Advanced Python Programming Course

Lecture 2.

Git and GitHub

Assoc. Prof. Kovalenko S.M.
Department of software engineering and intelligent management technologies,
NTU “KhPI”

Google Colaboratory

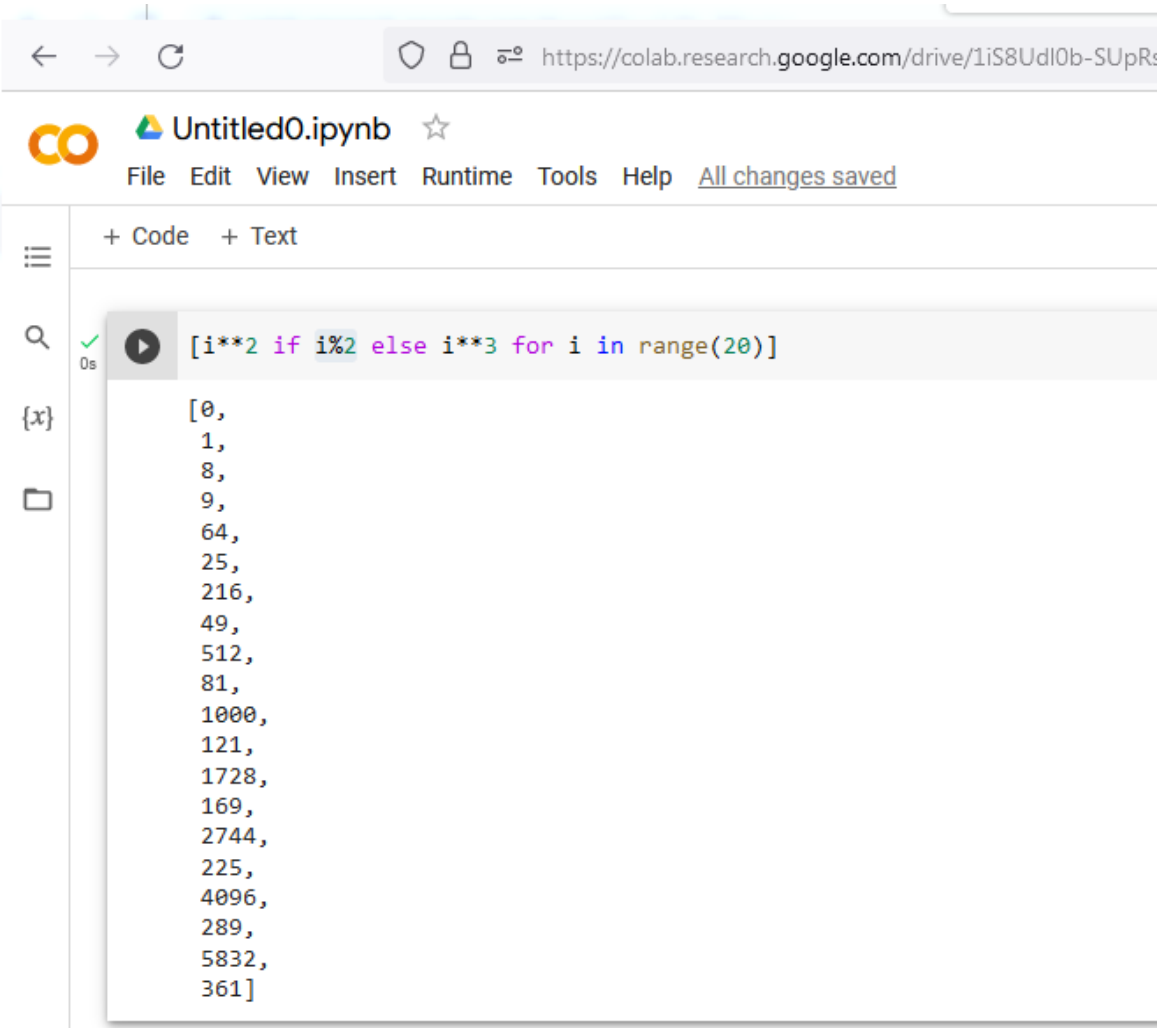
https://colab.research.google.com/?hl=en

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

But you need a Google account

Google Colaboratory



How to share notebooks

- 1. Push the notebook on GitHub
- 2a. Enter to browser address bar the path Ввести в адресний рядок https://colab.research.google.com/path_to_your_file_on_GitHub
- 2b. Go to **nbviewer (**https://nbviewer.jupyter.org/) and enter GitHub repository in the appropriate field. In the open repository window, select the desired file.

https://github.com/svniko/Advanced_Python/blob/main/Lecture_1.ipynb

What is Git?

