Programing in Python

Lecture 1 - Introduction

Instructor: Zhandos Yessenbayev

Outline

- Syllabus
- What is Python and why learn it?
- Programming Environment

General Information

Course Title	**Data Structures and Algorithms**
Instructors	Zhandos Yessenbayev, PhD
Level	Undergraduate
Year	2021
Term	Autumn
Credits	3
Status	Mandatory
Hours per Week	2 lectures + 6 seminars
Total Hours	55 hours
Form of Education	Daytime
Language of Instruction	Kazakh, Russian, English
Schedule	Tue-Fri, 10:30-12:20

Reading Materials

- Kenneth A. Lambert, **The Fundamentals of Python: First Programs**, 2011, Cengage Learning, ISBN: 978-1111822705.

- Allen Downey, 2015, **Think Python**, Green Tea Press

- **The Python Tutorial** (https://docs.python.org/3/tutorial/): This is the official tutorial from the Python website. No more authoritative source is available.

- **Problem Solving with Algorithms and Data Structures** (http://interactivepython.org/runestone/static/pythonds/index.html (Links to an external site.)Links to an external site.)

- **Fluent Python** (http://shop.oreilly.com/product/0636920032519.do): All python3, and focused on getting the advanced details right. Good place to go once you've got the basics down.

Grading

- There will be no explicit Midterms and Finals!
- Grading will be based on the course work done:
  - Seminar tasks - 60%
  - Final project - 40%

Progress Board

https://docs.google.com/spreadsheets/d/1E_CtL-nnB_9MKXOyMEu-yphAQhpIf10qdThKE55Xiis/edit?usp=sharing

Name	ST1*	ST2	ST3	ST4	ST5	ST6	ST7	Project (40%)	Total (100%)
Абекен Айбек									
Абжаппар Елшібай									
Айтилеуов Ғазиз									
Амержан Әлибек									
Демесинов Әділет									
Жаңабайұлы Еркебұлан									
Изтілеу Олжас									
Кажымұрат Ғалымжан									
...									

* 60% of the grade will be distributed over the tasks

Lectures

- **Lecture 1. Introduction**
- **Lecture 2. Basic Operations and Data Types**
- **Lecture 3. Flow Control and Loops**
- **Lecture 4. Functions, Modules, Packages**
- **Lecture 5. String, List, Tuples, Dictionary**
- **Lecture 6. File Operation**
- **Lecture 7. Debugging and Exception Handling**

What is Python?

- **Python** is an interpreted, object-oriented, high-level programming language with dynamic semantics.

- **Guido van Rossum** (1956) - created Python programming language in 1980

python

Why Learn Python?

- Python is simple and popular
- Python is modular and extensible
- Python is extremely versatile
  - data science, AI & ML, web dev, automation, testing
- It has active community
- Good career opportunities and salary

  Source: https://codingnomads.co/why-learn-python/

Programming Environment

For the programming environment we will use:

- **Ubuntu Linux** as a platform

- **Git** and **GitHub** as version control system

- **Virtualenv** as a Python environment

- **Python 3** as a programming language

- Any **editor** for code writing (nano, gedit, vi)

Ubuntu Linux

There are several ways to install **Ubuntu Linux**, including:

- **Standalone** installation of Ubuntu on your machine
- **Dual** installation of Ubuntu together with Windows
- Installation of Ubuntu into **Virtualbox** on Windows
- Installation of Ubuntu with **wubi** on Windows
- Rent a **virtual machine** with Ubuntu on Amazon or other

Ubuntu Linux

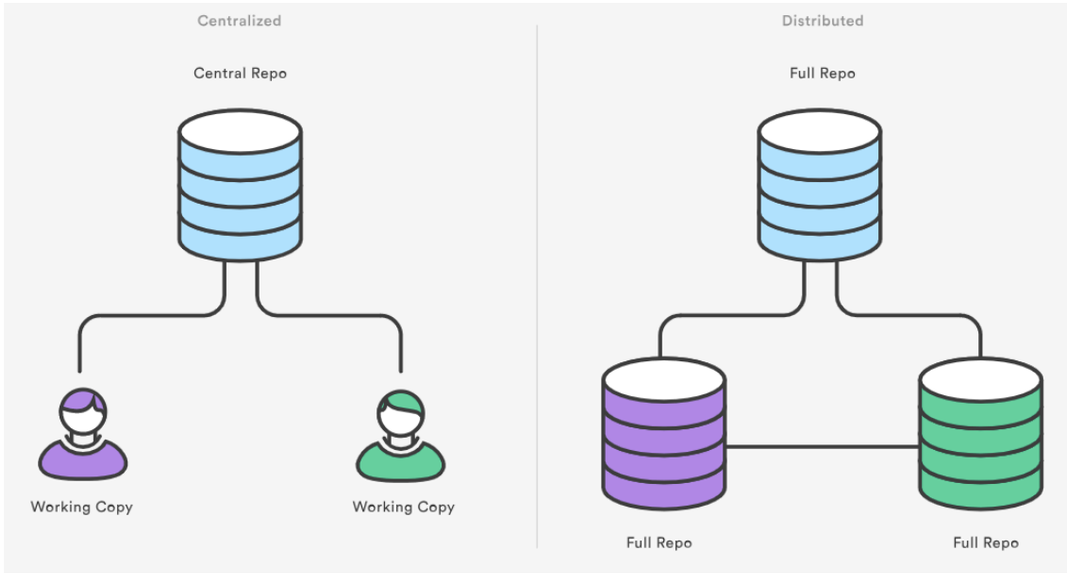
Some useful link s:

- Ubuntu - https://www.ubuntu.com/download/desktop
- Virtualbox - https://www.virtualbox.org/wiki/Downloads
- Wubi - https://github.com/hakuna-m/wubiuefi/releases
- Amazon - https://aws.amazon.com/getting-started/tutorials/launch-a-virtual-machine/

Git

Git is a free and open source distributed version control system, which features:

- distributed architecture
- performance and small size
- easy and flexible branching
- cryptographic data integrity
- de-facto standard in development



SVN, CVS **Git**

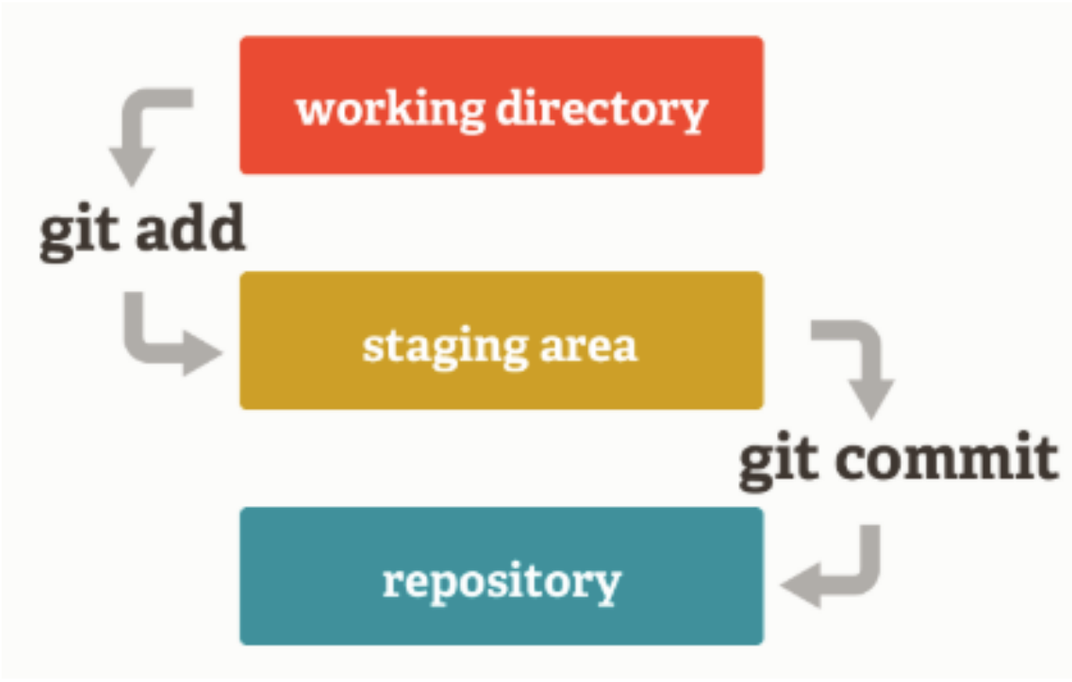
Git

```
# to create or clone a new git repository
git init
git clone username@host:/path/to/repository

# add (stage) and commit files
git add <filename>
git commit -m "Commit message”

# pushing changes to server
git push origin main
git remote add origin <server>

# get various info
git status
git log --graph --oneline --decorate --all
```



Github

GitHub is a web-based code hosting platform for version control and collaboration

Usage:

- Create a **GitHub.com** account **USERNAME**
- Create a new repository **REPO** (Step 1) using the tutorial: https://guides.github.com/activities/hello-world/
- Clone your repository to the local machine with the Git command: *git clone https://github.com/**USERNAME**/**REPO**.git*
- Start working locally on your project

Virtualenv

virtualenv is a tool to create **isolated** Python environments

- allowing to install only needed packages for the project
- managing the packages locally (without root privileges)

Usage (in bash):

```
$ pip install virtualenv
$ mkdir my_env    # environment folder
$ # setup environment
$ virtualenv -p /usr/bin/python3 my_env
$ source my_env/bin/activate
(my_env)$ pip install <packages>
(my_env)$ # do some work in Python
(my_env)$ deactivate    # finish work
```

Python 3

- The latest version is Python 3.9.7
- Why learn Python 3?

Python 2	Python 3
Legacy Python 2 ends support after 2020	**Future**
ASCII Strings are stored in ASCII by default	**UNICODE** Strings are stored in UNICODE by default
5/2 = 2	**5/2 = 2.5**
print “hello”	**print(“hello”)**

Source:
https://learntocodewith.me/programming/python/python-2-vs-python-3/#2018-differences-of-python2-vs-3

Coding Conventions

- The official **Style Guide for Python Code** is available online at http://www.python.org/dev/peps/pep-0008/
- Python code blocks are typically indented by 4 spaces.
- Use meaningful names for identifiers
  - **Classes** are *CamelCase*
  - **Functions** are lowercase [with underscore]: *distance( ), print_xy( )*
  - **Variables** are lowercase: *student, price, …*
  - **Constants** are uppercase [with underscore]: FLAG, MAX_INT
- Classes and functions must have **docstring, (“””some text”””)**

Let’s get started!