

# CISC/CMPE 327 Software Quality Assurance

Queen's University, 2019-fall

Lecture #9

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

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

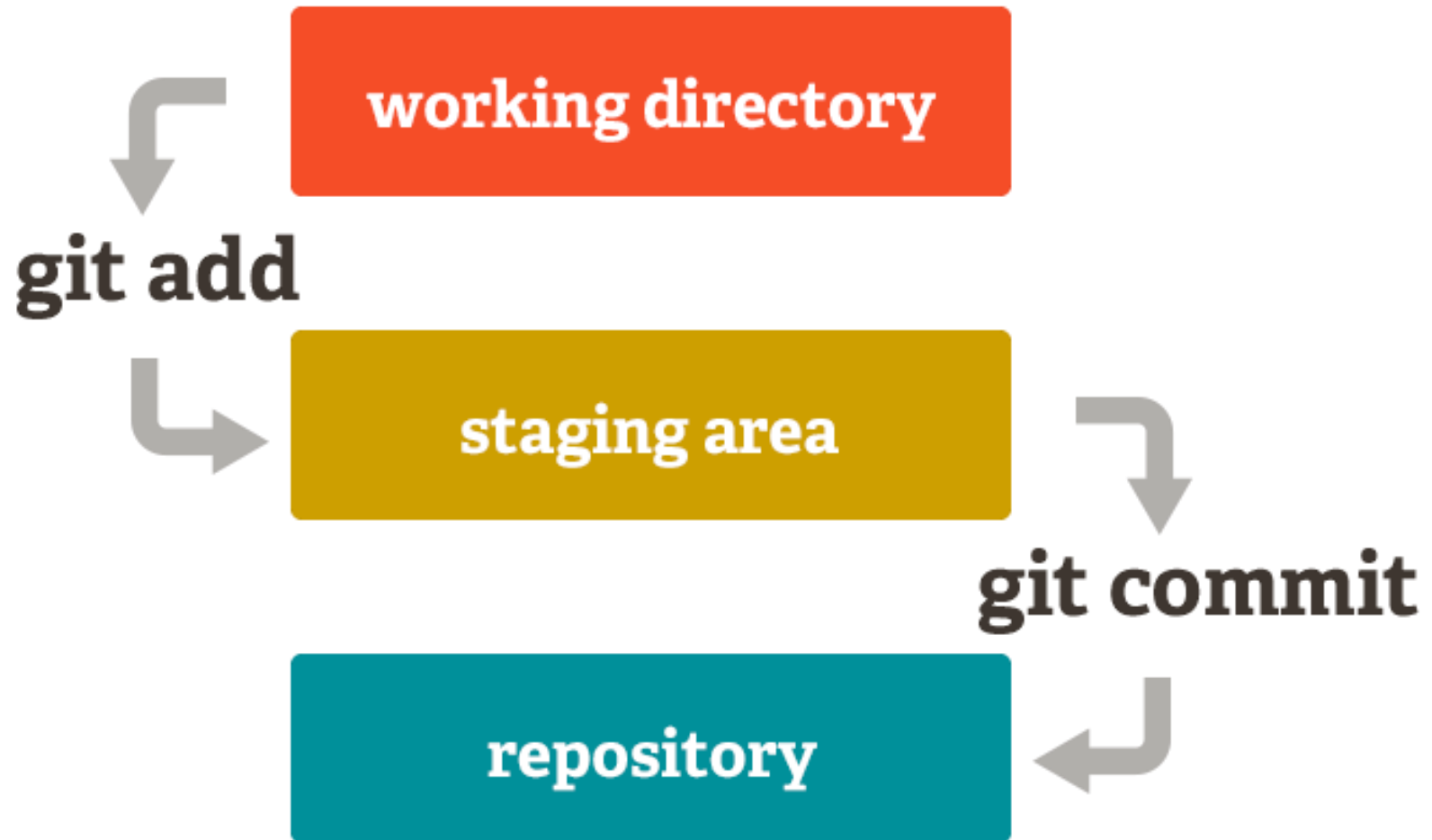
- Create a folder: my-repo
- `cd my-repo`
- Init a repos:
  - `git init`(so everything will be tracked for any modification)