- Git is a popular version control system. It was created by Linus Torvalds in 2005

- It is used for:
  - Tracking code changes
  - Tracking who made changes
  - Coding collaboration

What does Git do?

- Manage projects with **Repositories**
- **Clone** a project to work on a local copy
- Control and track changes with **Staging** and **Committing**
- **Branch** and **Merge** to allow for work on different parts and versions of a project
- **Pull** the latest version of the project to a local copy
- **Push** local updates to the main project

What is GitHub?

- git is not the same as GitHub.
- GitHub makes tools that use git.
- GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.

Also you can try:

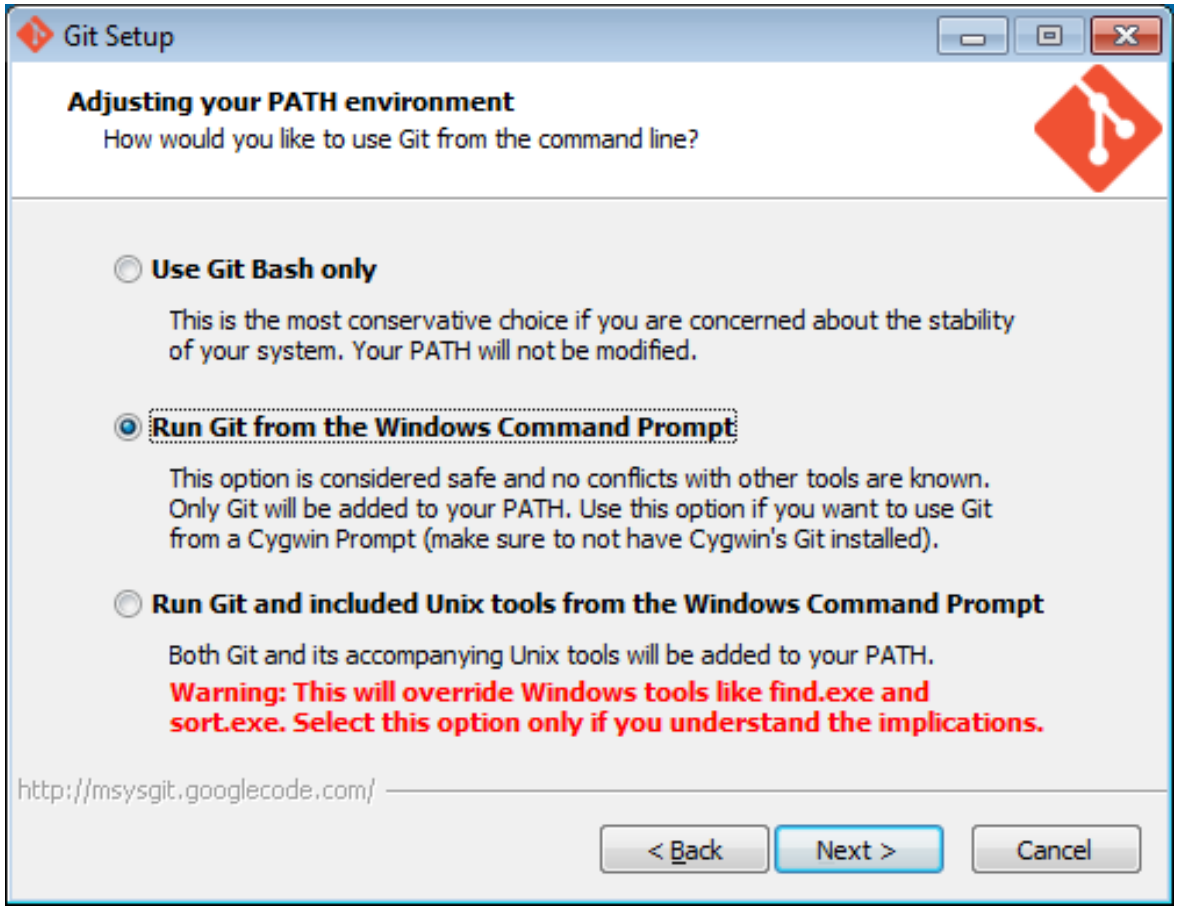
- GitLab
- Bitbucket

