FS12

Week 01
Python Basics I

Mikhail Masyagin
Daniil Devyatkin

Course Links

Telegram

GitHub

Course Plan

- 17 weeks;
- 4 modules:
  - basic Python and object-oriented programming;
  - sorting/string algorithms and basic data structures;
  - trees and hash-maps, SQL database essentials;
  - basics of networking and parallel/async coding;
- 3 tests, lots of coding hometasks, pear review;
- 2 lecturers :)

Course Rules & Recommendations

- you have to **pass 3 tests;**
- you have to do all **coding tasks**;
- teamwork is appreciated;
- cheating is prohibited;
- don't be shy to ask questions;
- leave feedback to make course better;
- lecturer can be wrong.

- Linux is a recommended OS;
- please, read materials before a class.

About Lecturers

GitHub

Mikhail Masyagin

- Senior Software Engineer at Anecdote;
- Software Engineer at STC Atlas;
- Assistant Lecturer at BMSTU.

About Lecturers

GitHub

Daniil Devyatkin

- Data Scientist at VK;
- Assistant Lecturer at BMSTU.

What is ~~Love~~ Python?

- Python is a high-level, general-purpose programming language with Object-Oriented programming support and Garbage Collector.
- It was originally designed by Guido van Rossum in the late 1980s: Python 1.0. Today we use Python 3.*. Mostly Python 3.7-3.11.

BDFL

Why Python?

- Python is often described as a "batteries included" language due to its comprehensive standard library.
- It is used in most popular and fast growing programming spheres:
  - web-development;
  - machine learning;
  - computer experiments & hypothesis testing;
  - everywhere!
- Course authors don’t know C++ (nobody knows it, including Bjarne Stroustrup).

Python Installation I

Open a terminal (powershell in Windows) and type `` `python` `` (maybe `` `python3` ``). If command doesn't fail, you already have Python on your computer, otherwise go via QR-code and install it.

Windows

MacOS

There are other ways to install Python on your machine...

Python Installation II

- As you could see, there is no QR for Linux. This is due to almost all Linux distributions have special programs – package managers, which can be used to install Python from terminal. For example Ubuntu has APT package manager, so Python can be installed via following command:

```
> sudo apt install python3
```

- MacOS has its own package manager named Homebrew. You can install Python via it to.

