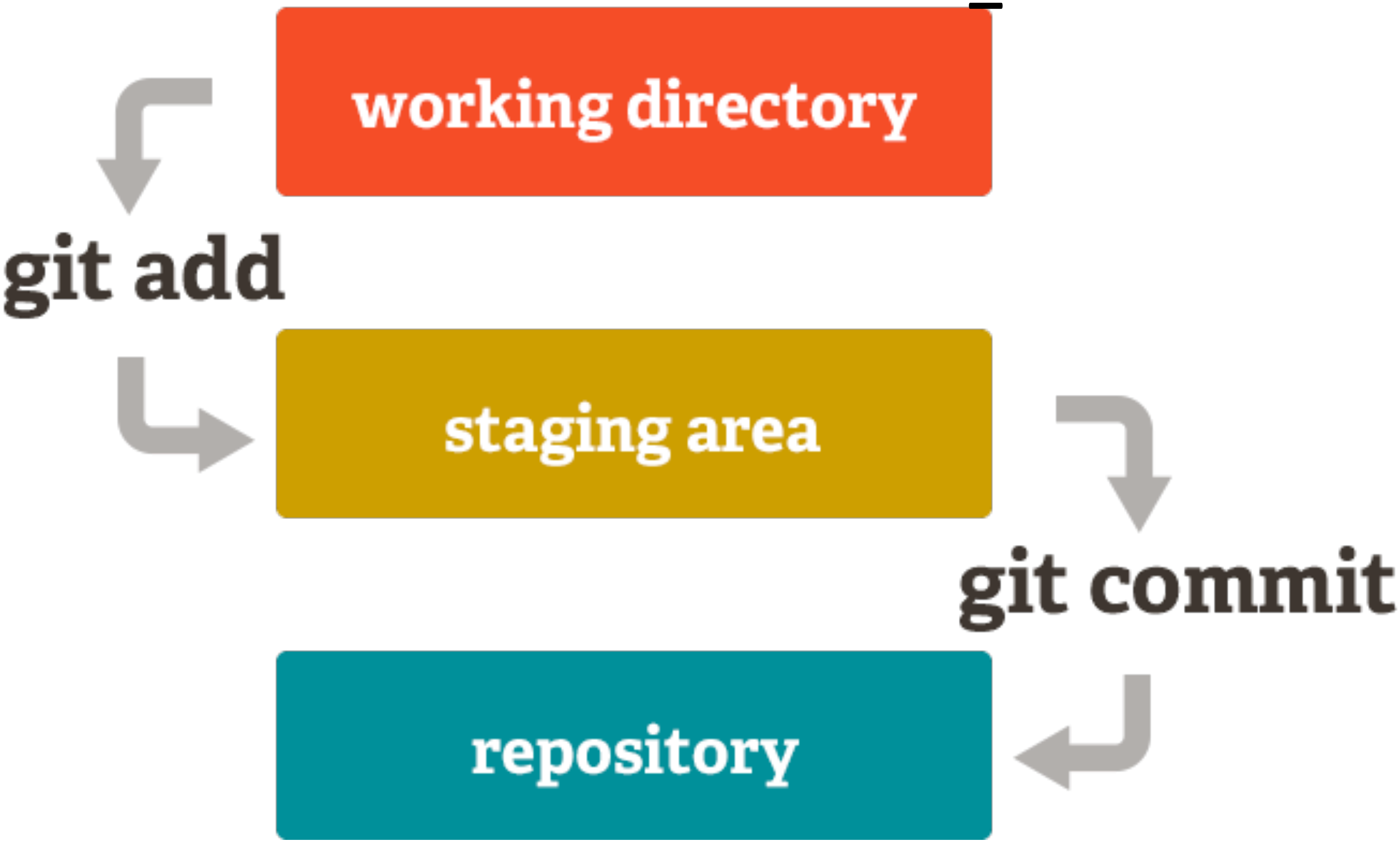
- Create a text file
  - Markdown text file
  - Type 'hello git', saved

- Add all new changes to the repository
  - git add .
  - Or you can specify which file to add git add README.md

