GIT-GITHUB

CHAPTER 1

- Git nedir?

  Neredeyse tüm dosya tipleri ile kullanılabilen bir versiyon kontrol sistemi. Versiyonlar arası geçiş yapılabilir. Dallanma ile beraber proje geliştirilebilir. Github, git ile entegre bir repository yönetim alanıdır.

- Git projesi oluşturma.
  git init
  nano topla.py
  git status
  git add . (git add topla.py)
  git commit -m “topla eklendi”
  git status
  git log

- DIFF
  nano carpma.py
  nano bolme.py
  git diff
  git diff carpma.py
  git diff bolme.py

- REMOVE
  - 1
    Delete carpma.py
    git add carpma.py
  - 2
    git rm -r(for folder) carpma.py

  git commit -m “carpma.py silindi”

- RENAME
  - 1
    Rename carpma.py → carpma.txt
    git add carpma.txt
  - 2
    git mv -r(for folder) carpma.py

  git commit -m “carpma.py → carpma.txt”

CHAPTER 2

- GO BACK PREVIOUS VERSION(WORKING AREA)
  mkdir proje
  cd proje
  git init
  nano index.py
  git commit -m “v1”
  nano index.py(change content of file or delete file)
  git status
  git checkout -- index.py
  git status

- GO BACK PREVIOUS VERSION(STAGING AREA)
  mkdir proje
  cd proje
  git init
  nano index.py
  git commit -m “v1”
  nano index.py(change content of file or delete file)
  git status
  git add index.py
  git reset HEAD index.py
  git status
  git checkout -- index.py
  git status

CHAPTER 3

- CHANGE VERSION(V1 -> V2 -> V3 -> V2(COPY))

  mkdir proje
  cd proje
  git init

  nano v1.py
  git add v1.py
  git commit -m “v1”

  nano v2.py
  git add v2.py
  git commit -m “v2”

  nano v3.py
  git add v3.py
  git commit -m “v3”

  git rm v1.py v2.py
  git commit -m “v4”

  git log
  git checkout <hash_code_of_version(commit)> -- . (convert all files in specified version with .(dot). If specific file name is written, only that file is converted.)

CHAPTER 4

- PUSH GIT PROJECT TO GITHUB
  git init
  nano d1.py d2.py d3.py
  git add .
  git commit -m “v1”

  New empty repository on github

  cd project/path
  git init
  git remote add githubrepo https://github.com/…
  git remote
  git push -u githubrepo master

  git rm d1.py
  git commit -m “v2”
  git push -u githubrepo master

CHAPTER 5

- GITIGNORE - https://github.com/github/gitignore
  git init
  touch d1.py d2.py d3.py
  git add .
  git commit -m “v1”
  nano d1.py
  git status
  cat >> .gitignore (d1.py)
  cat .gitignore
  git status

  !!!**(GITIGNORE NESTED FOLDERS EXCEPT SITUATION)**

CHAPTER 6

- BRANCH

  git init
  touch d1.py d2.py d3.py
  git add .
  git commit -m “v1”
  git branch
  git branch yandal1
  git branch
  git checkout yandal1
  git branch
  touch yandal1.py
  git add .
  git commit -m “yandal1 tarafından oluşturuldu”
  git checkout master
  git diff master yandal1
  git merge yandal1

CHAPTER 7

- README - https://help.github.com/en/github/writing-on-github/basic-writing-and-formatting-syntax
  github.com web arayüz gösterimi

  ```
  ------------------------------------------------------------
  .       # Proje Başlığı                                    .
  .                                                          .
  .        ## Proje Amacı                                    .
  .                                                          .
  .       **kalın metin** <br/>                              .
  .       *italik metin*                                     .
  .       `kod`                                              .
  .       [Google Link](www.google.com)                      .
  ------------------------------------------------------------
  ```

CHAPTER 8

- WATCH, STAR, FORK
- GISTS

CHAPTER 9

- GIT STASH
  git clone ...
  nano d1.py
  git status
  git pull (not allowed)
  git stash
  git pull
  git stash list
  git stash pop 0
  git status
  git add .
  git commit

https://youtu.be/jJFd__VX-GU

CHAPTER 10

- GITHUB BEST PRACTICES
  https://resources.github.com/videos/github-best-practices/