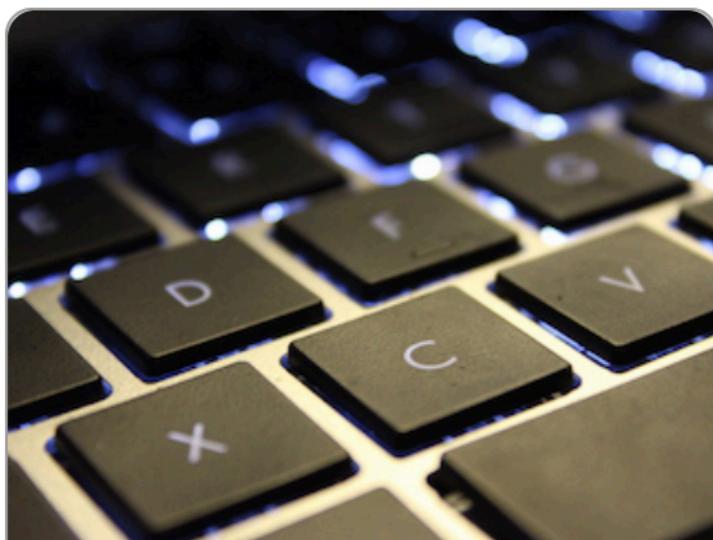


OOP Python

Prof. Nick Wang

Introduction to Computer Science and Programming in Python

- <https://github.com/ARG-NCTU/oop-python-nycu/tree/main/mit-ocw-exercises>
- <https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/>



6.0001 is intended to teach basic programming concepts to students with no prior coding

COURSE DESCRIPTION

6.0001 Introduction to Computer Science and Programming in Python is intended for students with little or no programming experience. It aims to provide students with an understanding of the role computation can play in

COURSE INFO

Instructors:	Dr. Ana Bell Prof. Eric Grimson Prof. John Guttag
Course Number:	6.0001
Departments:	Electrical Engineering and Computer Science

Learning Objectives in this course

- Lecture: Monday n56
 - Watch OCW video
 - More programming related topics
- Interaction: Monday abc
 - Interact with TA and your team
 - Finish the TA activity and demo
 - Work on lecture code
 - Ask questions to ChatGPT
 - Share your **sequence of questions**



Lecture 5: Tuples, Lists, Aliasing, Mutability, and Cloning



Lecture 6: Recursion and Dictionaries



Lecture 7: Testing, Debugging, Exceptions, and Assertions



Lecture 8: Object Oriented Programming



Lecture 9: Python Classes and Inheritance

SES #: 5

TOPICS: Tuples, Lists, Aliasing, Mutability, Cloning

LECTURE SLIDES: [Slides for Lecture 5 \(PDF\)](#)

LECTURE CODES: [Code for Lecture 5 \(PY\)](#)

SES #: 6

TOPICS: Recursion, Dictionaries

LECTURE SLIDES: [Slides for Lecture 6 \(PDF - 1.3MB\)](#)

LECTURE CODES: [Code for Lecture 6 \(PY\)](#)

How will we use ChatGPT?

Learning to ASK!

- Fun Perspective:
 - Humans are an API to ChatGPT.
 - **ChatGPT is an API to Python.**
 - **Python is an API to C.**
 - C is an API to assembly.
 - Assembly is an API to binary.
 - Binary is an API to physics.
 - Physics is an API to the machine that runs the universe.
- It's computation all the way down.



How will we use ChatGPT?

Learning to ASK!

- 學術倫理
 - <https://icml.cc/Conferences/2023/lm-policy>
 - The Large Language Model (LLM) policy for ICML 2023 prohibits text produced entirely by LLMs (i.e., “generated”). This does not prohibit authors from using LLMs for editing or polishing author-written text.
 - The LLM policy is largely predicated on the principle of being conservative with respect to guarding against potential issues of using LLMs, including plagiarism.

How will we use ChatGPT?

Learning to ASK!

- Will you answer exactly the same answers every time I ask the same question?
- Can you guess what questions I may ask you next?

Ask ChatGPT

Data Types in Python (1)

- What are the unique data types for Python over C++?
- Please give an example of dictionary?
- How can I loop in the dictionary you provide?
- Can I use an index to loop for dictionary?
- How to find unique values in dictionary?
- What's the difference of this set() function compared to the set data type?
- You just mentioned that set is an unordered collection; is there a data type for ordered collection?
- Why do we need an unordered collection like set? Why not just use list?
- What would happen if I intended to add an item already exist in a set?