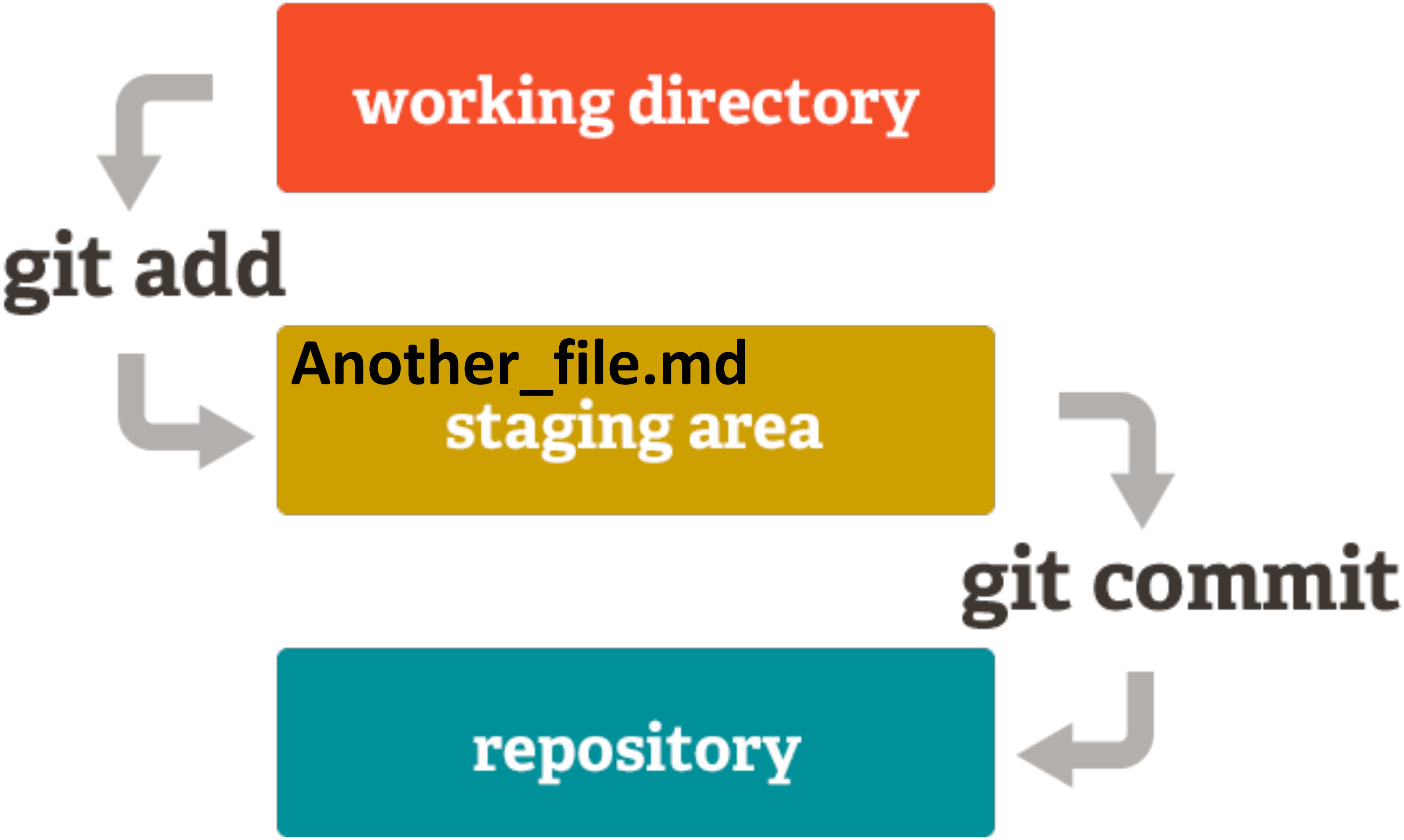
- Create a commit:
  - git commit -m “added a new file”

Create a text file & edit

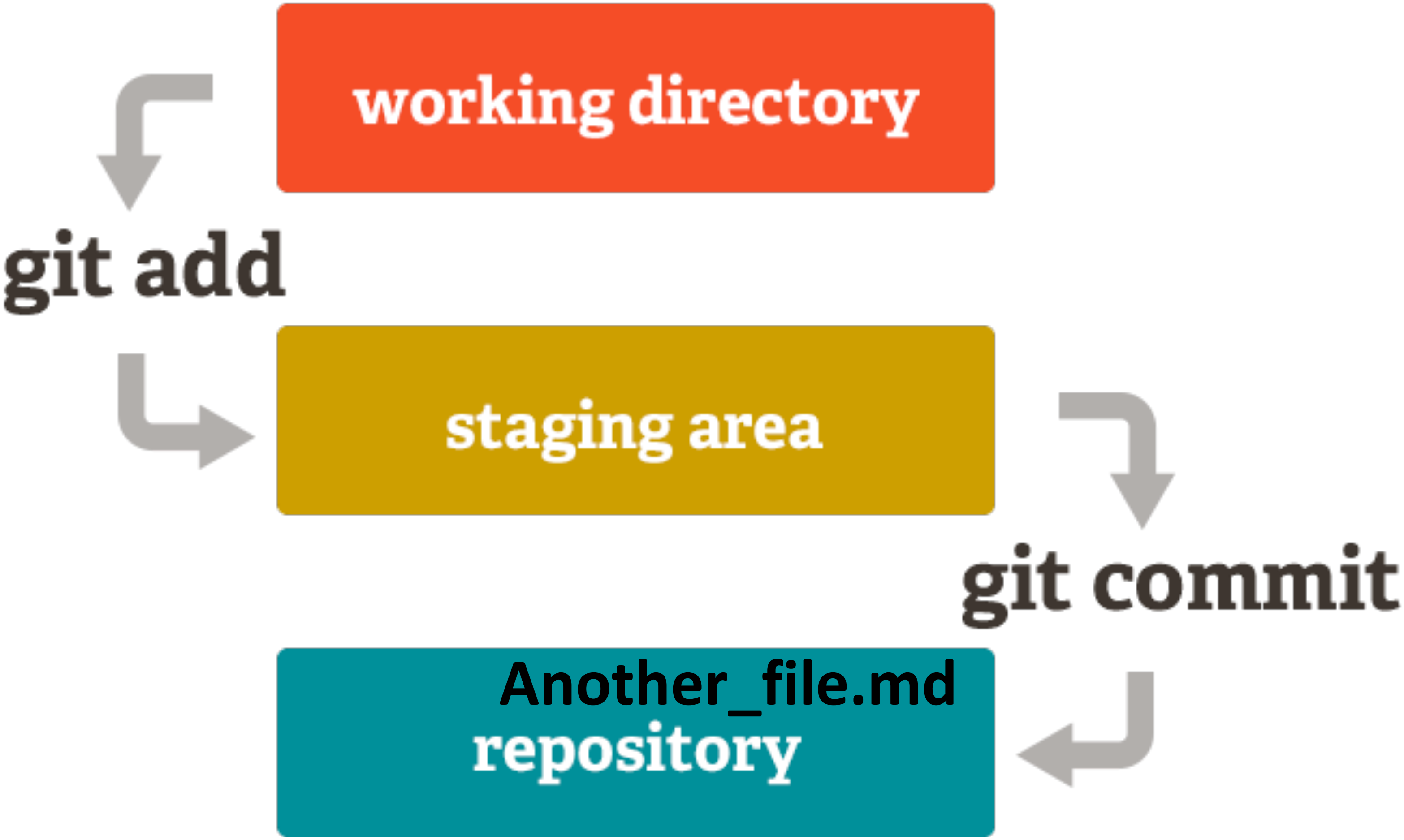
Another_file.md



git add .


