Git & GitHub

1. **What is GitHub?**
   - Version control system
   - Keeps track of new/old version of documents
   - Manages/stores set of files

2. **What is repository?**
   - Folder where the files are saved and
   - It may contain single, collections of files, or single projects.

3. **What is Remote & Local Repository?**
   - *Remote Repository:* Host on server(GITHUB) Our changes go from local to remote repo
   - *Local Repository:* Typically, on your computer -Our changes are done here consist of Working Directory, index and HEAD

4. **What are Git commands?**
   - **Add**: add to staging area
   - **Commit**: add from working directory and local repo
   - **Push**: add to remote repo
   - **Pull**: take changes from remote to working directory
   - Clone with url: clones url into directory
   - Git version: give you version of git
   - Git status: shows you what branch you're on, any changed files that aren't tracked
     - Origin: name of remote
     - Master: name of branch
   - Git add:
     - Adding to staging area
     - Recursive add
     - Adds everything
   - **git commit -m**: "message will apply for all files"
   - **git push**: origin nameOfBranch
   - **git ignore**:
     - Notepad.gitignore → In the notepad add files you don't want to add to staging area
     - YOU MUST PUSH THE .GITIGNORE FILE TO REPO IN ORDER FOR THE FILES YOU WANT TO IGNORE TO BE IGNORED ON GIT
     - Some files don't matter and shouldn't be pushed to git
   - Remove file-git → GIT ADD REMOVE POM → COMMIT THAT → AND PUSH Creating own branch
   - checkout branch -git → Git checkout -b nameOfBranch master

5. **How do I use Git in terminal?**
   - create new repo-git
   - echo "# SqlMentor" >> README.md

     ```
     git init
     git add README.md
     git commit -m "first commit"
     git remote add origin
     https://github.com/Andylam224/SqlMentor.git git push -u origin
     master
     ```
   - push an existing repo-git
     - git remote add origin https://github.com/Andylam224/SqlMentor.git
       git push -u origin master
   - Default editor

6. GIT Commands?

```
git init
git add .
git commit -m "my comment"
----------------------------------------------------------------
git info
git log
git push -u origin master
git push
---------------------------------------------------------------
git init
git remote add origin URL    //  copy paste https:// url to URL place
git add src/     // if i want to add only this folder
git commit -m "my comment"
git log
git push -u origin master
```

7. Returning to the latest version?

```
// we need to type both of them
git fetch origin
git reset --hard origin/master
```

8. Adding couple files in one time?

```
git add file1 file2 file3  //
```

9. GIT Branch branches?

```
git rm file.java
git commit -m "removing"
git push origin master

git branch BranchName                        :- Creating branch
git branch                                   :- checking branch master
git checkout  BranchName                     :- name is a branch name where you want to switch
git branch -d          BranchName            :- deleting brach on local
git push origin : deletedBranchName          :- deleting Branch deteled on local(intellij) from Remote(gitHub WebSite)
git branch -a                                :- Cheking all branchs even deleted on Local (but not in remote)
git checkout -b BranchName                   :- Creating branch and Switching to the new branch
git merge BranchName                         :- Merging branch
git push --set-upstream origin BranchName    :- Pushing branch to remote (gitHub WebSite) from local (intellij)
git fetch origin BranchName                  :- Pulling branch to local (intellij) from remote(github WebSite)

git push origin branch1:branch2
git pull origin branch1:branch2
```

10. **Merging branch with master**

--go to your second branch do next steps

```
git add .
git commit -m "your comment"
```

--go to your master branch

```
git merge "branchName"
```

--if its not merging we need to do git commit -m "comment" again from master branch

11. **Merging new Branch from GitHub repository to Local master with Changes in code GIT Commands?**

```
git fetch origin BranchName
git checkout BranchName
git branch                    // you must be in new branch
git checkout master
git branch                    // you must be in master branch
git merge BranchName          // it will merge. if any conflict you need to fix it, if you don't have merge conflict it will pas

git branch -a                             // we are able to see both local and remote branches
git clone URL of what you want to clone                     // and after copy link
git fetch                     // ctobi obnovit obnovleniya v glavnom
git merge
git log --graph                           // showing what's commited and happining
git log --graph -- online     // showing in one line what's happining
```

--if you have conflict go to project right click -> git - > resolve conflict -> merge :

wq   **and escape**

12. **CHECK THE GITHUB URL?**

```
git remote -v
git config --get remote.origin.url
git remote show origin
git config --get remote.origin.url
```

13. **What is pull request?**

git merge fetch_head --allow-unrelated-histories

- Resolved an issue for pulling a non-fast-forward issue
- Press escape then
    - Press shift ":x!" ⟶ Saves and exit
    - ":q!" ⟶ No save and exit

14. **What is pull request?**

- Git merge fetch_head --allow-unrelated-histories
    - Resolved an issue for pulling an non-fast-forward issue