Ask ChatGPT

Data Types in Python (2)

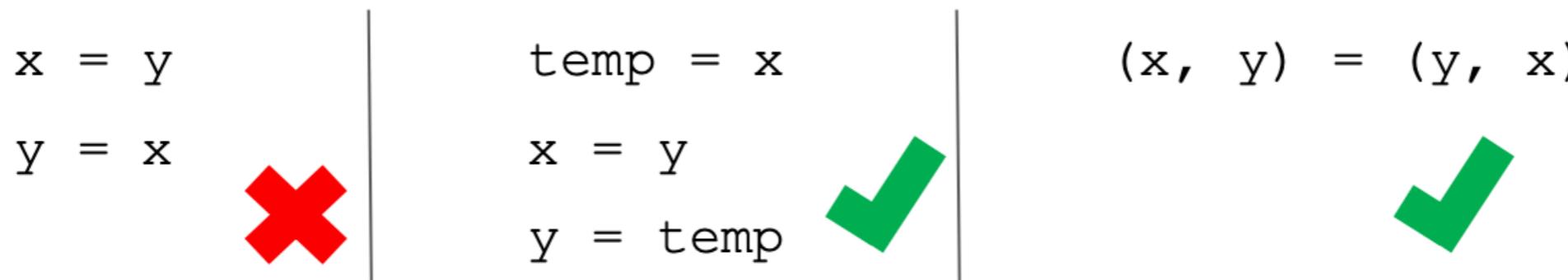
- Can I have identical key in dictionary?
- Is the key case sensitive?
- When do we use Tuple over list?
- Give an example of unpacking?
- From the example you provide, it seems that we did not specify the returning values are within a tuple?
- Is there a situation that I need to specify tuple?
- Why tuple use immutable? What's the difference between immutable and constant variables?
- How does the data in dictionary saved to files? Any popular format?
- What's the deep copy in Python?
- How do I deep copy a list?
- A deep copy example in OpenCV Python?
- How about a shallow copy for OpenCV?

Tuple vs. List

- They are both used to store collection of data
- They are both **heterogeneous/mixed data types** means that you can store any kind of data type
- They are both **ordered** means the order in which you put the items are kept.
- They are both sequential data types so you can **iterate** over the items contained.
- Items of both types can be accessed by an integer index operator, provided in square brackets, **[index]**
- <https://towardsdatascience.com/python-tuples-when-to-use-them-over-lists-75e443f9dcd7>

When to use Tuple?

- immutable, more memory efficient
- data not meant to be changed
- tuples can be used as keys in dictionaries and as elements of sets, while lists cannot
- conveniently used to **swap** variable values



- used to **return more than one value** from a function

```
def quotient_and_remainder(x, y):  
    q = x // y  
    r = x % y  
    return (q, r)  
  
(quot, rem) = quotient_and_remainder(4, 5)
```

integer
division

Lists: More Operations

- delete element at a specific index with `del(L[index])`
- remove element at end of list with `L.pop()`, returns the removed element
- remove a specific element with `L.remove(element)`
 - **add** elements to end of list with `L.append(element)`
 - **mutates** the list!

`L = [2, 1, 3]`

`L.append(5)` → L is now [2, 1, 3, 5]



- what is the dot?
 - lists are Python objects, everything in Python is an object
 - objects have data
 - objects have methods and functions
 - access this information by `object_name.do_something()`
 - will learn more about these later

Iterating a List

- https://github.com/ARG-NCTU/oop-python-nycu/blob/main/mit-ocw-exercises/lec5_tuples_lists.py

