Introduction to Python Applications of Python

Sugarkhuu Radnaa

Py4Econ in Ulaanbaatar *py4econ@gmail.com*

About the course

1. Basic programming know-hows
2. Elementary to Intermediate Python
   - Introductory section topics:
     - Python specific: Data basics, Functions/Class, Some useful knowledge
     - General programming: Code editor (VScode), Git/Github
   - Applications topics: Visualizaiton, Automation, Webscraping, Files, ML/DL
3. Homework after each session

About the course

- Following should help make the best out of the course:
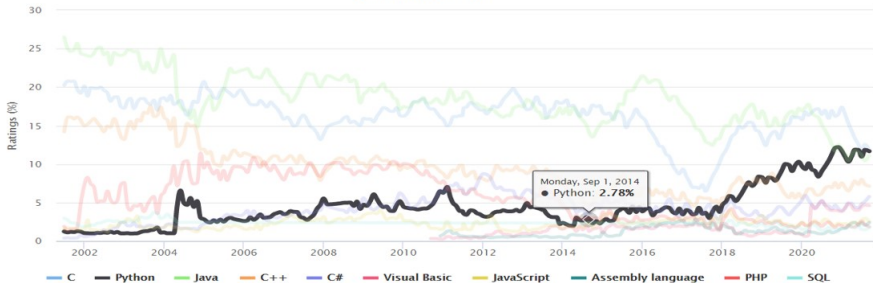 - Doing homeworks in time
  - Asking questions

Week 1: Learning objectives

1. Background information
2. Computer basics (Folder structures, Path, Shell)
3. Python /Anaconda/
4. VScode, Jupyter notebook and other IDEs
5. Git & Github

Why Python?

+

Beginner friendly

Popular applications of Python

1. Data science and data visualization
2. Machine learning & AI
3. Scientific computing (incl. Financial modelling)
4. Web & Game development
5. Desktop applications & Software & GUI
6. Automation

Python was first released in 1991
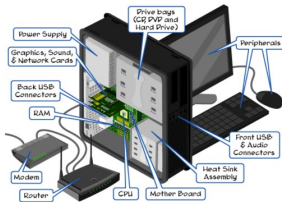
Timeline Of Programming Languages :

Source: https://javaconceptoftheday.com/history-of-programming-languages/
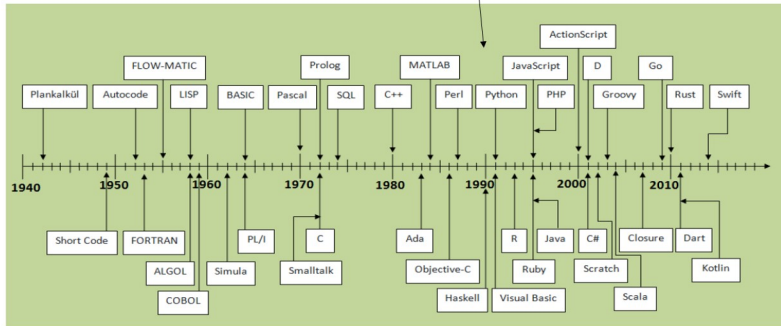
Communities and learning platforms

1. Python - Official
2. Stack overflow - Forum
3. Medium - Blog
4. Towardsdatascience - Blog
5. Tutorialspoint
6. Geek for geeks
7. W3schools

8. Real Python
9. Programiz
10. Kaggle - Competition and Learning resource
11. DrivenData
12. TopCoder
13. DataHack
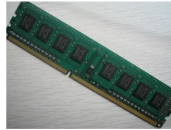
Computer basics

Processor

RAM

Hard disk

Data units

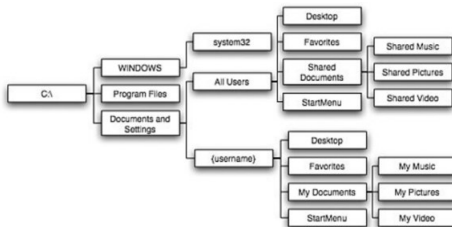
CPU (chip)

Operating systems

Mac, Linux,
Windows

Folder structure

Folder structure

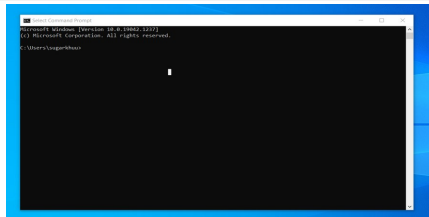Figure 4.7. Windows XP file structure.