working directory
git add
Another_file.md
staging area
git commit
repository

git commit

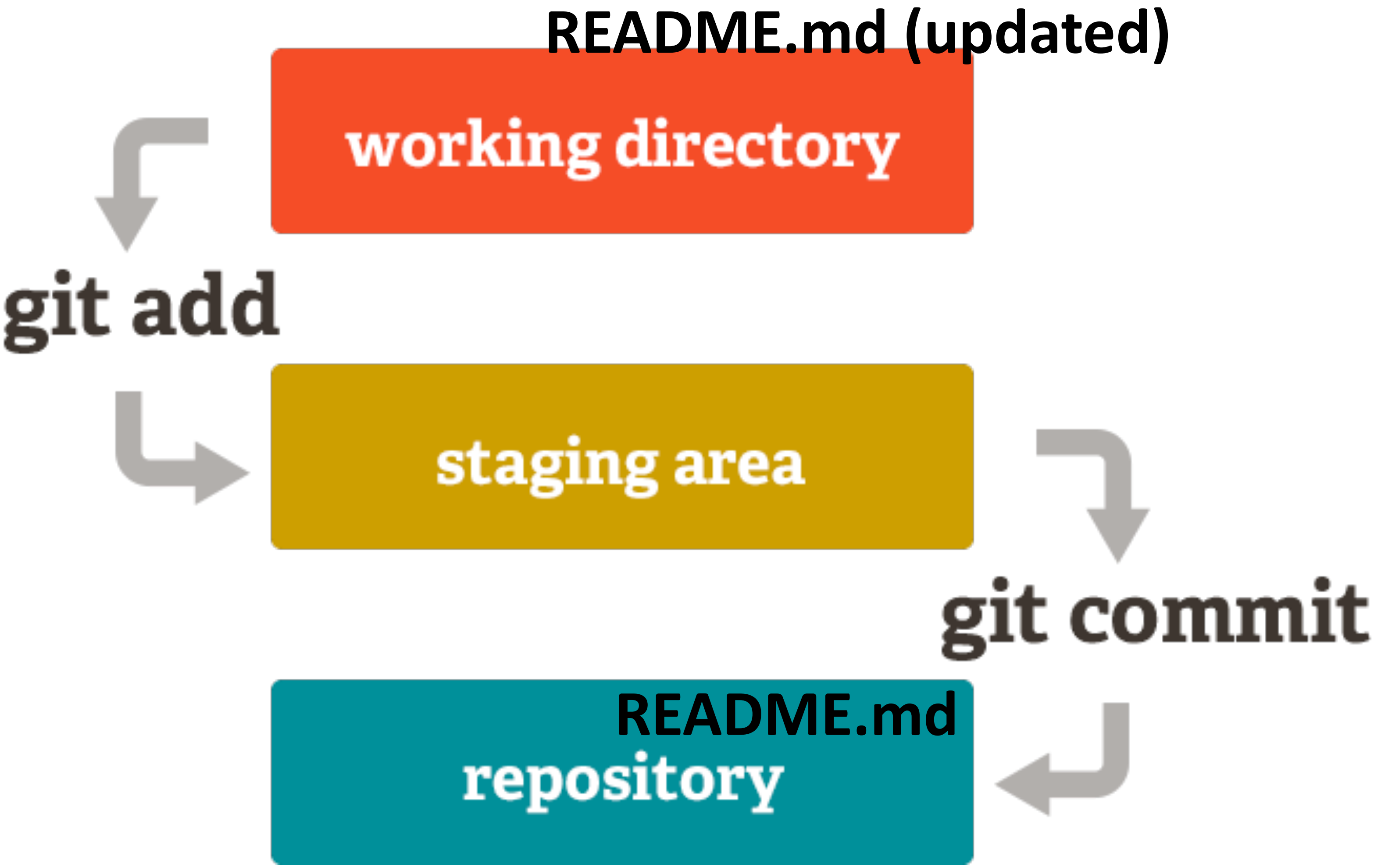

working directory
git add
staging area
git commit
Another_file.md
repository

Check in over time

Commit_0

Anoter_file.md

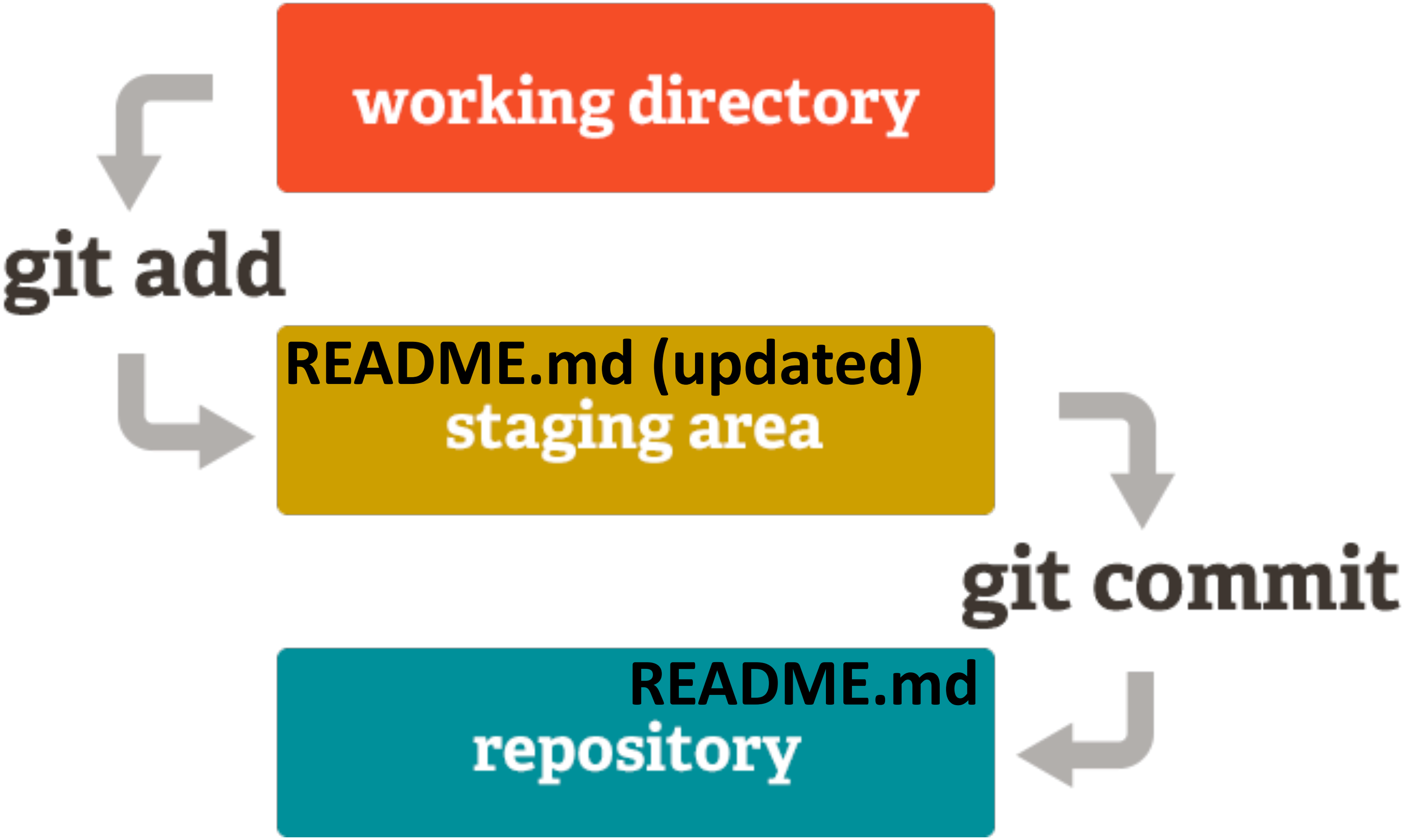
Edit again



README.md (updated)

README.md

git add .



working directory

git add

README.md (updated)