- compute the **sum of elements** of a list

- common pattern, iterate over list elements

```
total = 0
for i in range(len(L)):
    total += L[i]
print total
```

```
total = 0
for i in L:
    total += i
print total
```

like strings,
can iterate
over list
elements
directly

- notice

- list elements are indexed 0 to len(L) -1
- range(n) goes from 0 to n-1

```
55 #####
56 ## EXAMPLE: sum of elements in a list
57 #####
58 def sum_elem_method1(L):
59     total = 0
60     for i in range(len(L)):
61         total += L[i]
62     return total
63
64 def sum_elem_method2(L):
65     total = 0
66     for i in L:
67         total += i
68     return total
69
70 print(sum_elem_method1([1,2,3,4]))
71 print(sum_elem_method2([1,2,3,4]))
```

list vs dict

- **ordered** sequence of elements
- look up elements by an integer index
- indices have an **order**
- index is an **integer**
- You are encourage to run the code!
[https://github.com/ARG-NCTU/oop-python-nycu/blob/main/
mit-ocw-exercises/lec5_tuples_lists.py](https://github.com/ARG-NCTU/oop-python-nycu/blob/main/mit-ocw-exercises/lec5_tuples_lists.py)
- **matches** “keys” to “values”
- look up one item by another item
- **no order** is guaranteed
- key can be any **immutable** type

<https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/>

Ask ChatGPT

Python Function

- What is a generator?
- So a function of generator has the yield keyword instead of return?
- Could you print the result of the example you just show?
- Can we just use a regular function to implement the generator you just show?
- Why use an iterator not just loop over via the index?
- Does iterator deal with two dimensional data, compared to index based loops?

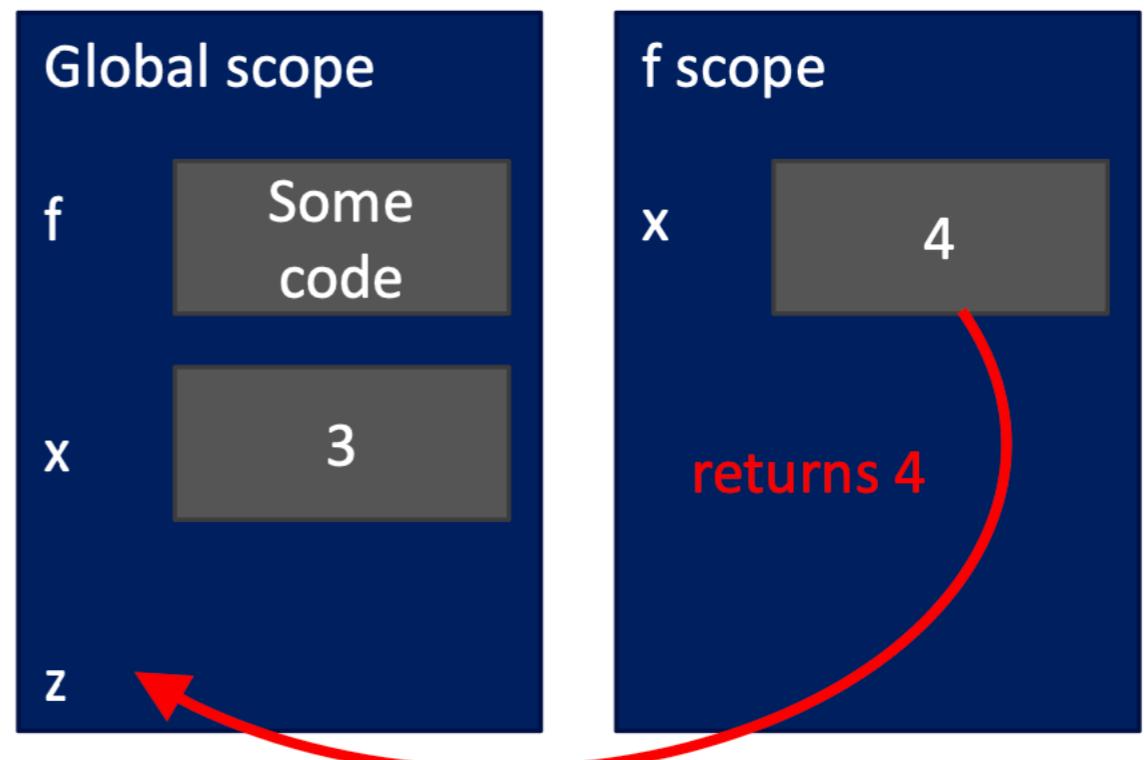
Python function/class docstring (__doc__)

- <https://betterprogramming.pub/3-different-docstring-formats-for-python-d27be81e0d68>

```
keyword      name      parameters or arguments      specification, docstring
def is_even( i ):  
    """  
        Input: i, a positive int  
        Returns True if i is even, otherwise False  
    """  
  
    body  
    print("inside is_even")  
    return i%2 == 0  
  
is_even(3)  
  
later in the code, you call the function using its name and values for parameters
```