Python Pip & Venv I

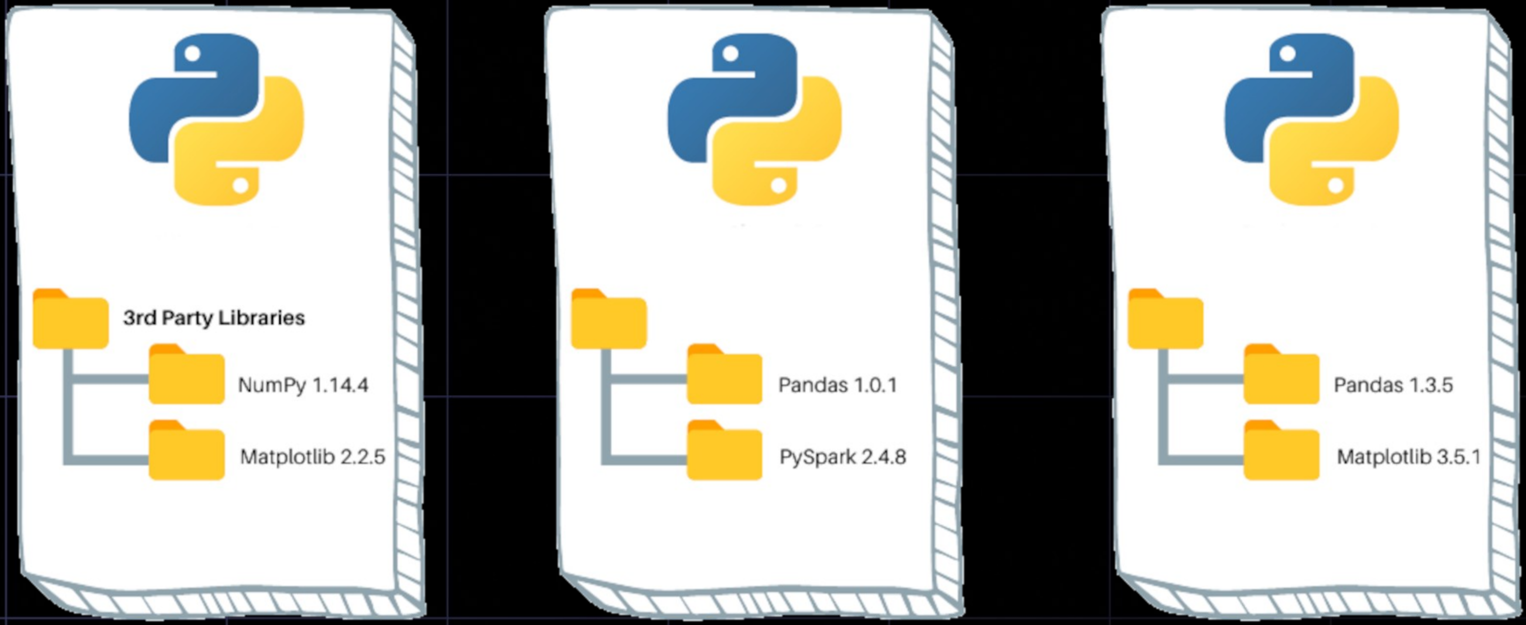
- As Linux distributions have package managers, Python has its own smaller package manager named **pip**. If you have installed Python via QR, you should already have it. Otherwise you need to install it too.
- You may know all this stories about BSOD on Windows and following reinstallation of operating system. Python will not break your computer, but sometimes you can install some conflicting packages via pip or suddenly delete some python files. We have a solution for you – **venv**!

Python Pip & Venv II

- Also if you have many Python projects (some for university, some for work etc) you don't want to mess their dependencies in single Python installation, and it is really hard to install two similar Pythons on one desktop. The solution is the same – **venv**!
- Venv is a Python "virtual environment", independent set of Python packages. It is created on top of an existing Python installation, known as the virtual environment's "base" Python.

Python Pip & Venv III

- You can’t have Pandas 1.0.1 & Pandas 1.3.5 in one Python, but you can have them in two different virtual environments.



Python Pip & Venv IV

- Let's create and run it:

```
> python3 -m venv .venv
> source .venv/bin/activate
> pip list
```

- Now you are using Python and its Pip in virtual environment, so you will not break your entire Python installation.

Hello, World! I

- Now we have installed Python and its active virtual environment, so we are ready to run our first program – "Hello, World!". To do it, just type in terminal `python` and then write the following line of code:

```
>>> "Hello, World!"
```

- You will see following text on the next line after pressing enter:

```
'Hello, World!'
```

Hello, World! II

- Now let's create a file named `` `hello_world.py` `` with following line in it:
```
"Hello, World!"
```
- Then let's run our program:
```
python3 hello_world.py
```
- You will not see any output in the terminal. It is because firstly we have opened **Python's REPL**, but secondly we have created a **Python program** and ran it by **Python's interpreter**.

Hello, World! III

- REPL is:
  - <u>R</u>ead the user input.
  - <u>E</u>valuate your code.
  - <u>P</u>rint any results.
  - <u>L</u>oop back to step 1.
- REPL is a really good tool for some simple debugging or showing demos. For example, I'll use it during first course module. But forget about it if you want to write a complex Python program.
- REPL will output to terminal everything you type, but program will not.

REPL

Hello, World! IV

- To overcome problem with `` `hello_world.py` `` we will wrap our string `` `"Hello, World!"` `` into function call `` `print` ``, so final code will be:

```
print("Hello, World!")
```

- Let’s run our program:

```
python3 hello_world.py
```

- We will see following output:

```
Hello, World!
```

- What is function and what is `` `print` `` will be discussed in the future.