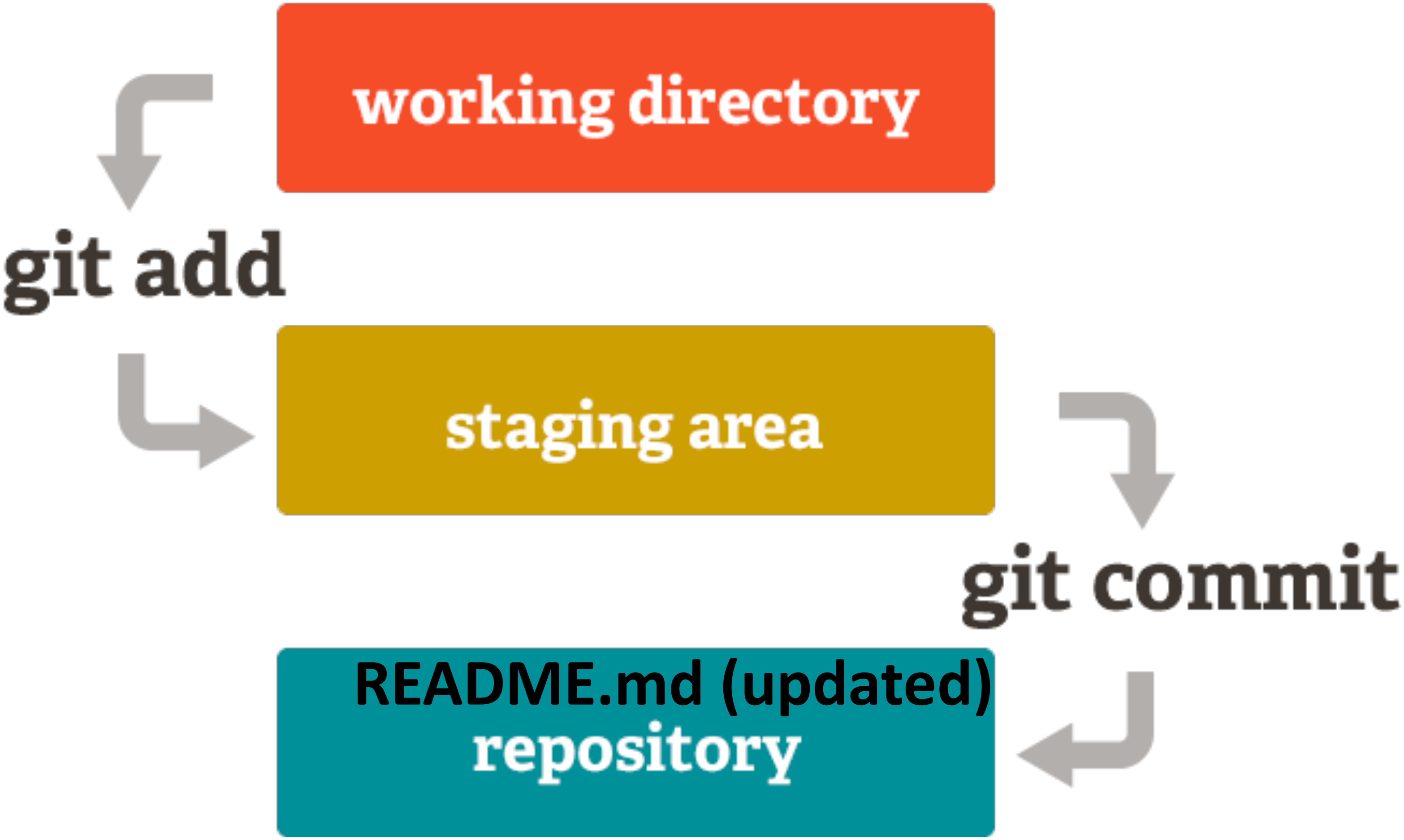
staging area

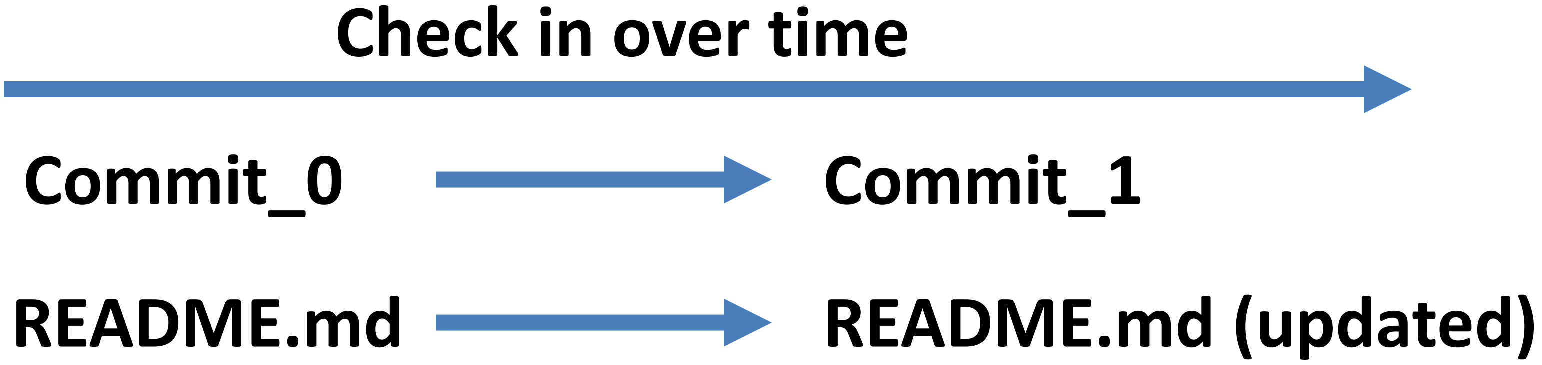
git commit

README.md

repository

git commit

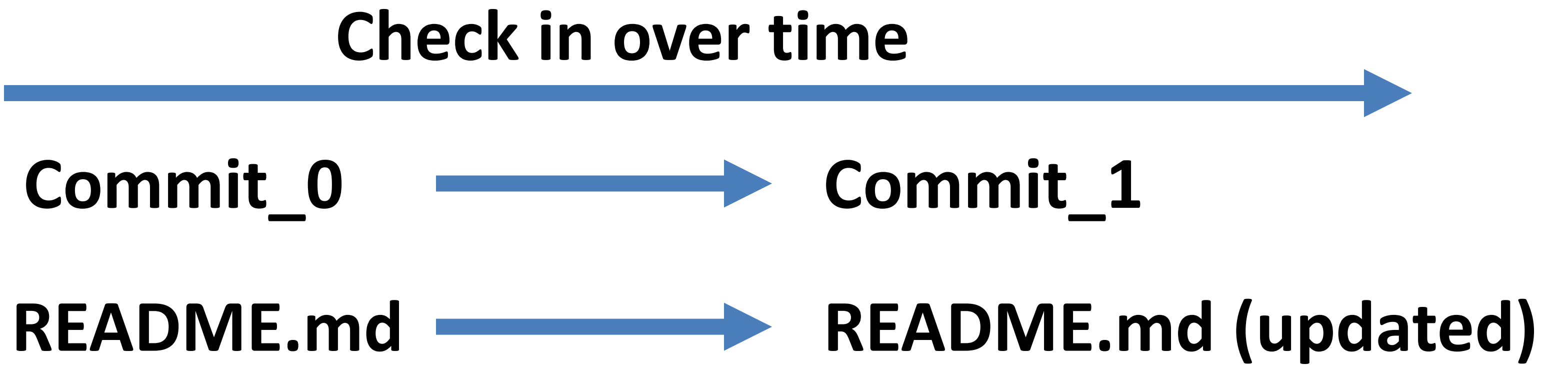

working directory
git add
staging area
git commit