Variable Scope

```
def f( x ):  
    x = x + 1  
    print('in f(x): x =', x)  
    return x  
  
x = 3  
z = f( x )
```



Survive Command Line

Ubuntu Cheatsheet

- [https://oceanai.mit.edu/ivpman/pmwiki/pmwiki.php?
n=Lab.ClassSetup#sec_command_line](https://oceanai.mit.edu/ivpman/pmwiki/pmwiki.php?n=Lab.ClassSetup#sec_command_line)
- You may open multiple terminals at the same time. This may be tricky for remote machine.
- Local and remote terminals. Check your bash. Suggested commands
 - \$ byobu # for remote
 - \$ terminator # for local
- Paste in a terminal: Ctl + Shift + v
- Search for files: \$ grep

System-level

- \$ sudo apt install
 - /usr/local/
 - /etc/
- \$ sudo reboot
- \$ sudo shutdown -h now
- About sudo
 - Never sudo on any user-level usages

User-level

- \$ cd ~/
 - /home/[username]/
- .bashrc
 - Add path; we use this a lot!
- User's settings
 - .ssh/
 - .vimrc

GitHub Cheatsheet

- Cheat Sheet

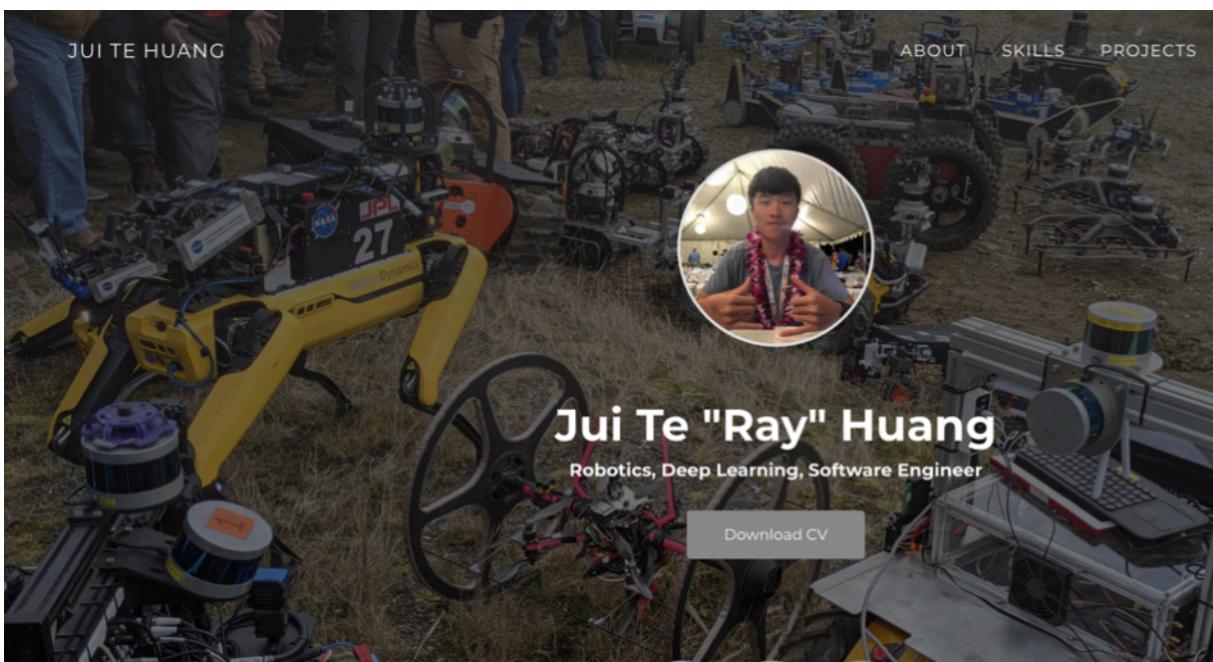
<https://education.github.com/git-cheat-sheet-education.pdf>

- SSH Key
- github.io
- Do NOT use someone's account or "team account"



經營你的Github

- GitHub: <https://github.com/huangjuite>
- Webpage: <https://huangjuite.github.io/>