# Add a file

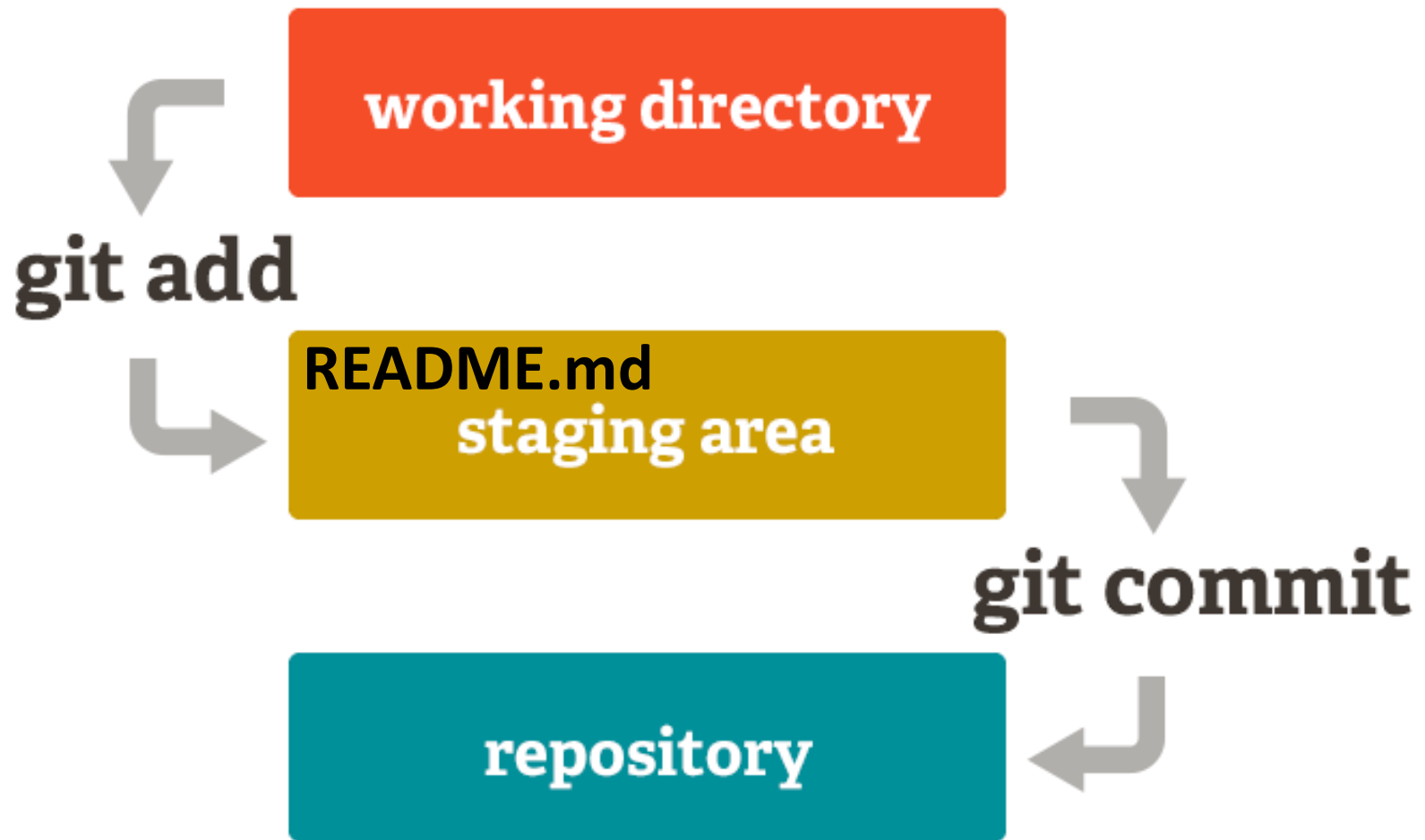
- Create a text file
  - Markdown text file
  - Type 'hello git', saved
- Add all new changes to the repository
  - `git add .`
  - Or you can specify which file to add `git add README.md`
- Create a **commit**:
  - `git commit -m "added a new file"`

