

# Git

- **Distributed version control systems**



**git**

# Local Storage

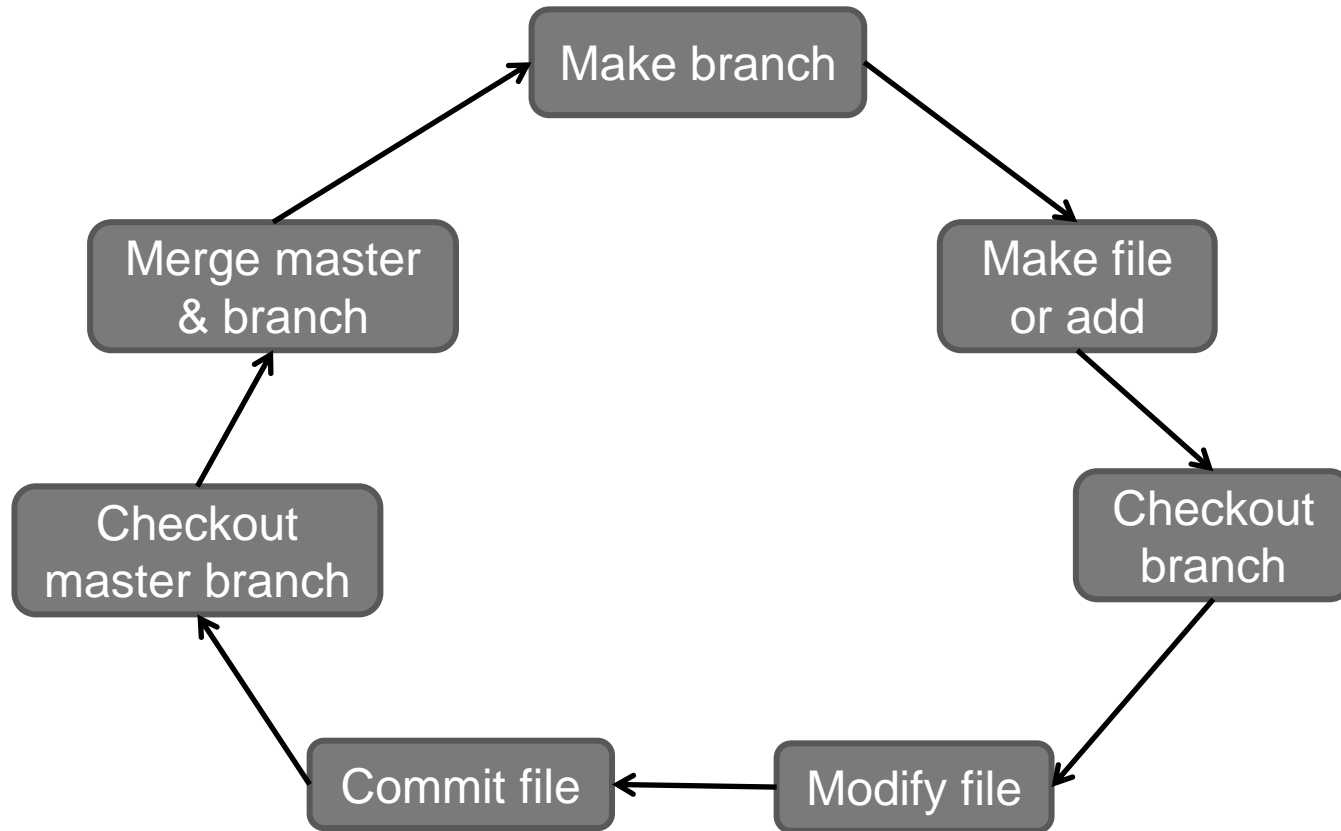
## ■ Basic command

```
$ git init // Make new local storage
$ git add "filename" // Store file to storage
$ git commit // Store fixed file to storage
$ git status // Check status
```

## ■ Branch command

```
$ git branch "name" // Make branch with "name"
$ git checkout "branchname" // Change working branch
$ git merge "branchname" // Merge working branch &
branch of branchname
```

# Flow Chart (Branch)



# Example

- **Basic command**

```
$ mkdir git_tutorial && cd git_tutorial
$ git init
$ vi hello.c
$ git status
$ git add hello.c
$ git status
$ git commit // Write explanation of commit file
$ git config -global core.editor vim
    // Change commit editor to vim
$ git commit
$ vi hello.c
$ git status
$ git commit -a // (or git add hello.c)
```

# Example

- **Branch command**

```
$ git branch // Check list of branches
$ git branch sw3 // Working branch has (*)
$ git checkout sw3 // git checkout -b sw3
$ vi hello.c
$ git status
$ git commit -a // git commit -am "abcdefg"
// Cannot use "" in commit message
$ git checkout master
$ cat hello.c
$ git merge sw3
$ cat hello.c
```

# .gitignore

- Ignore some files in commit

```
$ touch .gitignore
$ ls && ls -al
$ vi aa.a
$ git status
$ vi .gitignore

# .gitignore
*.log
!sw3.log
/home
sw3/

$ git commit -a
$ git add .gitignore && git commit -m "add ignore"
```

# Conflict

- **Resolve conflict of file between branches**

```
$ git checkout master
$ git merge sw3
    // hello.c is different between branches
$ git checkout sw3 // error
    // Need to resolve current error first
$ vi hello.c
    // << HEAD / >> branch
$ git status // both modified
$ git commit -am "conflict resolved"
```

# Git Log

```
$ git log -p // Show all changes in commit
$ git log --word-diff // Show with diff command
$ git log -stat / --name-only / --graph
    / --relative-date
```



# GitHub

- **Remote storage (place of co-work)**
- **Procedure**
  - <https://github.com> & create account with mail
  - Repository
    - Create / fork
    - Public / private
      - GitHub student pack

# GitHub

## ▪ Basic Command

```
$ git clone // Copy remote storage to local
$ git remote // Link local storage & remote
$ git push // Push local storage contents to remote
$ git fetch
    // Compare difference between local & remote
$ git pull // <-> git push
```

# Example

## ■ Basic Command

```
$ git clone http://github.com/username/tutorial.git
$ ls && rm -r tutorial
$ git remote add tutorial http://github...
    // byname
$ git remote -v
$ git push origin -all
    // git push "byname" "local branchname"
    // ! [rejected] master -> master (fetch first)
    // Need to update change of remote to local
    // $ git pull https://github...
$ git branch -a
$ git fetch tutorial
    // Get commit log of remote storage
$ git merge
```