Github帳號 命名原則

- 帳號不要有大寫 (有大寫基本上是Organization)
- 帳號不要有一堆數字，像是學號，很多oooowww有點怪怪的，之後你找工作這也會是第一印象，這個帳號會跟著你很久，最好用你的名字
- 帳號最好是“timeless”，例如：nctu, nycu不一定很好，未來你可能去別的地方深造或工作。
- 帳號盡量不要有其他符號“-”, “.”等等，基本上跟命名hostname類似
- repo命名: <https://stackoverflow.com/questions/11947587/is-there-a-naming-convention-for-git-repositories> repo名稱也會有命名原則，一般會用some-node-about-ros這樣的格式，命名隨便的話雖然還是能跑，但這會讓內行的人第一眼覺得你是不是來亂的，連convention都不知道，還有repo名稱要反映出裡面在做什麼，例如：graphviz-ros-diagram

Git

Version Control Systems

- \$ git –version
- \$ git config –global user.name “Nick Wang”
- \$ git config –global user.email hchengwang@gmail.com
- \$ git clone
- \$ git clone –recursive

Git Workflow

- \$ git status
- \$ git add
- I don't like -a
- \$ git commit -m “xxx”
- \$ git push

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

`git status`

show modified files in working directory, staged for your next commit

`git add [file]`

add a file as it looks now to your next commit (stage)

`git reset [file]`

unstage a file while retaining the changes in working directory

`git diff`

diff of what is changed but not staged

`git diff --staged`

diff of what is staged but not yet committed

`git commit -m “[descriptive message]”`

commit your staged content as a new commit snapshot

Repo Files

- .gitignore
- \$ git diff
- \$ git rm
- \$ git mv
- \$ git log

IGNORING PATTERNS

Preventing unintentional staging or committing of files

```
logs/  
*.notes  
pattern*/
```

TRACKING PATH CHANGES

Versioning file removes and path changes

```
git rm [file]
```

delete the file from project and stage the removal for commit

```
git mv [existing-path] [new-path]
```

change an existing file path and stage the move

```
git log --stat -M
```

show all commit logs with indication of any paths that moved

Git Branch

- \$ git branch
- \$ git checkout
- Pull Request (PR)
- Merge PR

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

`git branch`

list your branches. a * will appear next to the currently active branch

`git branch [branch-name]`

create a new branch at the current commit

`git checkout`

switch to another branch and check it out into your working directory

`git merge [branch]`

merge the specified branch's history into the current one

`git log`

show all commits in the current branch's history

Git Merge Conflicts

- You may find that you are not able to push, while your teammate edited the same file.
- `$ git stash`
- `$ git pull`
- `$ git stash pop`
- Edit the conflict file
- `$ git push`

TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

git stash

Save modified and staged changes

git stash list

list stack-order of stashed file changes

git stash pop

write working from top of stash stack

git stash drop

discard the changes from top of stash stack

oop-python-nycu

- <https://github.com/ARG-NCTU/oop-python-nycu>
- A refresher for OOP C++ and Python
- Get familiar with GitHub, Docker, and Vim
- Running local runtime (kernel) for a jupyter notebook
- Parts of the OOP Course in Spring 2023

oop-python-nycu

☰ README.md

For first time user or any Dockerfile update

```
docker pull argnctu/oop:latest
```