# Create a text file & edit

README.md

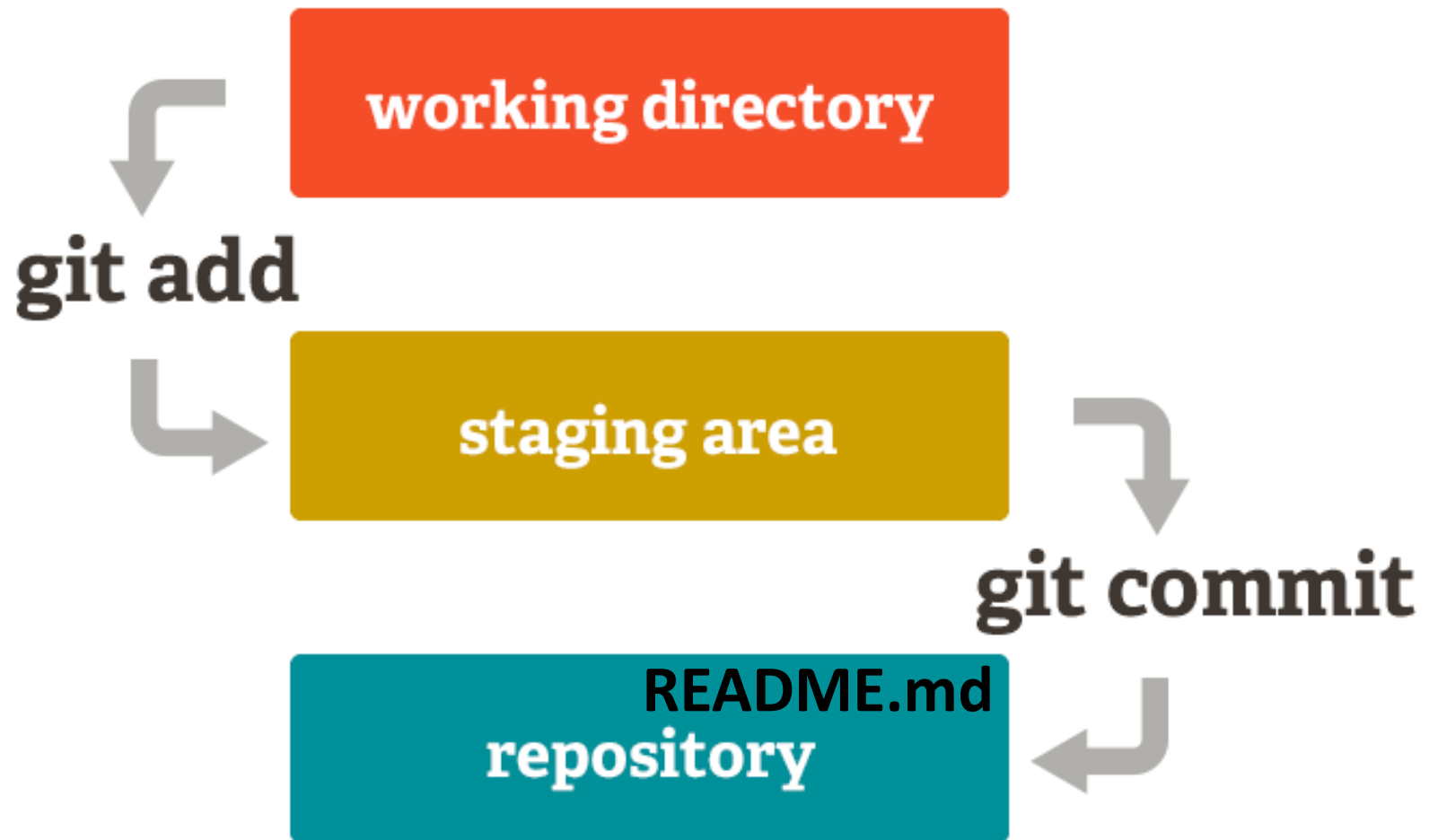


`git add .`





# git commit



**Check in over time**



**Commit\_0**

**README.md**

# Edit again

README.md (updated)