Git. Client

- http://git-scm.com/downloads

Git. Client

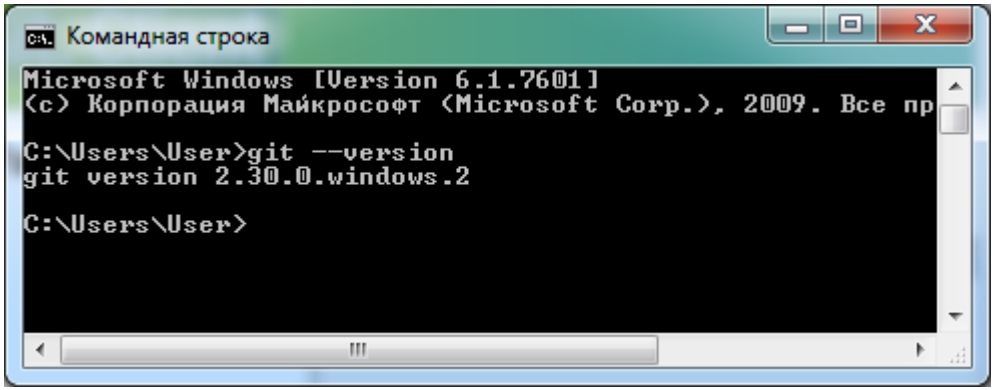
- Path Dialog

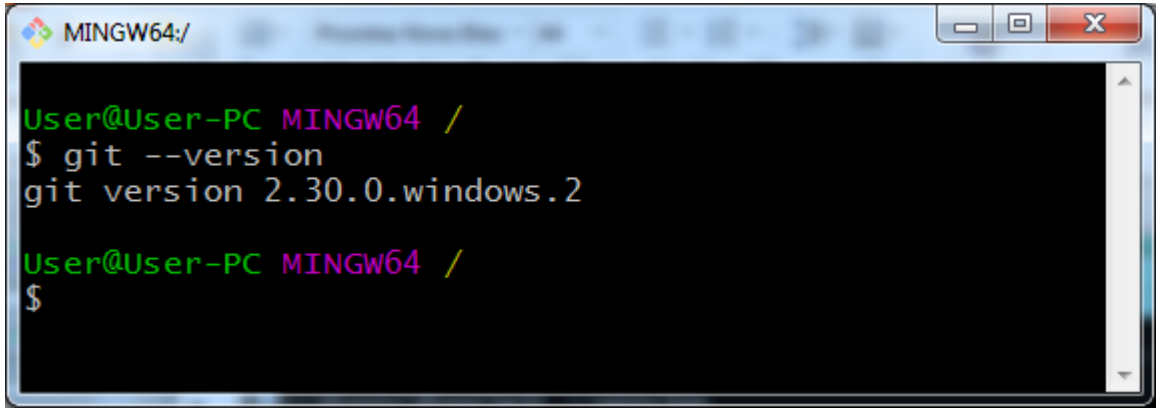


Git. Client

- Check if git is installed

```
git --version
```





Git. Create a local repo

- Go to the directory of your project and initialize a local Git repository by

```
git init
```

- Register your name and e-mail in git

```
git config --global user.name <desired name
to be seen as author of code>
git config --global user.name spiderman
git config --global user.email <desired e-mail
to be seen as authors of code>
git config --global user.email hohoho@gmail.com
git config --list
```

.gitignore

In the directory of your project create a file .gitignore (**the dot before the name is compulsory**)