README.md (updated)
repository



Check in over time

Commit_0 → Commit_1

README.md → README.md (updated)



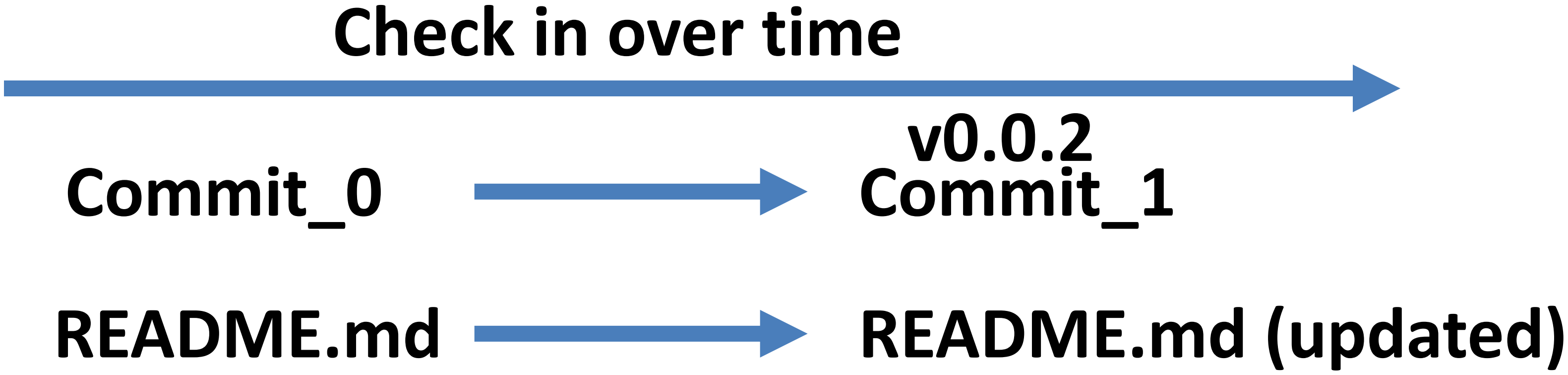
Check in over time

Commit_0 → Commit_1

README.md → README.md (updated)

