```
git init open-tutorial-2023
cd open-tutorial-2023
conda activate base
jupyter lab

n = 10
d = 2

def divide(n, d):
    result = n / d
    return result

git status
git add divide.py
git status

git commit -m “Add simple division script”

git status    # no longer in staging area
git log

def divide(n, d):
    return n / d

git add divide.py
git commit -m "Remove 'result' to slim down division"

git status
git log

ls    # how is git tracking all this?
ls -al    # to see .git

git log
git checkout SHA    # read git’s message here
less divide
git diff 5bee:divide.py 3d35:divide.py

git branch
git checkout master
git branch

vim .gitignore    # .ipynb_checkpoints
```

```
git add .gitignore
git commit -m "Add .gitignore for .ipynb_checkpoints"

git checkout -b square-dev
git branch

def square(x):    # jupyter square.py
    return x * x

git add square.py
git commit -m "Add new squaring script"
git diff master square-dev
git checkout master
git status
ls

git checkout master
git merge square-dev

git checkout -b square-bad
def square(x):    # jupyter square.py
    return x * 2
git add square.py
git commit -m "Doing great evil to squaring script"
git diff master square-bad

git checkout master
def square(x):    # vim square.py
    return x ** 2
git merge square-bad    # merge conflict

git branch -D square-bad
git branch -D square-dev

git restore square.py
git status

# create empty repository with no README, LICENSE on GitHub
git remote add origin https://github.com/snastase/open-tutorial-2021.git
git push -u origin master

# try git cloning the repository!
git clone https://github.com/snastase/open-tutorial-2021.git
```

```
# create README on GitHub
git pull
ls
```

```
# on GitHub, click “Add file”, type LICENSE, and then click Choose license
git pull
ls
```

```
$ python3
>>> from divide import divide
>>> divide(10, 2)
>>> divide(10, 3)
>>> exit()

$ python2
>>> from divide import divide
>>> divide(10, 2)
>>> divide(10, 3)
>>> exit()

$ which python
```

```
pylint divide.py

# create test_divide.py
from divide import divide

def test_divide():
    assert divide(10, 2) == 5
    assert divide(10, 4) == 2.5

test_divide()

$ py.test

# change divide.py to // and re-run py.test
```

```
# fork course directory on GitHub
# clone your fork
history -w bash_history.txt
cd
git add bash_history.txt
git commit -m “Add my personal bash history for class”
git push
```