A `.gitignore` file specifies intentionally untracked files that Git should ignore

`.gitignore` generator

`https://www.toptal.com/developers/gitignore`

Some Linux command

- `cd` - change directory
- `ls` - display a list of files and sub-directories
- `ls –a` - list all files including hidden file starting with '.‘
- `touch <file name>` - create a file without any content
- `cat <fileName>` - display the content of a file
- `vim <file name>` - open file for editing in Vim. If file does not exist, it will be created.
  - **Some Vim commands**:
    - `<I>(I key)` - insert mode
    - **<**`Esc`**>** - command mode
    - `:wq<Enter>` - save file and quit

Git. Status of Files

- See the state of our repo

```
git status
```

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be commi
        .gitignore
        Lecture_1.ipynb
        Out.txt

nothing added to commit but untracked files present (use "g
d" to track)
```

Add files to the git staging area

- Add a file

```
git add <file_name>
```

- Add all

```
git add .
```

- Check status

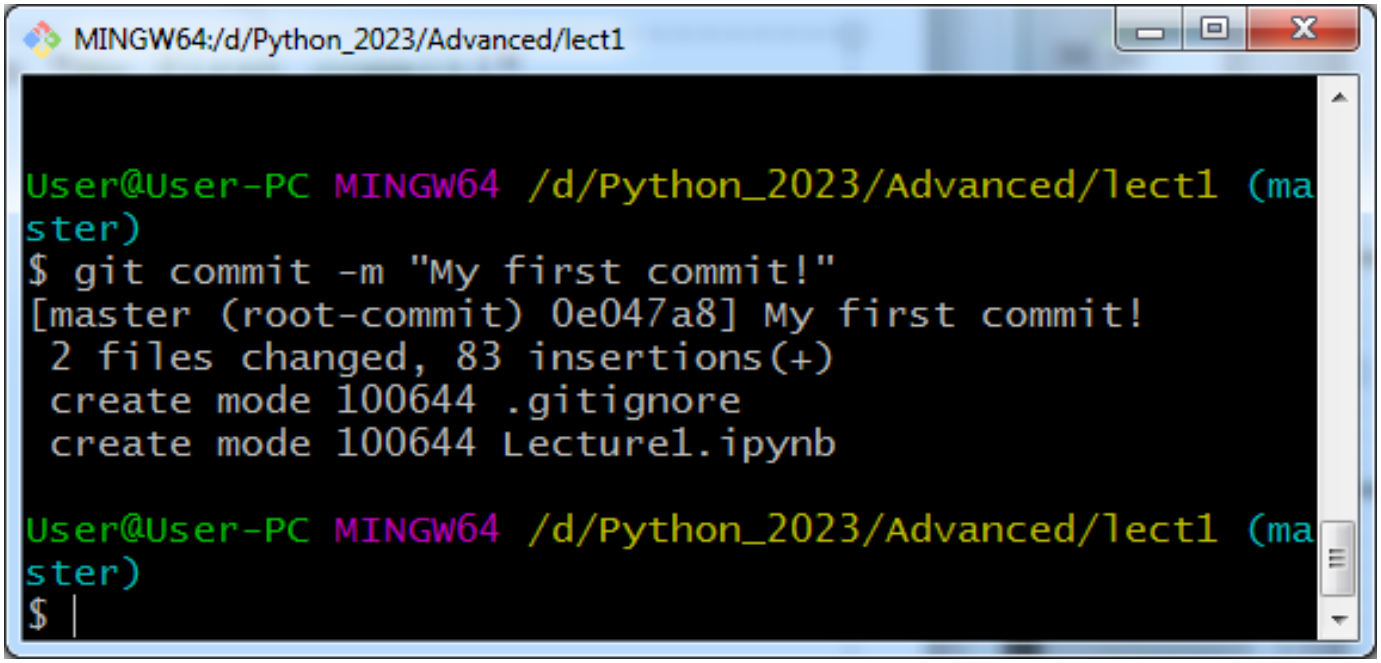
```
git status
```

- Check branch