15. **How do you resolve conflict on git?**

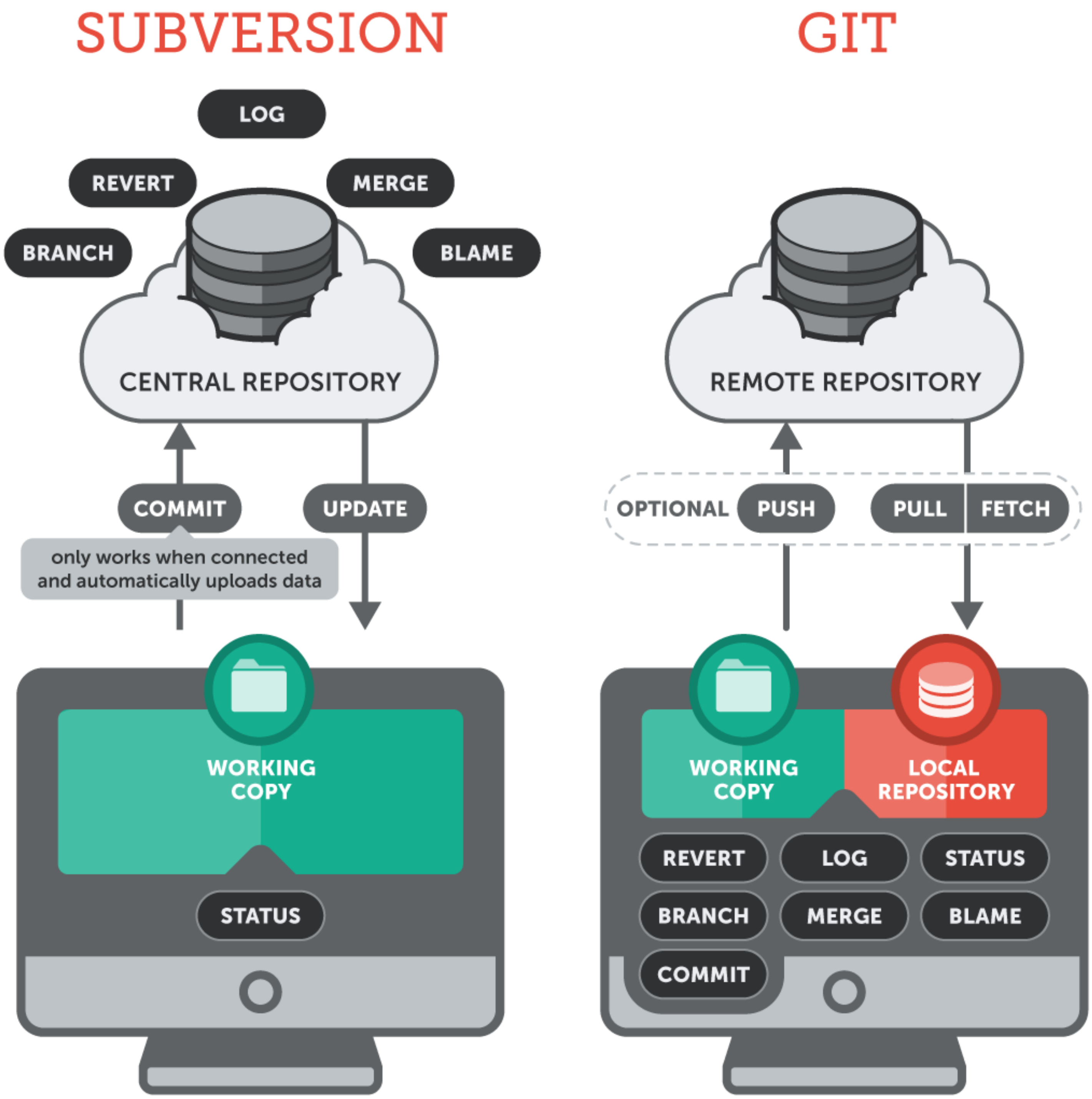
- your repository ⟶ cd ~/<repo_directory>
- Pull recent version repo ⟶ git pull
- Checkout the source branch ⟶ git checkout <feature_branch>
- Pull destination branch into the source branch ⟶ git pull origin <destination_branch>
- Fix conflicts and then commit the result.

16. Git vs SVN commands

Comparison table of Git-Subversion commands

Command	Operation	Subversion
git clone	Copy a repository	**svn** checkout
git commit	Record changes to file history	**svn** commit
git show	View commit details	**svn** cat
git status	Confirm status	**svn** status
git diff	Check differences	**svn** diff
git log	Check log	**svn** log
git add	Addition	**svn** add
git mv	Move	**svn** mv
git rm	Delete	**svn** rm
git checkout	Cancel change	**svn** revert[1]
git reset	Cancel change	**svn** revert[1]
git branch	Make a branch	**svn** copy[2]
git checkout	Switch branch	**svn** switch
git merge	Merge	**svn** merge
git tag	Create a tag	**svn** copy[2]
git pull	Update	**svn** update
git fetch	Update	**svn** update
git push	It is reflected on the remote	**svn** commit[3]
git ignore	Ignore file list	**svn** ignore

1. Revert in SVN is the cancel of change but Revert in Git is the commit for negation. The meanings of Revert are different.
2. Branch and tag are the same in the structure in SVN, but they are clearly different in Git
3. SVN does not have the concept of local repository/remote repository, accordingly, commit is directly reflected in the remote. However, Git has different reflecting methods for reflecting to the local repository and for reflecting to the remote repository.


SUBVERSION
LOG
REVERT
MERGE
BRANCH
BLAME
CENTRAL REPOSITORY
COMMIT
UPDATE
only works when connected and automatically uploads data
WORKING COPY
STATUS
GIT
REMOTE REPOSITORY
OPTIONAL
PUSH
PULL
FETCH
WORKING COPY
LOCAL REPOSITORY
REVERT
LOG
STATUS
BRANCH
MERGE
BLAME
COMMIT