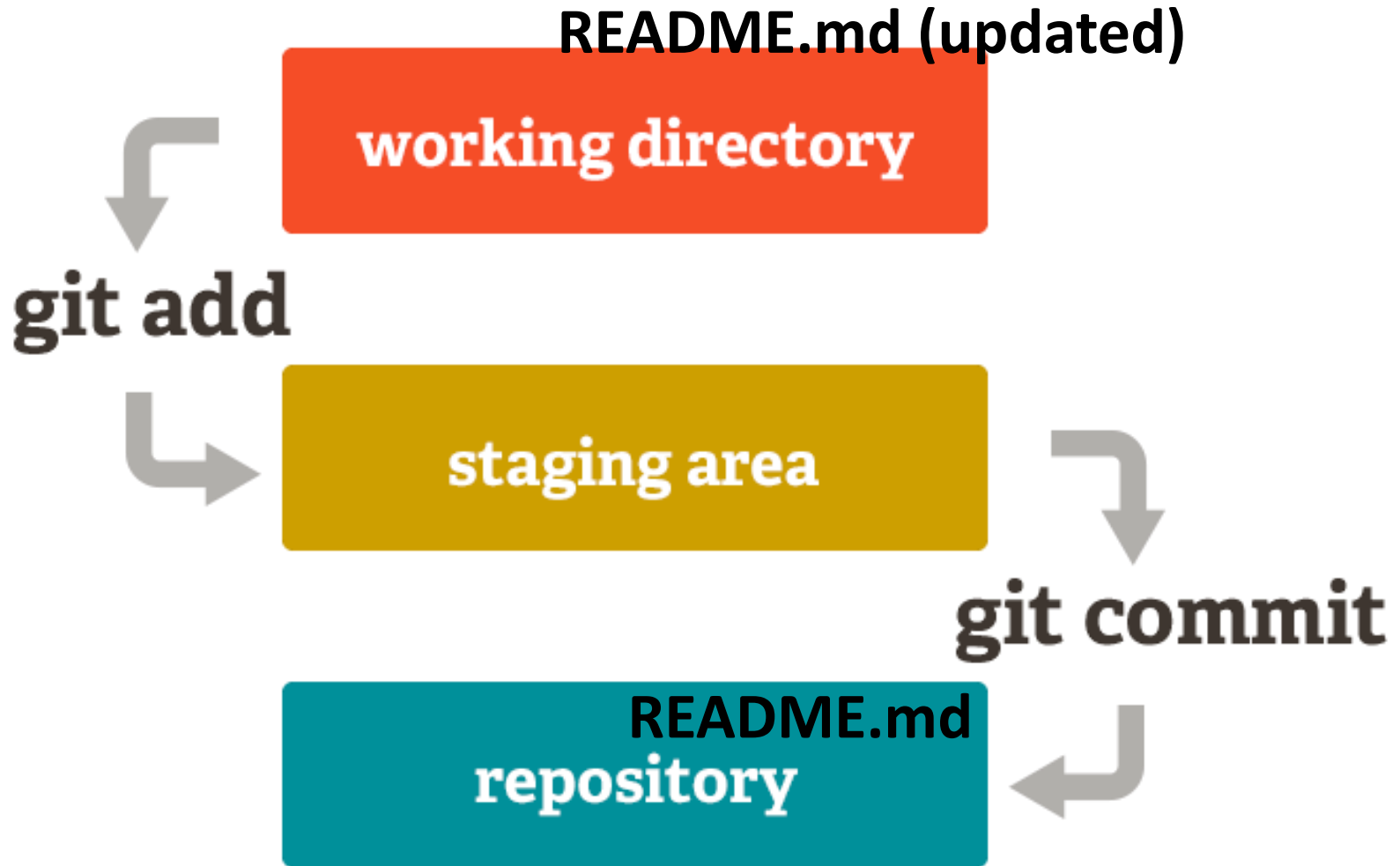
**working directory**

**git add**

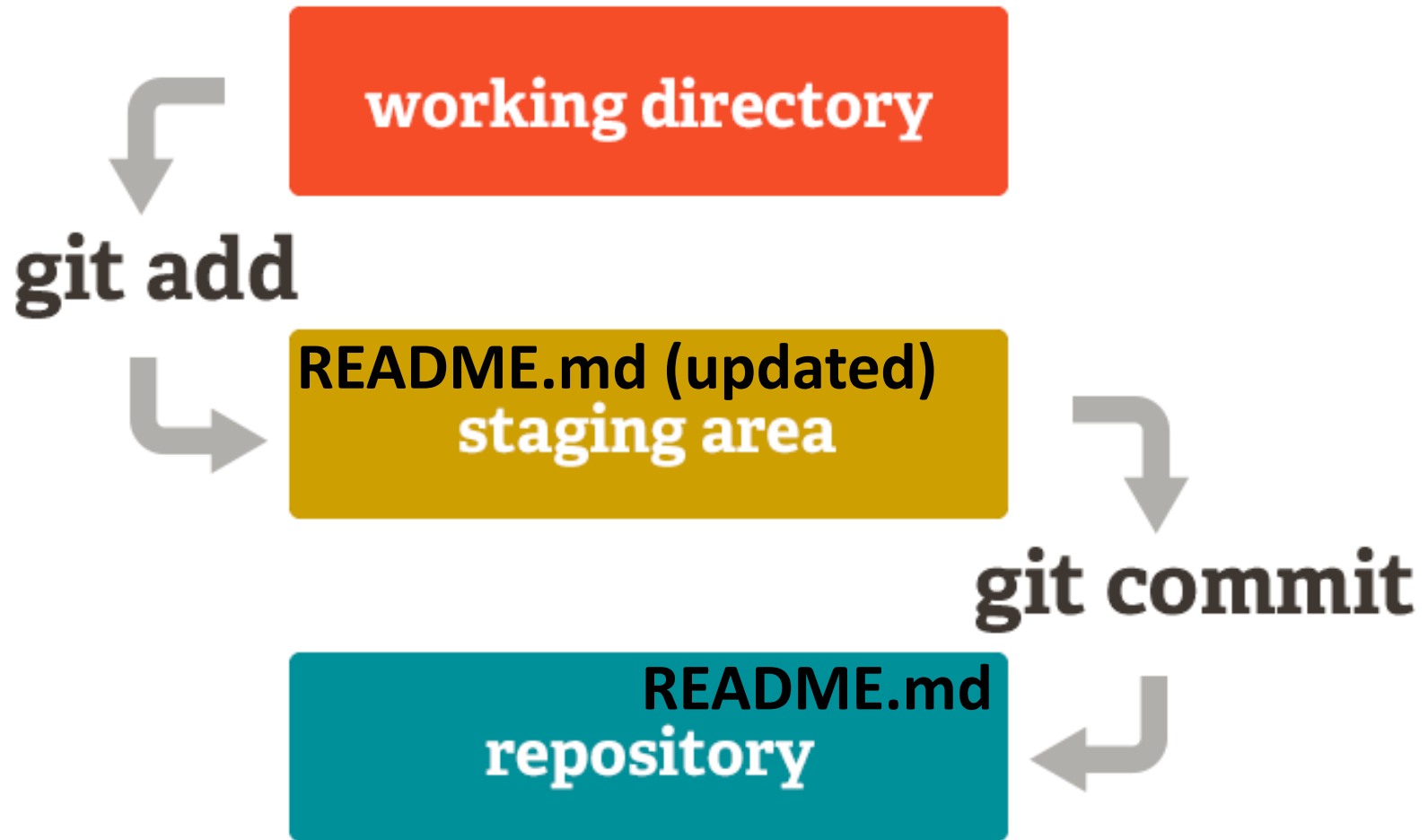
**staging area**

**git commit**

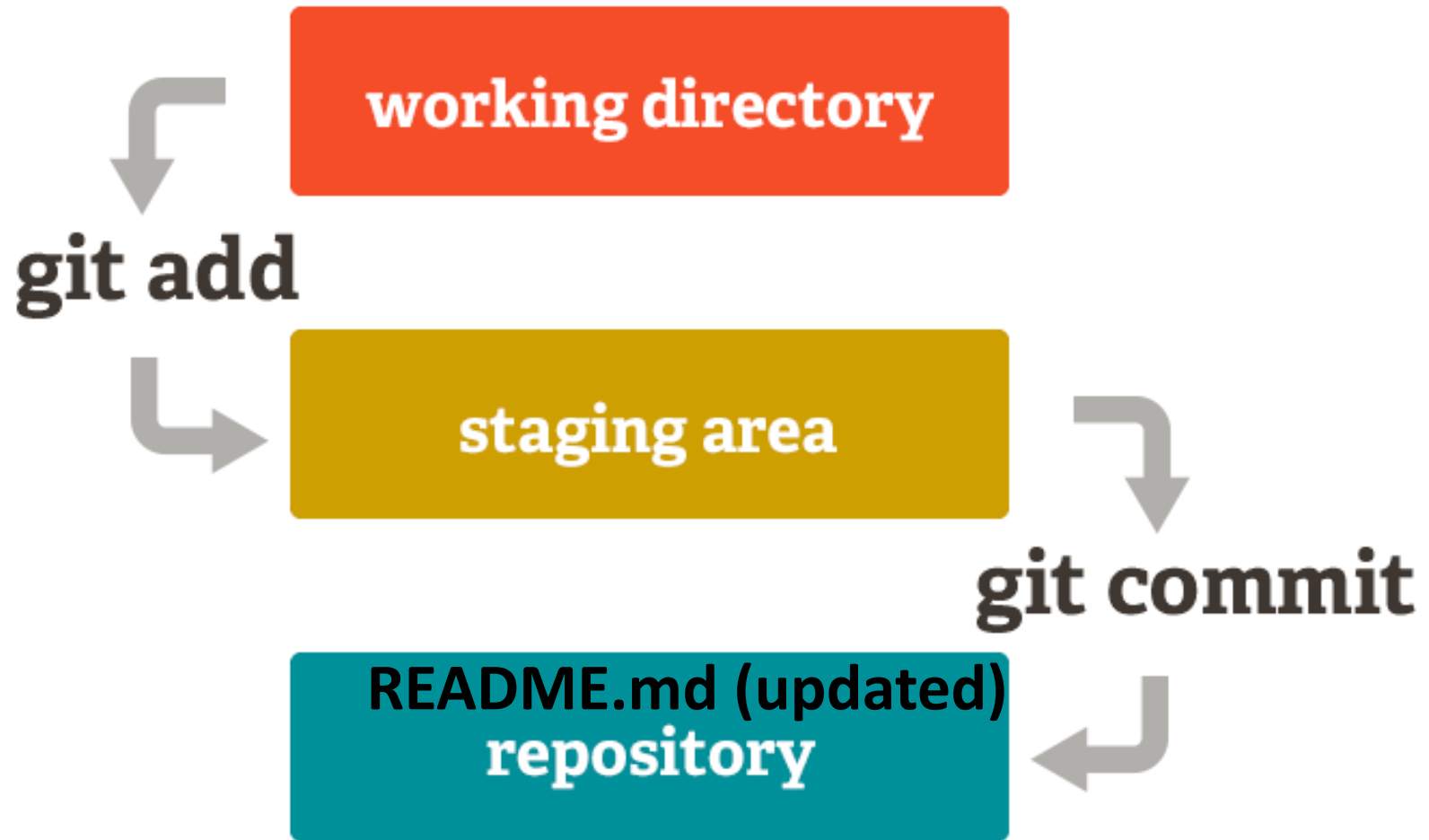
README.md  
**repository**



`git add .`



