github

# github

It is a source code management service for software development projects. It is an indispensable service for developers as it has simple bug management, SNS functions, and collaboration.

# Basic knowledge

**Knowledge 1: commit and push**

* commit : adding files or saving changes to storage.
* push : adding files or uploading changes to remote storage.

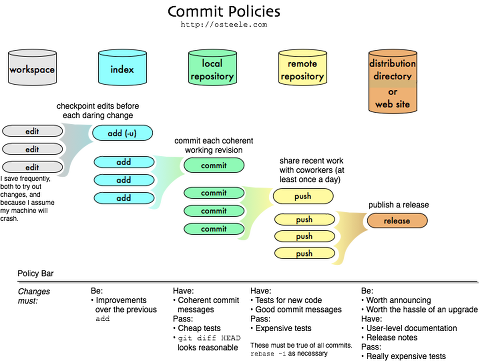
**Knowledge 2: Local and Remote storage**

* Local storage: where you store files or directories on your computer.
* Remote storage: a place to store files or directories on a network, such as a server.

**Knowledge 3: branch**

* Multiple version management methods performed in parallel to add new features and modify bugs while maintaining the currently released version.

# github terms



**<GitHub Commit Policy>**

* Repository: space to track all changes made by the operator.
* Working Tree: the current viewpoint of the worker looking at the storage at a point in time.
* Checkout: the operation of changing the operator’s work tree to match a specific point in time in the storage.
* Staging area: where to prepare a commit before commiting to the storage.
* Branch: it means one development line.
* Master: master branch is a copy of HEAD in the storage you have copied.
* Origin: the name that git basically uses to refer to the storage it copied.

# github use

* Github initial setting.
* Create or duplicate a storage on Github (git clone)
* Create and edit files.
* Create / change / elete file to git index (git add)
* Commit the result of the change to the local repository (git commit)
* Push the local storage and reflect it in the remote storage (git push)

# Github initial setting

* Install Git bash
* Open the Git bash and write an initial setting command at the terminal.
* ‘git config –global user.name operator name’: set operator name.
* ‘git config –global user.email operator email’: operator email settings.
* ‘git config –global –list’: check setting value.

# create or duplicate a storage on Github

* Create Repository.
* Description: describe the storage.
* Public can be used all / Private can only be created by paid members.

텍스트이(가) 표시된 사진

자동 생성된 설명

**<respository Creation Screen>**

* Create and edit a file and go to the folder where it is located.
* Enter the ‘git init’ command to the terminal in that folder (git init: executing this command with the command to create a new git repository convert
* 해당 폴더에 터미널에 ‘git init’ 명령어 입력하기 (git init: if you execute this command with a command to create a new git repository, convert the current directory to git repository.
* ‘git clone repository address’: stores information in the repository in a local llocation.

# create / change / delete file to git index

* ‘git add .’: a command to add all current files to the index. An index is a location where changes are temporarily saved to prepare to commit to the repository.

# commit the result of the change to the local repository.

* ‘git commit -m “content about commit”: a command to commit files added to the index. Commit is the operation of recording the addition or change of a file or diredctory into a repository.
* ‘git status’: a command to check the status of adding files to the storage.
* ‘git remote add origin https://github.com/~~ : add remote storage information before reflecting it in remote storage. It is a command to add this information because there is no information connecting the local and remote strage, although you have created a repository and have to push it from the internal storage.

# push local storage and relfect it in remote storage

* ‘git push origin master’: a command to reflect changes in local storage to remote storage in GitHub.
* ‘git pull origin master’: a command to get what someone else has worked on.

텍스트이(가) 표시된 사진

자동 생성된 설명

**<A collection of instructions to upload the document in remote storage>**

**텍스트, 스크린샷, 모니터, 실내이(가) 표시된 사진

자동 생성된 설명**

**<The result of uploading a document to a remote respository>**

# using branch

* Creating and moving branches.
* ‘git branch’: a command to know the current branch list. (if you don’t have a branch, you’ll get **\*master** results.)
* ‘git branch subdir01’: a command to create a branch.
* ‘git checkout subdir01’: a command to move to subdir01 branch. (Branch changes to **\*subdir01**)
* Development work at a branch.
* If you change a point, no matter what change you make, there is no change in other points.
* Create a ‘test2.html’ file and push it to the branch.
* Push it to the branch.
* ‘git add test2.html’
* ‘git commit -m “second commit subdir01”
* ‘git push origin subdir01’

텍스트, 전자기기, 측정기이(가) 표시된 사진

자동 생성된 설명

**<A collection of instructions to upload documents to subdir01>**

텍스트, 모니터, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

**<The result of uploading the document to subdir01>**

* Merge branches.
* ‘git checkout master’: change the branch to master
* ‘git merge subdir01’: subdir01 merges branch result.
* ‘git push origin master’: push the merged result.

텍스트, 스크린샷, 모니터, 검은색이(가) 표시된 사진

자동 생성된 설명

* Delete the branch.
* ‘git branch -d subdir01’: delete the branch of subdir01.