Clone the repo

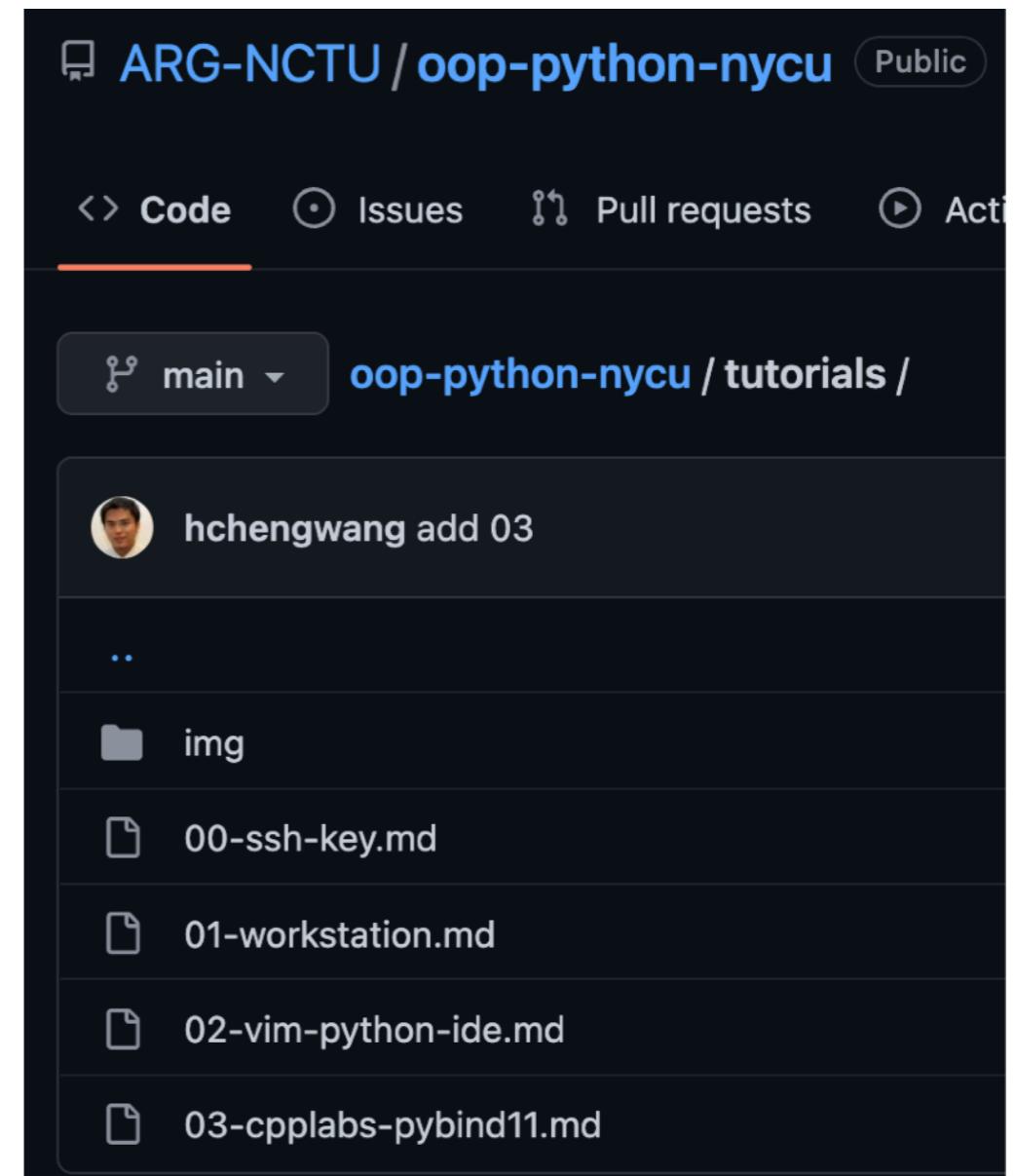
```
git clone git@github.com:ARG-NCTU/oop-python-nycu.git
```

or if you do not have ssh key yet (you should set it up soon)

```
git clone https://github.com/ARG-NCTU/oop-python-nycu.git
```

For your first docker terminal:

```
source docker_run.sh
```



vim-python-ide

- <https://github.com/ARG-NCTU/vim-python-ide.git>
- <https://github.com/ARG-NCTU/oop-python-nycu/blob/main/tutorials/02-vim-python-ide.md>
- You are able to code remotely with a nice IDE.

A screenshot of the Vim editor interface. On the left is a terminal window showing a file named 'Dockerfile' with the following content:

```
Dockerfile docker_run.sh | docker_join.sh |
" Press ? for help

.. (up a dir)
</hchengwang/oop-python-nycu/
▶ cpplabs/
▶ Docker/
▶ mit-ocw-exercises/
▶ tutorials/
  colab_jupyter.sh*
docker_join.sh*
docker_run.sh*
README.md
tags

76
77 RUN pip3 install --upgrade pip setuptools
78
79 RUN pip3 install \
80   gdown \
81   virtualenv \
82   pytest \
83   flask \
84   flask-socketio \
85   jupyter \
86   virtualenv \
87   nbdev
88
89 ##### vim-python-ide #####
90 #####
91 RUN git clone https://github.com/ARG-NCTU/vim-python-ide.git && \
92   cd vim-python-ide && ./install.sh
93
```

To the right of the terminal is a 'Key bindings' menu:

Action	Mode	key binding
Toogle NerdTree	Normal	crtl + n
Toogle Tagbar	Normal	f8
Run Python.	Normal	f9
Switch Focus	Normal	crtl + ww
Expand snippet	Insert	tab

Key bindings

What is Docker?