What’s next?

- Today you have learned two real cool Python’s features: REPL and interpreter, but how can you share your knowledge with other people in a best way?
- The answer is **GitHub** repository & **README.md**.

GitHub

README

GitHub I

- As GitHub uses **Git** version control system (VCS) internally, it is essential to install it on your computer.
- On Linux Distributions you can do it via package manager. For example on Ubuntu Linux you can run following command:

```
> sudo apt install git
```

- For MacOS you may use Homebrew.

Windows

MacOS

GitHub II

- GitHub is a platform and cloud-base service for software development and version control using Git, allowing developers to store and manage their code.
- There are two main benefits which you get if you decide to use GitHub during this course:
  - You will not lose your code if you forget your laptop or it is broken.
  - GitHub is like an Instagram, but for programmers (helpful for getting a job).

GitHub III

- Go via QR code and please register your GitHub account.
- It is essential to add additional level of security and use One-Time Password Algorithm. For it you can use Google Authenticator.
- Let's create our first repository `` `hello-world-py` `` and put `` `hello_world.py` `` into it.
- Process of it is shown in the video.

Google Auth

GitHub IV

- Basic command which you've to use:

```
> git clone git@.../hello-world-py.git
> cd hello-world-py
> git checkout -b develop
> nano hello_world.py
> git add hello_world.py
> git commit
> git push origin develop
> git checkout main
> git merge develop
> git push origin main
```

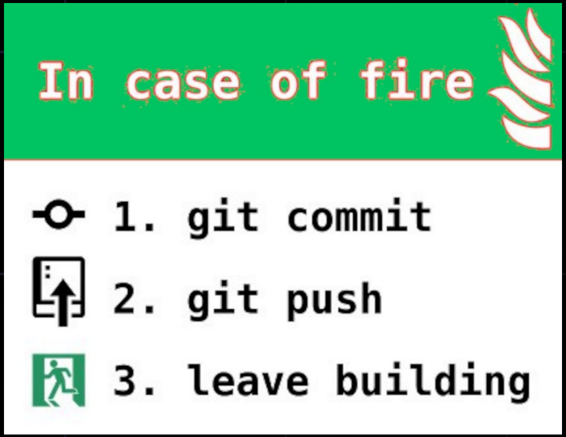
GitHub V

- Let’s discuss used commands:
- `git clone ...` download remote repository to your local machine;
- `git checkout -b ...` creates new branch. It is recommended to use for development process separate special branch (branches) because main branch should always be clear;
- `git add ...` adds selected files to your future commit (files addition to current branch);

GitHub VI

- \``git commit`\` adds commit-message to your code push to git. You have to provide this message manually. It helps readers of your code understand history of code changes and your logic.
- It is worth to use following commit form:

```
repo name: short commit description
<blank line>
* milestone 1
* ...
* milestone N
```


In case of fire

1. git commit
2. git push
3. leave building

GitHub VII

- `git push origin ...` pushes your code on local machine to remote git repository;
- `git merge ...` adds all data from another branch to current branch. Be careful, there can be conflicts which you have to resolve manually;

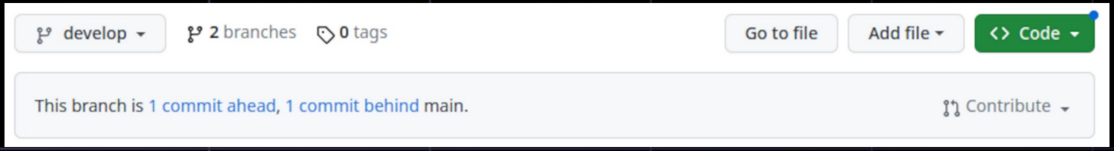
- Now you know how to basically use git.
- But also let's discuss how to protect your main branch and use **Pull Requests**.