Want to check out the first version through your working directory?

```
git checkout hash_of_commit_0
```



Check in over time

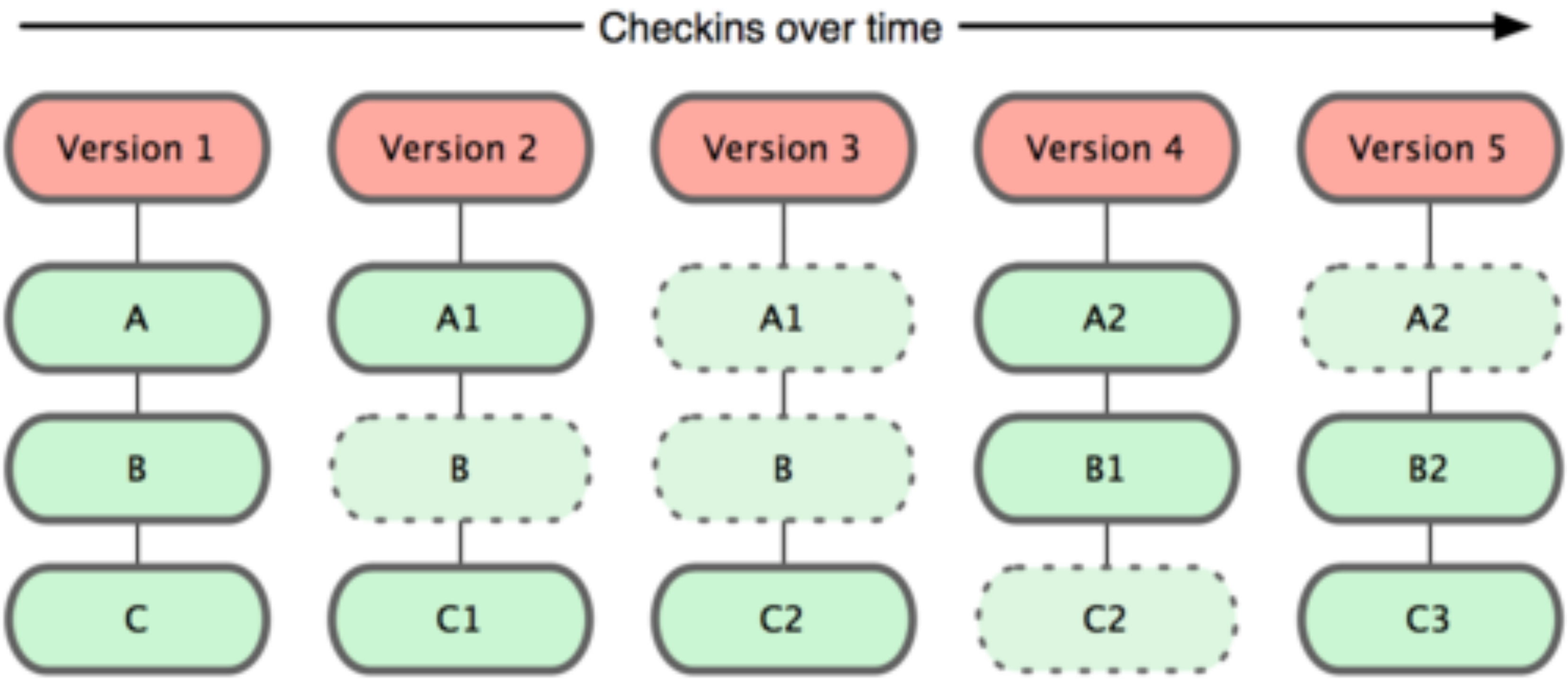
v0.0.2

Commit_0 → Commit_1

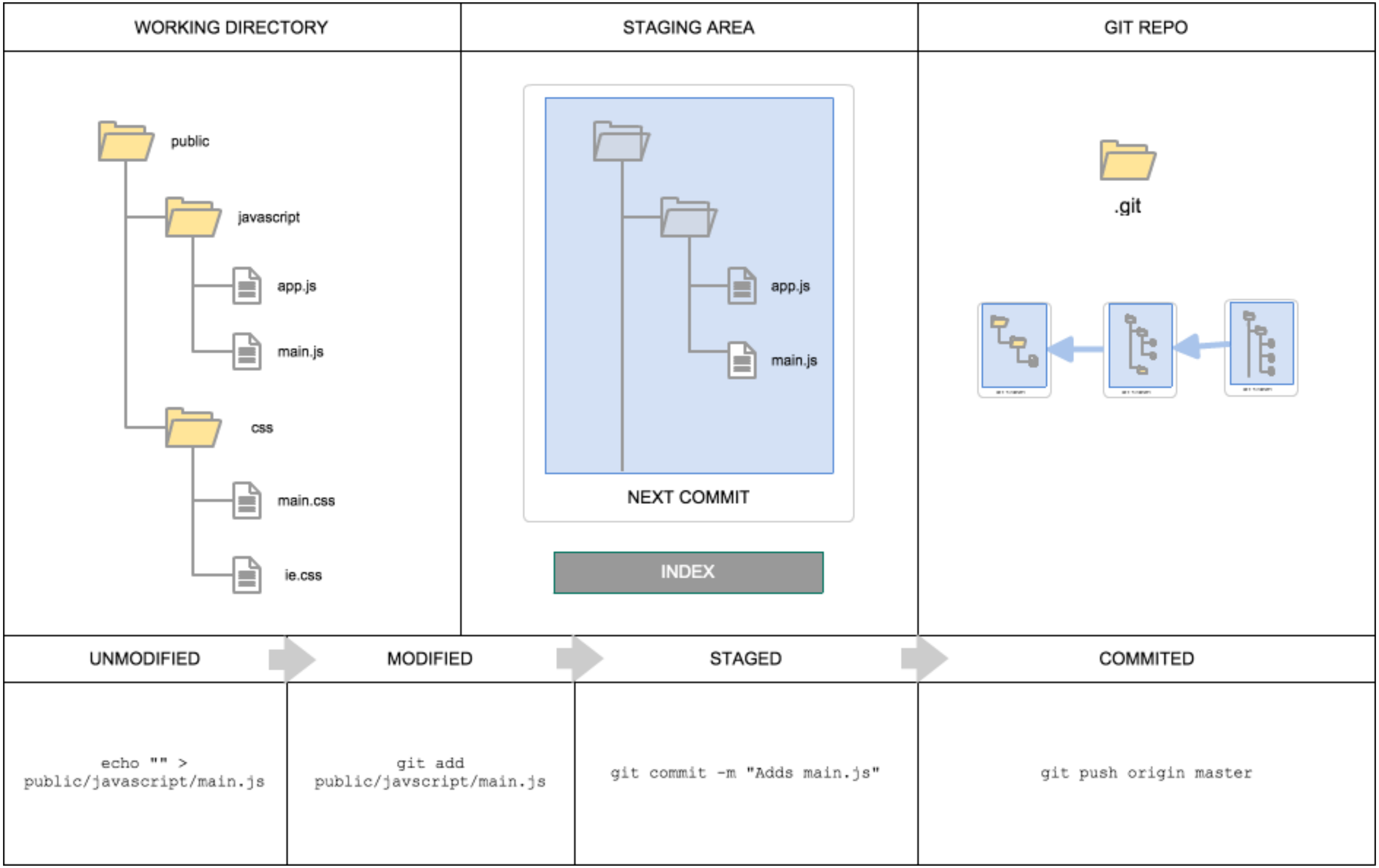
README.md → README.md (updated)

```
git tag v0.0.2
```

Commits