```
git branch
```

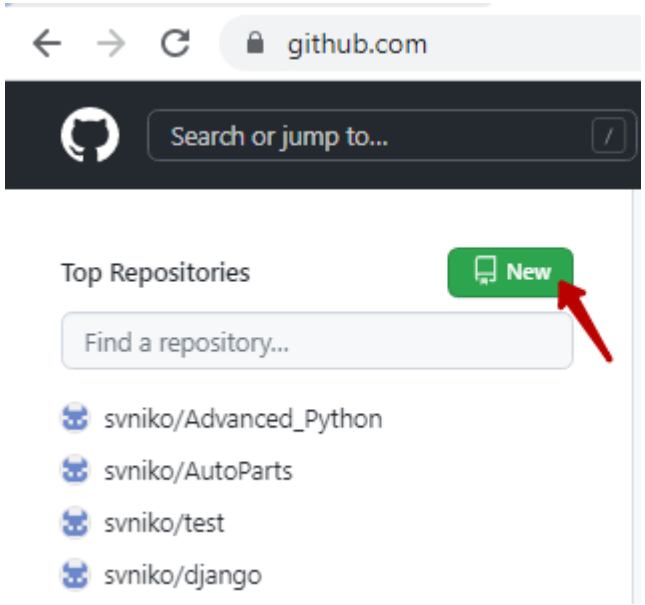
Git. Commit changes

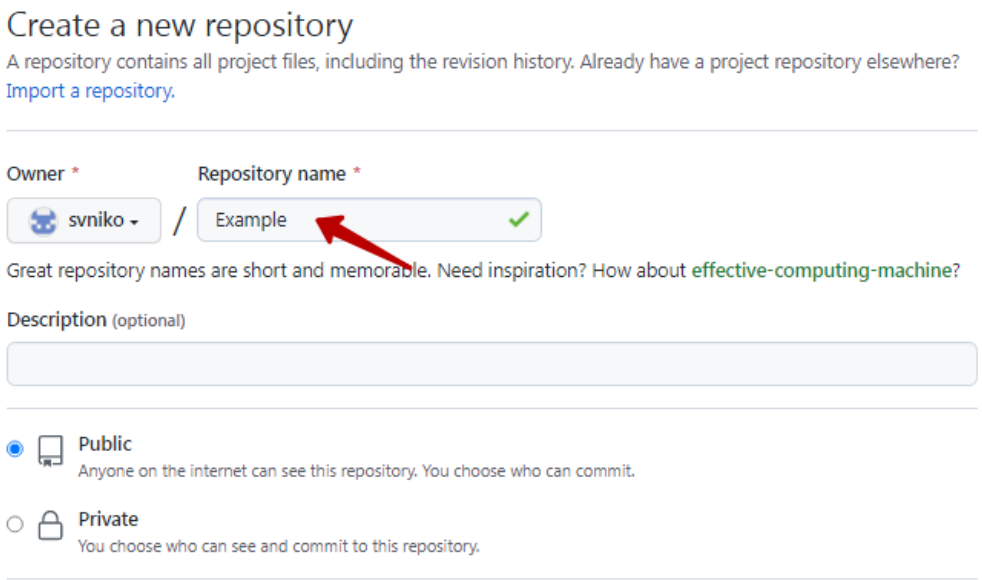
```
git commit -m "My first commit!"
```

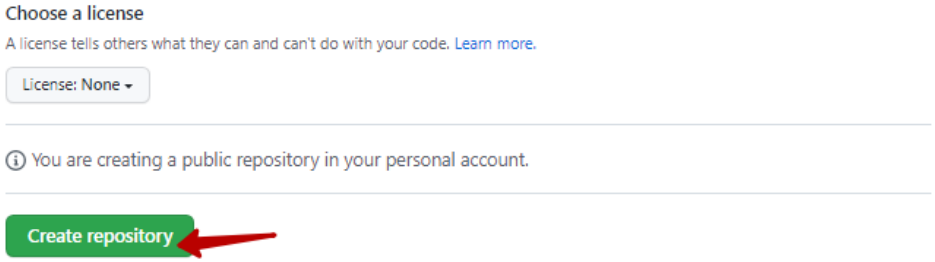


GitHub. Create a new repo

- Go to https://github.com/







GitHub. Create a new repo

Quick setup — if you’ve done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/svniko/Example.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE

...or create a new repository on the command line