- Docker is a tool that allows developers, sys-admins etc. to easily deploy their applications in a sandbox (called *containers*) to run on the host operating system i.e. Linux.
- The key benefit of Docker is that it allows users to **package an application with all of its dependencies into a standardized unit** for software development.



Docker vs. Virtual Machine (VM)

- The industry standard today is to use Virtual Machines (VMs) to run software applications. VMs run applications inside a **guest** Operating System, which runs on virtual hardware powered by the server's **host** OS.
- VMs are great at providing **full process isolation** for applications: there are very few ways a problem in the host operating system can affect the software running in the guest operating system, and vice-versa. But this isolation comes at great cost — the **computational overhead** spent virtualizing hardware for a guest OS to use is substantial.

Docker Container

- Unlike virtual machines, **containers** do not have high overhead and hence enable more **efficient** usage of the underlying system and resources.
- Containers take a different approach: by **leveraging the low-level mechanics** of the host operating system, containers provide **most of the isolation** of virtual machines at a fraction of the computing power.
- Due to these benefits, containers (& Docker) have seen **widespread adoption**. Companies like Google, Facebook, Netflix and Salesforce leverage containers to make large engineering teams more productive and to improve utilization of compute resources.

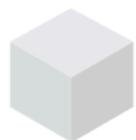
Docker Terminology

- **Images** - The blueprints of our application which form the basis of containers.
- **Containers** - Created from Docker images and run the actual application.
- **Docker Daemon** - The background service running on the host that manages building, running and distributing Docker containers. The daemon is the process that runs in the operating system which clients talk to.
- **Docker Client** - The command line tool that allows the user to interact with the daemon.
- **Dockerfile** - A simple text file that contains a list of commands that the Docker client calls while creating an image. It's a simple way to automate the image creation process. The best part is that the commands you write in a Dockerfile are almost identical to their equivalent Linux commands.

Docker Hub and Widely Used Docker Containers

- Docker Hub - A registry of Docker images. You can think of the registry as a directory of all available Docker images. If required, one can host their own Docker registries and can use them for pulling images.
- **Docker Hub**, Docker's public registry, can be your central source for 100,000+ container apps, images and repositories.

PRODUCT	DOCKER HUB PULLS	DOCKER HUB STARS
Wordpress	10,000,000	4169
Ubuntu	10,000,000	233
MySQL	10,000,000	225
MongoDB	10,000,000	168