GitHub VIII

- <u>**Pull Request**</u> allows you to propose and collaborate on changes to a repository. These changes are proposed in a separate branch. For example we can make a Pull Request from our created develop branch to master branch. It simplifies process of code review. Code review is necessary to minimize number of mistakes & bugs, which can be introduced by adding new code and features to project.
- Pull Requests are often used together with <u>**branch protection.**</u>

GitHub IX

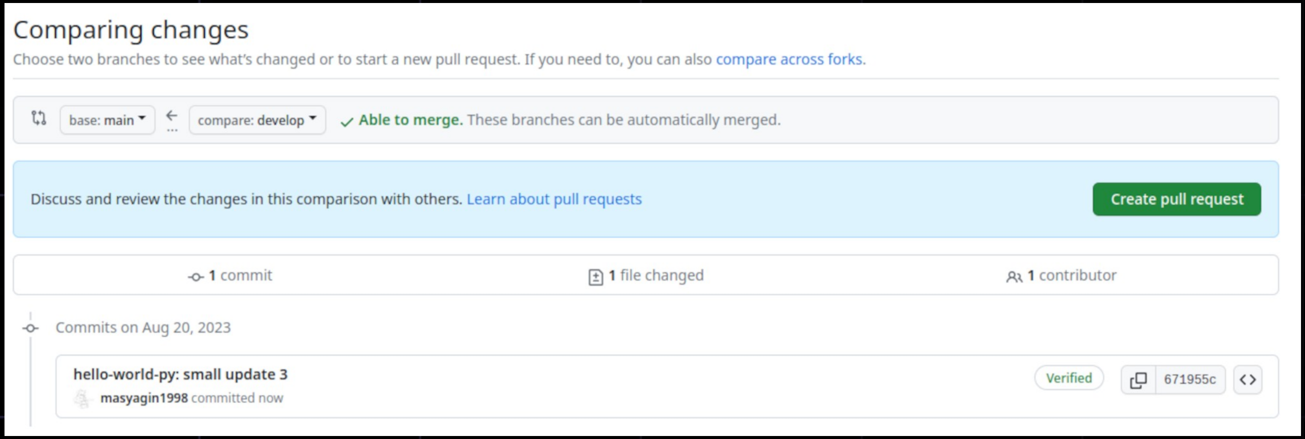
- Process of making and approving Pull Request is shown on video.



develop · 2 branches · 0 tags · Go to file · Add file · Code

This branch is 1 commit ahead, 1 commit behind main.

Contribute



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: main ← compare: develop ✓ **Able to merge.** These branches can be automatically merged.

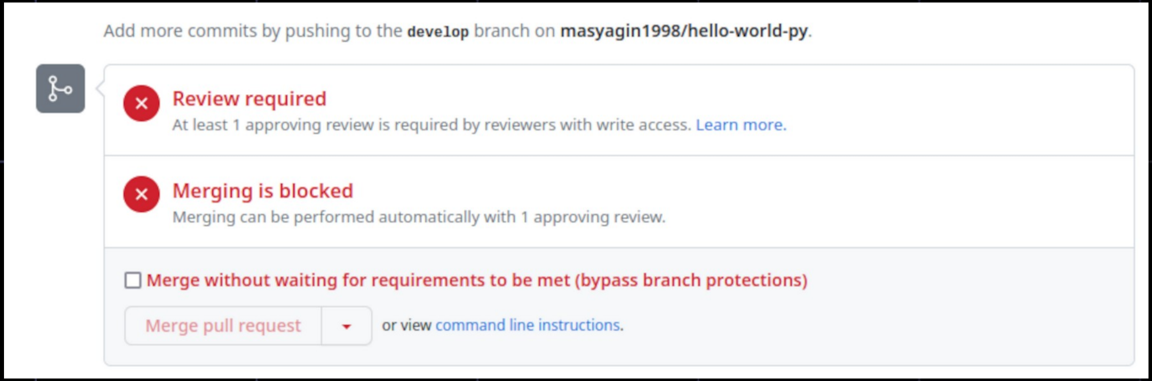
Discuss and review the changes in this comparison with others. Learn about pull requests

Create pull request

1 commit · 1 file changed · 1 contributor

Commits on Aug 20, 2023

hello-world-py: small update 3
masyagin1998 committed now · Verified · 671955c



Add more commits by pushing to the **develop** branch on **masyagin1998/hello-world-py**.