```
echo "# Example" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/svniko/Example.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/svniko/Example.git
git branch -M main
git push -u origin main
```

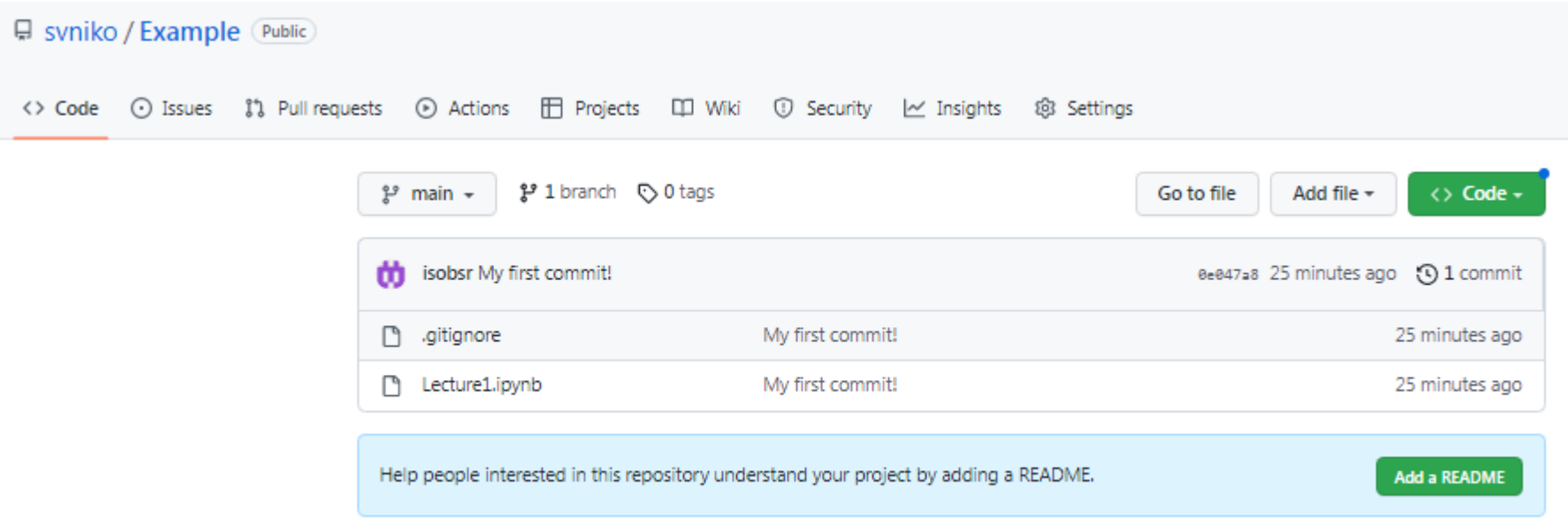
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

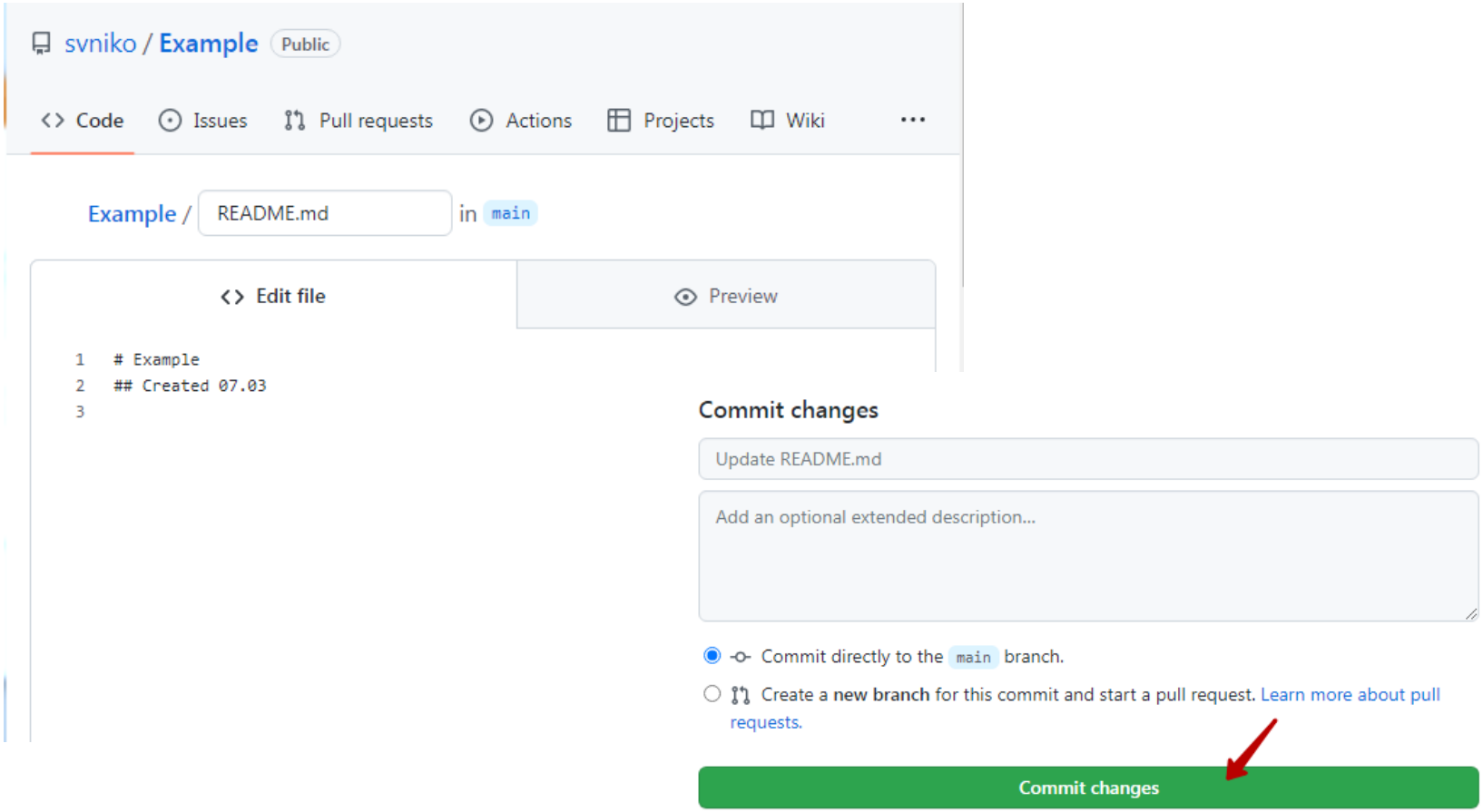
Push local repo on GitHub

- `git remote add origin <path to your GitHub repo>`
- `git branch -M main`
- `git push -u origin main`