# git commit



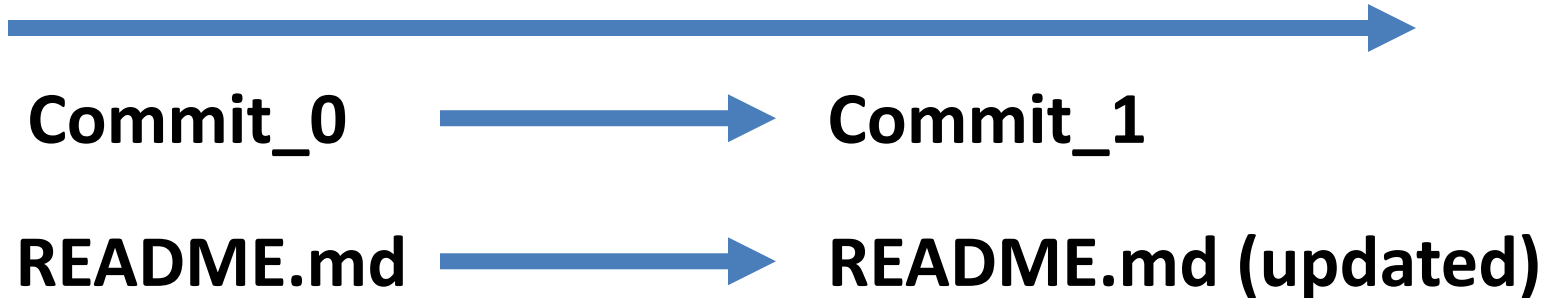
## Check in over time



**Commit\_0**            **Commit\_1**

**README.md**            **README.md (updated)**

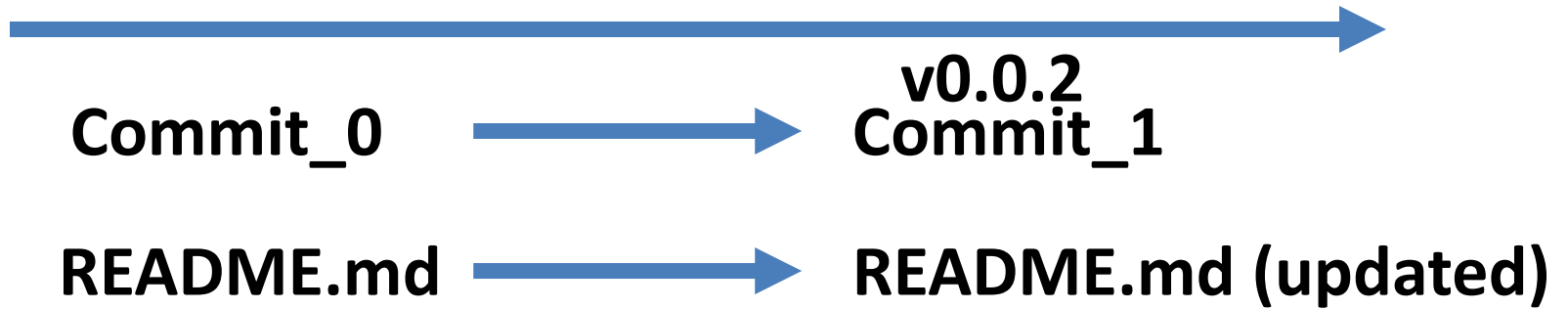
## Check in over time



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