Copy path:
- Ctrl + L = folder path
- Shift + Right mouse > copy as path (a) = file path

When naming folder & file:
- Avoid spaces and uncommon characters
- Use either camel or snake cases

C:\Documents and Settings\sugarkhuu\My Documents\My Video

Command line interpreters/Shells

You are able to control your computer through commanding the OS from terminals (Win+CMD, Ctrl+T). More powerful and flexible than usual GUI way of doing things

- Windows: Command prompt, Powershell. More recently, Windows Terminal
- Linux: Bash
- Mac: Terminal (zsh)

Common CMD/Bash commands

- cd [cd] – change directory. /? - help
- dir [ls] – directory content
- copy [cp] – copy file
- ren [mv] – rename file
- del [rm] – delete file
- mkdir [mkdir] – create new folder
- exit [exit] - close terminal
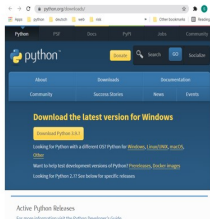- cls [ctrl+L] – clear terminal

Path (environment variable)

The PATH variable makes it easy to run commonly used programs located in their own folders.
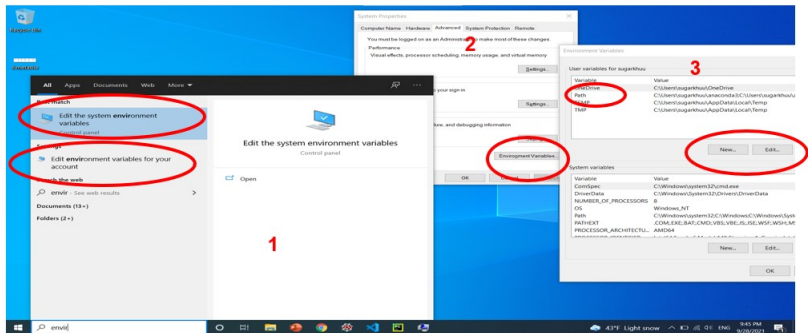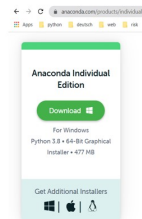
Getting Python

Standalone Python

https://www.python.org/downloads/

Pre-installed all
packages useful
for data science

→

Anaconda (incl. python)

https://www.anaconda.com/products/individual
https://www.datacamp.com/community/tutorials
/installing-anaconda-windows

What is a programming language?

A programming language is a formal language comprising a set of strings that produce various kinds of machine code output. Wikipedia

```
1
2
3    print("Hello Py4Econ!")
4
5
```
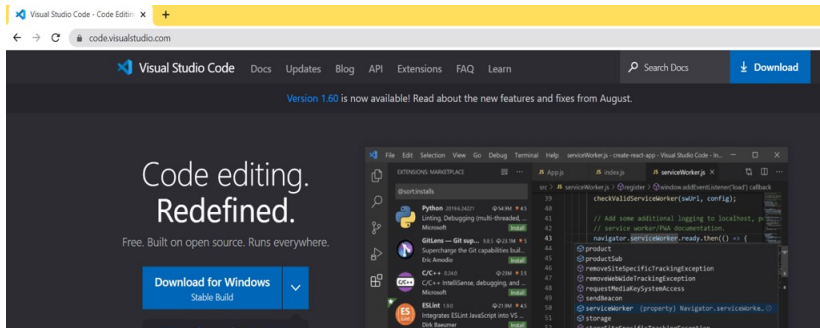
Python basic concepts for today:

- Basic syntax
- Basic operators
- Packages (+ pip)

Possible to use Python in many environments

① Terminal

② IDEs – Spyder or Pycharm (IntelliJ), VScode

③ Notebook - (Jupyter notebook)

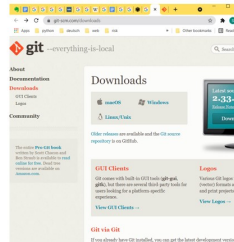# Using VS code (code editor) for Python

Git – Version control system (VCS)

Version control system (VCS):

Version control systems are a category of software
tools that helps in recording changes made to files by
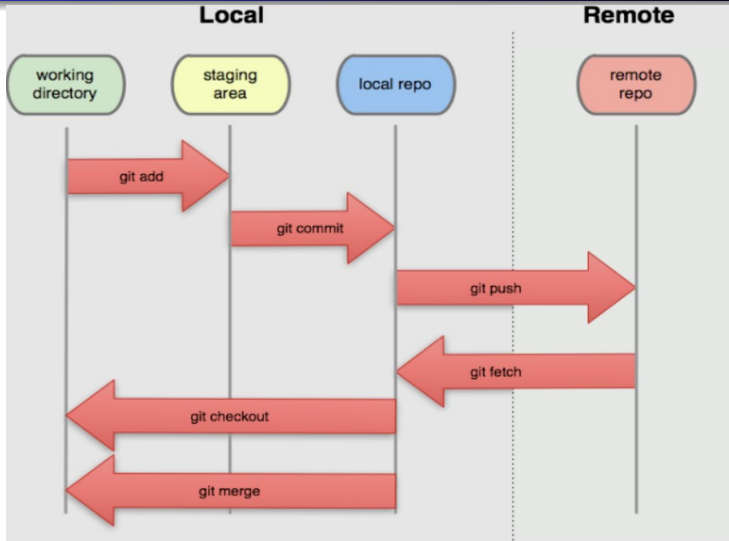keeping a track of modifications done to the code.

https://www.geeksforgeeks.org/version-control-systems/

Popular VCSs: Git, Subversion, Helix core, Microsoft TFS

https://git-scm.com/downloads

Git – Version control system (VCS)

Github – Share everything you want

https://code.visualstudio.com/

Git bash (terminal)

A unix based commands on Windows (Emulator)

Guide: Must-have basics for a good programmer

- Data Structure and Algorithm
- A Version Control Tool (Git)
- One Text Editors (VScode)
- IDEs (Spyder or Pycharm)
- Database and SQL
- UNIX (Linux)
- An OOP Programming language (C++, Java or Python)
- One Scripting language (automation)
- Networking basics
- *Cloud Platform (AWS, GCP, or Azure)*
- *Containers (Docker and Kubernetes)*

Homework

1. Task 1
2. Task 2

- Submit your result as a Github repository
- Deadline: 1 week

Task 1

1. Create a new repository in your Github
2. Clone the previous repository to your local machine
3. Create a Jupyter notebook in the local repo (in your folder)
4. In the notebook, write a code which asks 5 separate questions and receives the answers from the user (we had an example with only one question in the lecture)
5. Commit and push your changes question by question one at a time
6. Make sure your code follows the good practice

Task 2

1. Нэг компьютерт хоёр үйлдлийн систем суулгаж болох уу?

2. Path-д программаа оруулаагүй бол яах вэ?

3. Фолдерийн нэр нь дундаа зайтай байвал фолдерийг танихад ямар асуудал үүсэх вэ?

4. Git, Github хоёрын ялгаа юу вэ?

5. Jupyter notebook IDE мөн үү?

6. Commit, push хоёрын ялгаа юу вэ?

7. Push хийхээс өмнө олон дахин commit хийж болох уу?

8. Commit хийхэд github repo-д access хэрэгтэй юу?

Thank you!