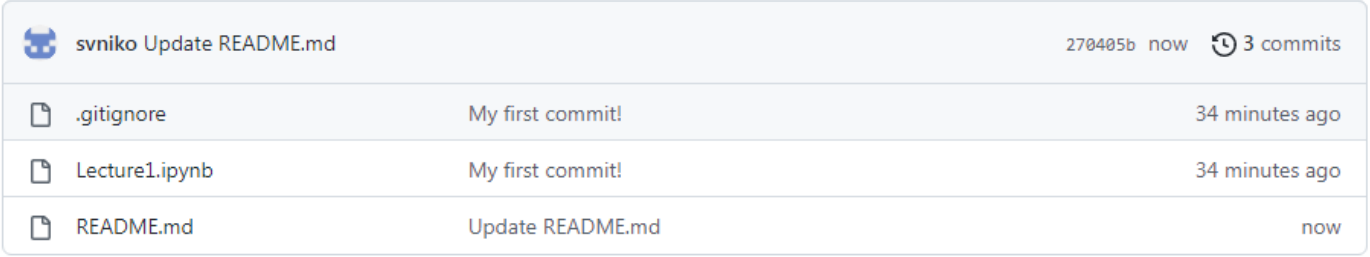


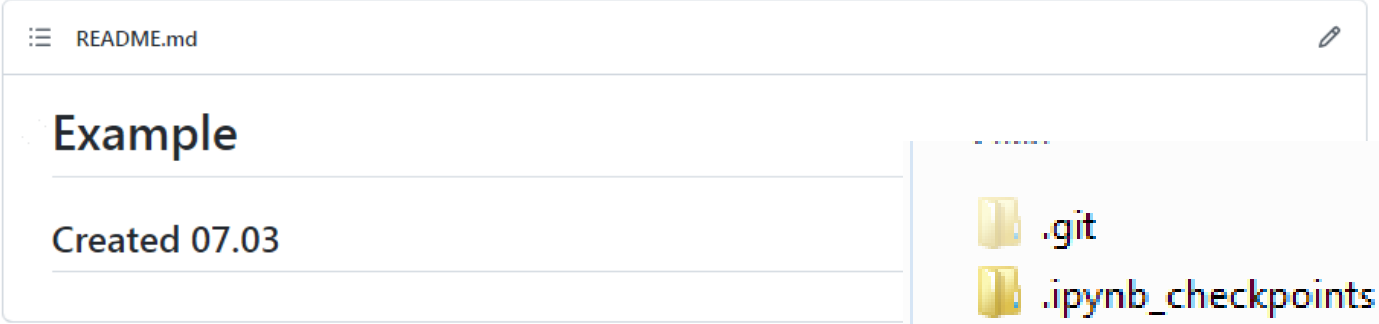
Pull changes

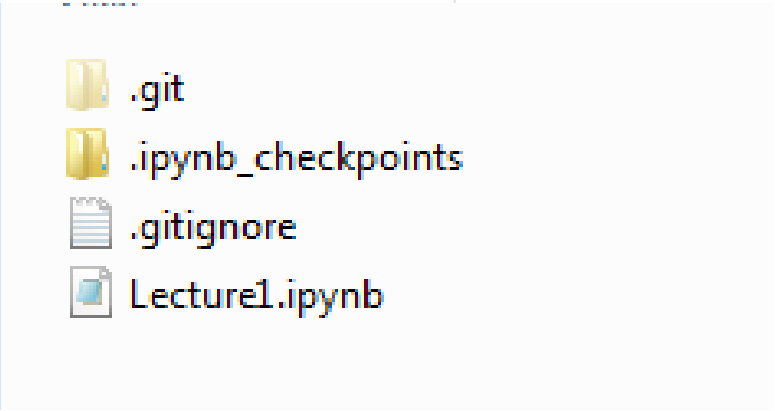
- Let’s create a README.md



GitHub repo vs local repo







Get data

Gets (fetch) data and automatically to merge with the code (master)

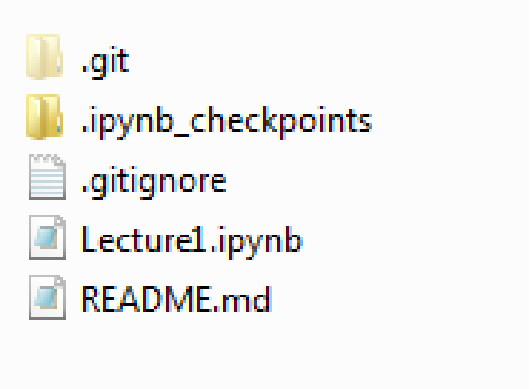
```
git pull
```

MINGW64:/d/Python_2023/Advanced/lect1

```
$ ^C

User@User-PC MINGW64 /d/Python_2023/Advanced/lect1 (ma
in)
$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-re
used 0
Unpacking objects: 100% (6/6), 1.31 KiB | 4.00 KiB/s,
done.
From https://github.com/svniko/Example
   0e047a8..270405b  main       -> origin/main
Updating 0e047a8..270405b
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 README.md

User@User-PC MINGW64 /d/Python_2023/Advanced/lect1 (ma
in)
$
```



After any file modification

```
1. git status
2. git add .
3. git commit -m "<message>"
4. git push
```