**git checkout hash\_of\_commit\_0**

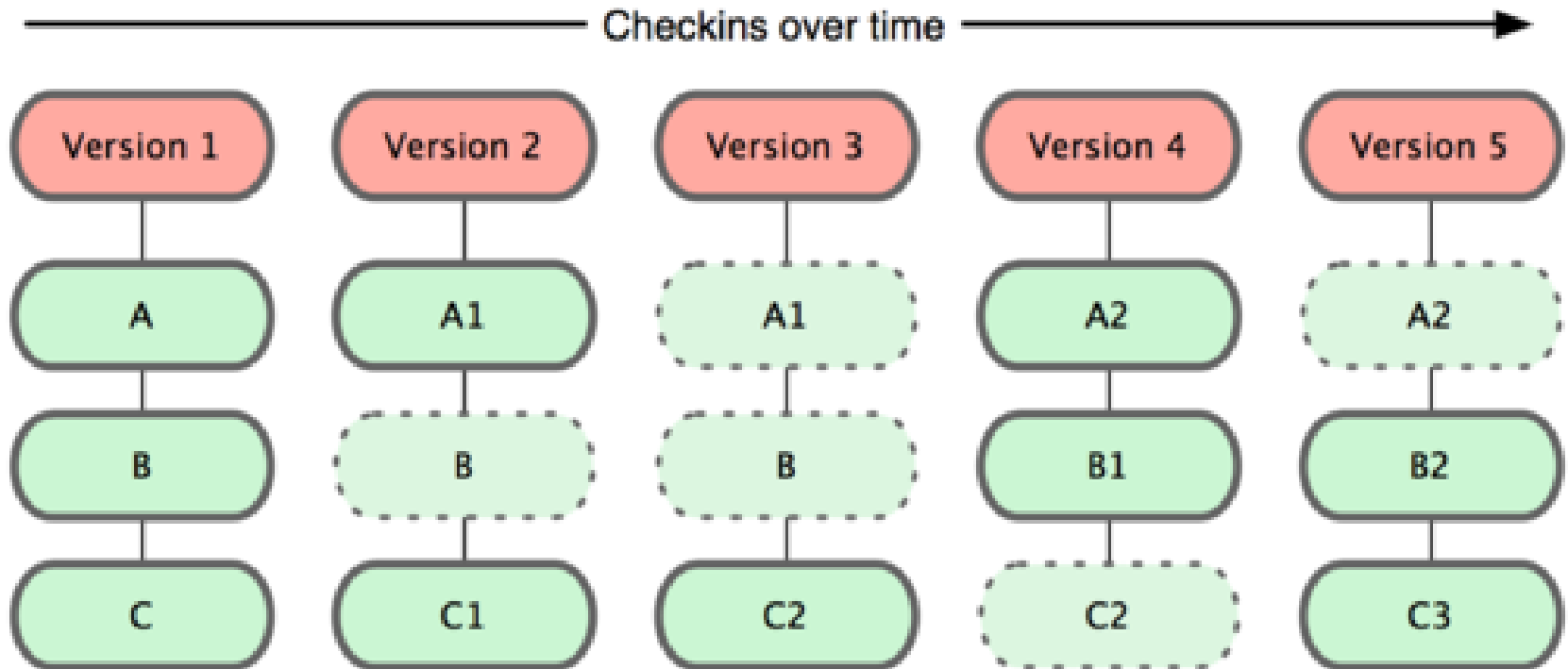
## Check in over time



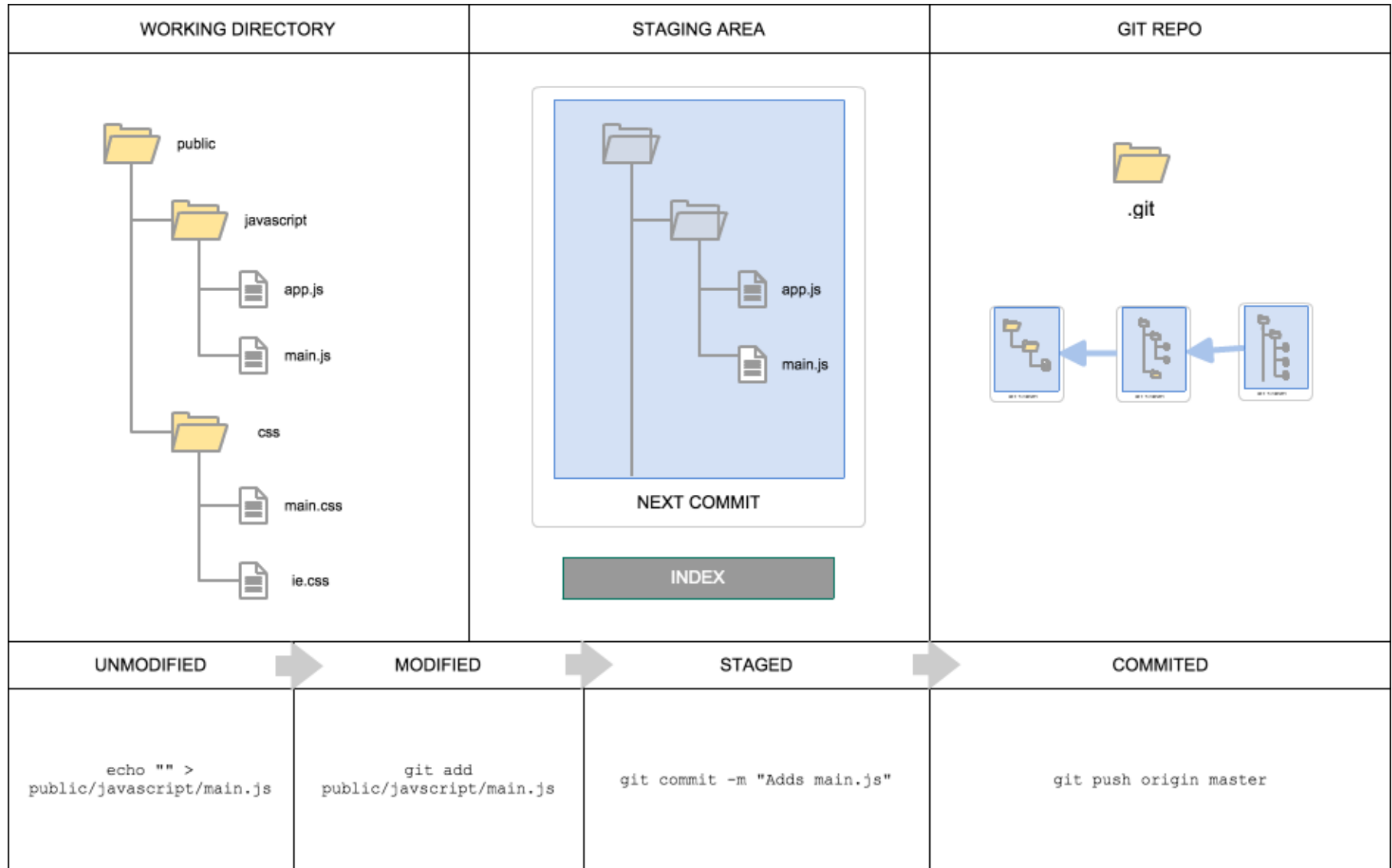
**git tag v0.0.2**



# Commits



# Another example



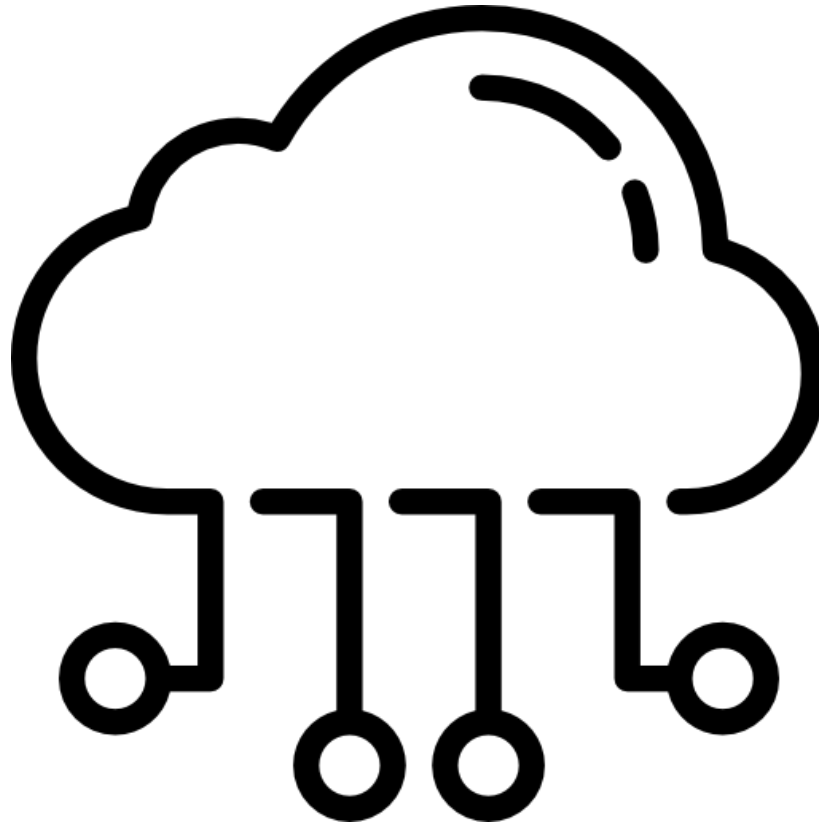