osrf/ros

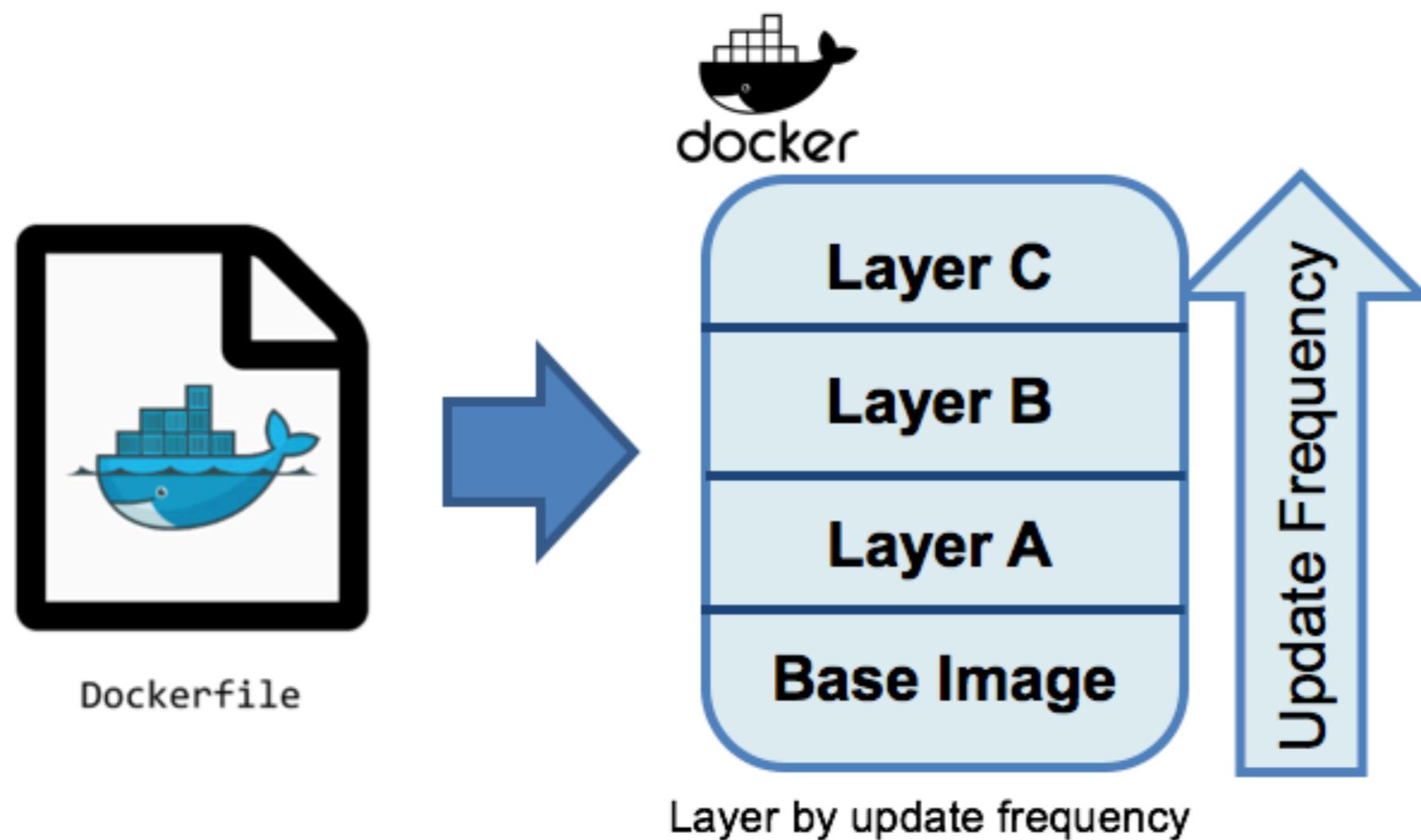
By [osrf](#) • Updated 23 days ago

500K+ 89
Downloads Stars

The Robot Operating System (ROS) is an open source project for building robot applications.

docker pull, Dockerfile, and layers

- Docker Hub provides the following major features:
 - [Repositories](#): Push and pull container images.
 - [Teams & Organizations](#): Manage access to private repositories of container images.
 - [Official Images](#): Pull and use high-quality container images provided by Docker.



Dockerfile

```
D/Dockerfile docker_run.sh | docker_join.sh |
" Press ? for help

.. (up a dir)
</hchengwang/oop-python-nycu/
► cpplabs/
► Docker/
► mit-ocw-exercises/
► tutorials/
  colab_jupyter.sh*
  docker_join.sh*
  docker_run.sh*
README.md
tags
~
~
```

```
1 FROM ubuntu:20.04
2
3 ENV DEBIAN_FRONTEND noninteractive
4 ENV LANG C.UTF-8
5 ENV LC_ALL C.UTF-8
6
7 #####
8
9 ENV SHELL=/bin/bash \
10    USER=arg \
11    UID=1000 \
12    LANG=en_US.UTF-8 \
13    LANGUAGE=en_US.UTF-8
14
15 ENV HOME=/home/${USER}
16
```

```
27 RUN apt-get -o Acquire::ForceIPv4=true update && apt-get -yq dist-upgrade \
28   && apt-get -o Acquire::ForceIPv4=true install -yq --no-install-recommends \
29     locales \
30     make \
31     cmake \
32     git \
33     vim \
34     gedit \
35     wget \
36     sudo \
37     lsb-release \
38     build-essential \
39     net-tools \
40     dirmngr \
41     gnupg2 \
42     mercurial \
43     libopencv-dev \
44     libcanberra-gtk-module \
45     libcanberra-gtk3-module \
46     libpcap-dev \
47     python3-pip \
48     python3-setuptools \

```

docker run and exec

- We usually **docker run** **one container in a machine** in most ROS applications.
 - Web applications often use multiple containers via docker-compose (e.g., web server + mysql)
- We may use multiple **docker exec** to start new terminals and enter the (only) container.
- The container will share the same network settings with the host.
- All changes in container will lost when you terminate it.

docker_run.sh

Docker with Filesystem Access