Checkins over time
Version 1
A
B
C
Version 2
A1
B
C1
Version 3
A1
B
C2
Version 4
A2
B1
C2
Version 5
A2
B2
C3

Another example



```
echo "" >
public/javascript/main.js
```

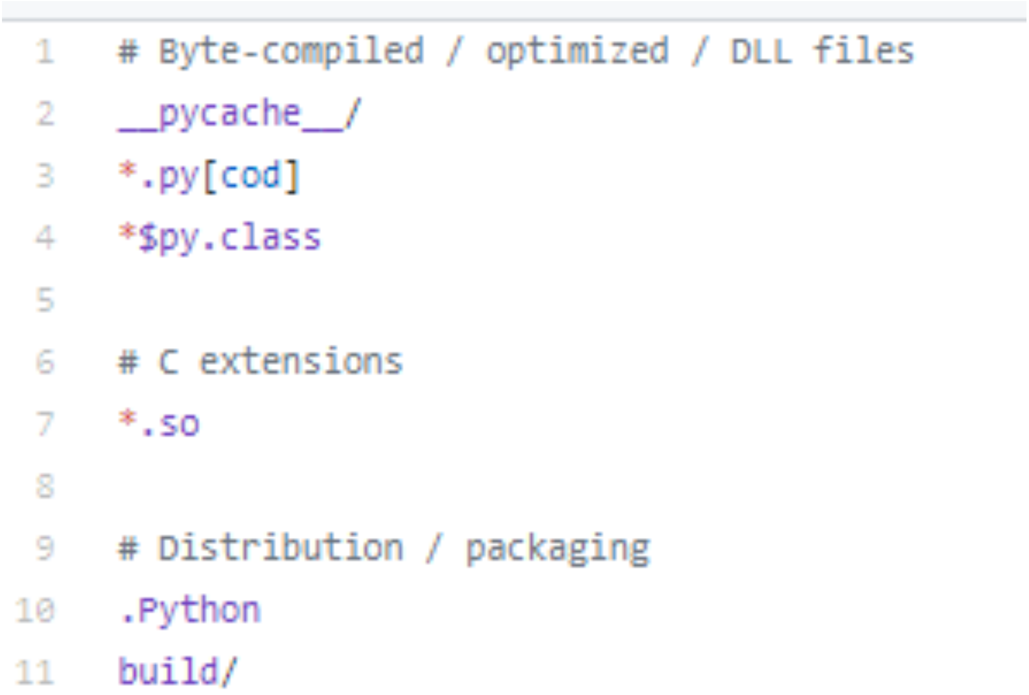
```
git add
public/javscript/main.js
```

```
git commit -m "Adds main.js"
```

```
git push origin master
```