# .gitignore file

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

```
1  # Byte-compiled / optimized / DLL files
2  __pycache__/
3  *.py[cod]
4  *$py.class
5
6  # C extensions
7  *.so
8
9  # Distribution / packaging
10 .Python
11 build/
```

# GitHub/GitLab/GitBucket

- In short – A remote repository



## Local repository

**Working  
Directory**

**Staging Area  
(Index)**

**Repository  
(HEAD)**

**Remote**

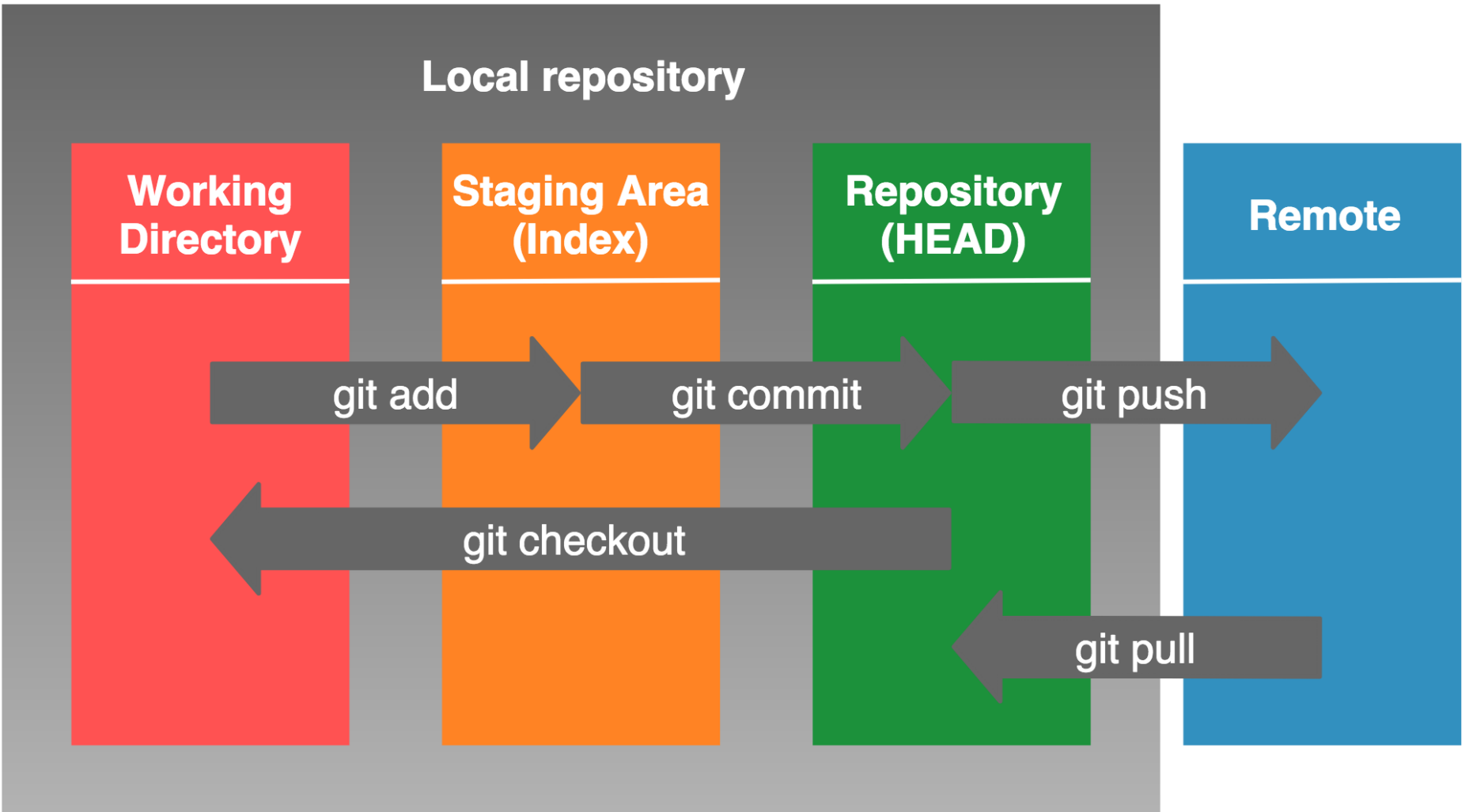
git add

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")