Review required
At least 1 approving review is required by reviewers with write access. Learn more.

Merging is blocked
Merging can be performed automatically with 1 approving review.

☐ **Merge without waiting for requirements to be met (bypass branch protections)**

Merge pull request · or view command line instructions.

README.md

- Commit messages are helpful to understand what was changed and why it was changed.
- But if you want to make people understand how does your project operate right now, you can write short documentation in a **md** format.
- You can edit md files directly on GitHub or on a special platform – dillinger.io.
- Process of making README.md is show on video.

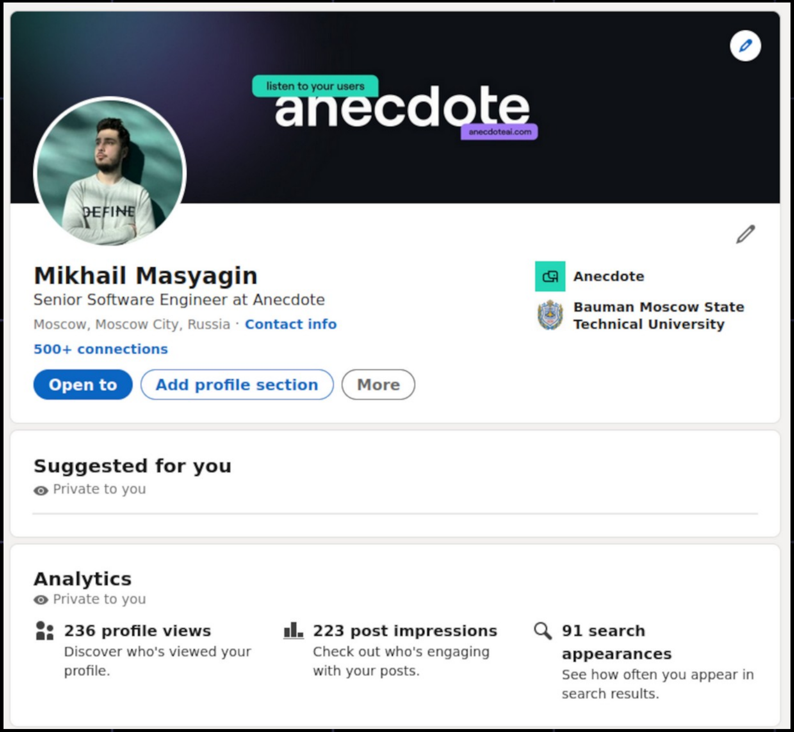
dillinger.io

Linkedin I

- The main goal of this course is to show you a real programming experience and help you to find the IT job as fast as possible.
- If GitHub is a programmers Instagram, Linkedin is a programmers Facebook. It is a social network where you can share you career and study growing, scientific articles and of course… find the job.
- It is highly recommended to create a Linkedin profile and keep it actual.

Linkedin II

- You may have problems connecting to Linkedin, because some providers block it. You know how to fix it.
- On video you will see example of Linkedin profile.



listen to your users
anecdote
anecdoteai.com

Mikhail Masyagin
Senior Software Engineer at Anecdote
Moscow, Moscow City, Russia · Contact info
500+ connections

Open to | Add profile section | More

Anecdote
Bauman Moscow State Technical University

Suggested for you
Private to you

Analytics
Private to you

236 profile views
Discover who's viewed your profile.

223 post impressions
Check out who's engaging with your posts.

91 search appearances
See how often you appear in search results.

Linkedin

That's all Folks!