.gitignore file

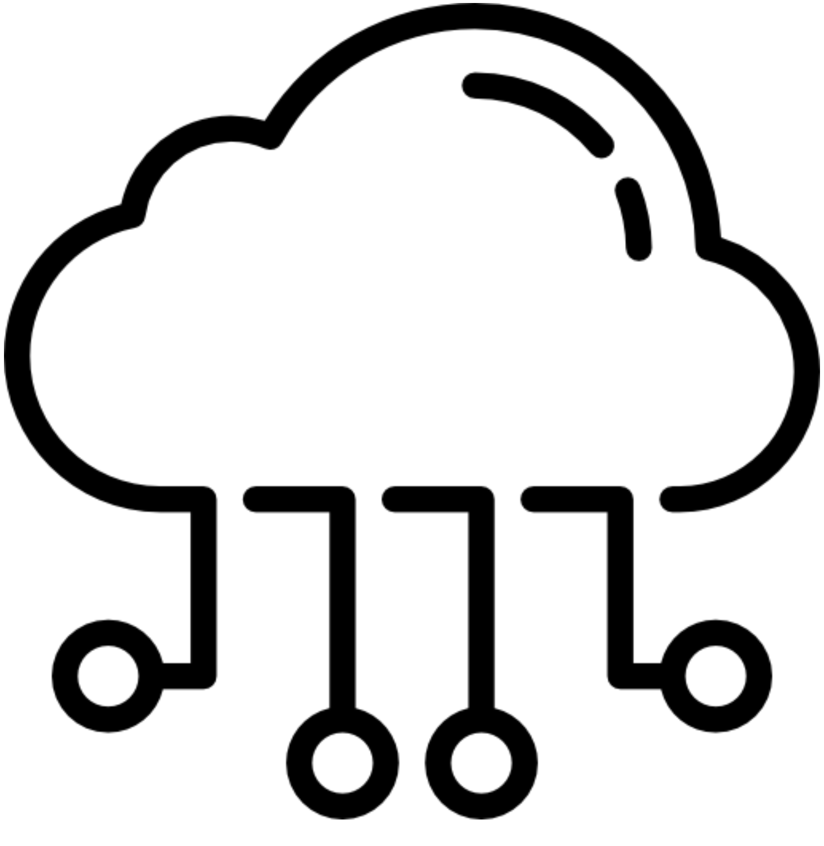
- Rules to exclude the files that you don't want to track
  - E.g. binaries, jars, temporary files, __pycache__

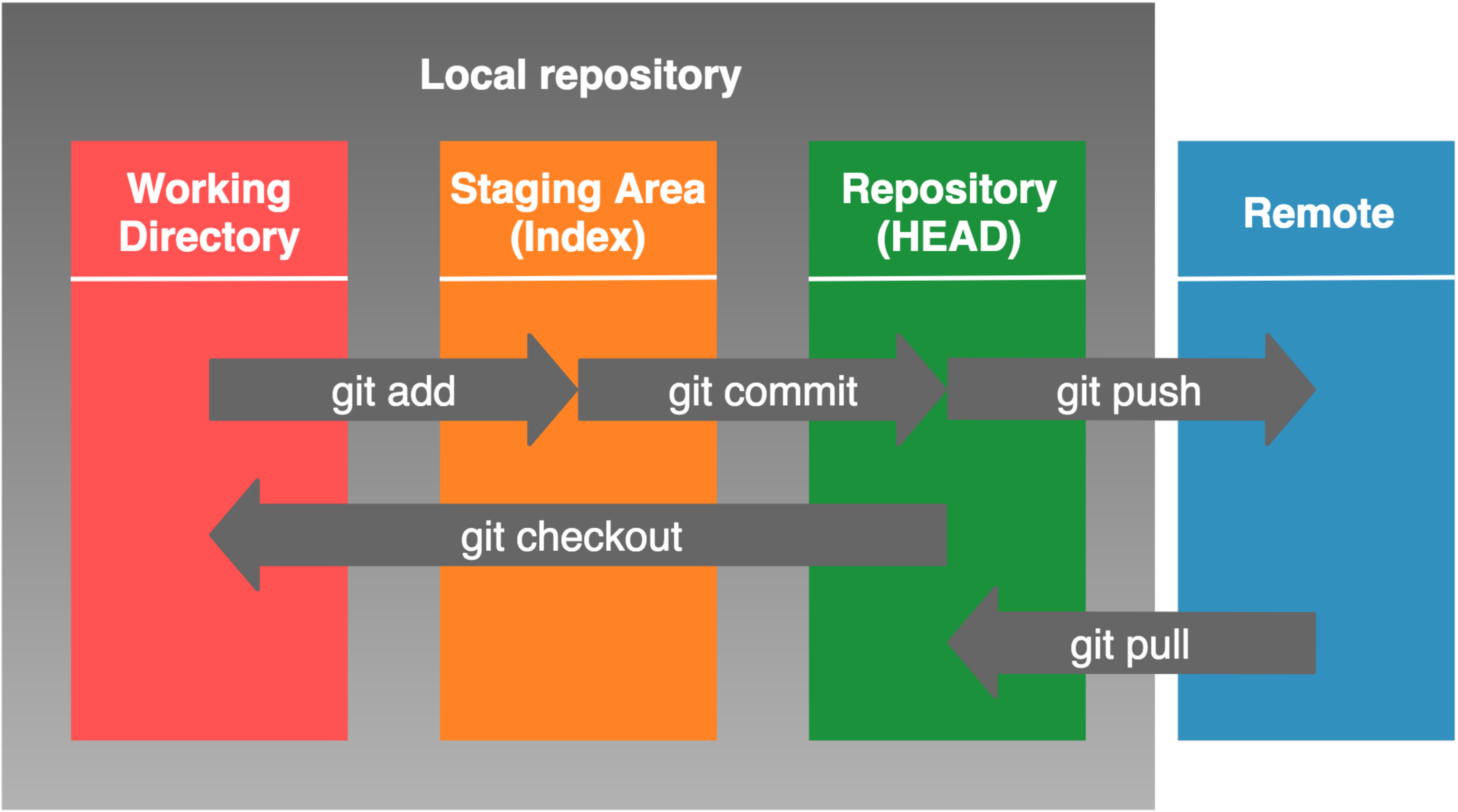


```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
```

GitHub/GitLab/GitBucket

- In short – A remote repository





Local repository

Working Directory

Staging Area (Index)

Repository (HEAD)

Remote

git add

git commit

git push

git checkout

git pull

git clone

- Clone the remote repository into your local directory:

- git clone https://github.com/CISC-CMPE-327/CI-Python

git status

- A summary of your working directory:

```
git status

On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   testfile.txt

no changes added to commit (use "git add" and/or "git commit -a")
```