

ISOM 3400 Lab 2

Agenda

- Use of VSCode
- Virtual Environment: Setup and implement
- Python Program: Example

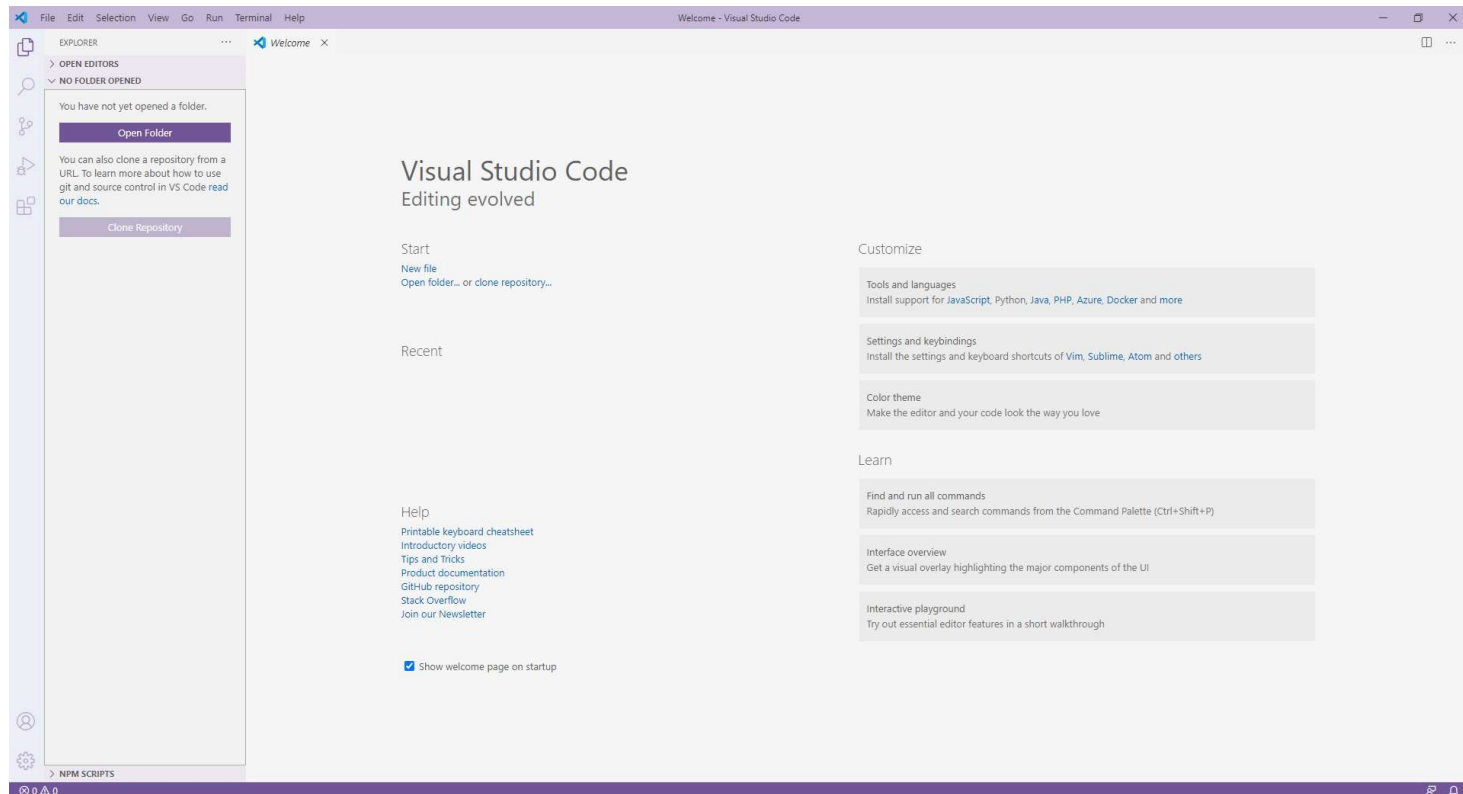
Use of VSCode

Use of VSCode - Prerequisite

- Install **Anaconda** to your computer
- Install **VSCode** to your computer
- Extension **Python** installed to your **VSCode**
- Review Lab 1 Video if necessary

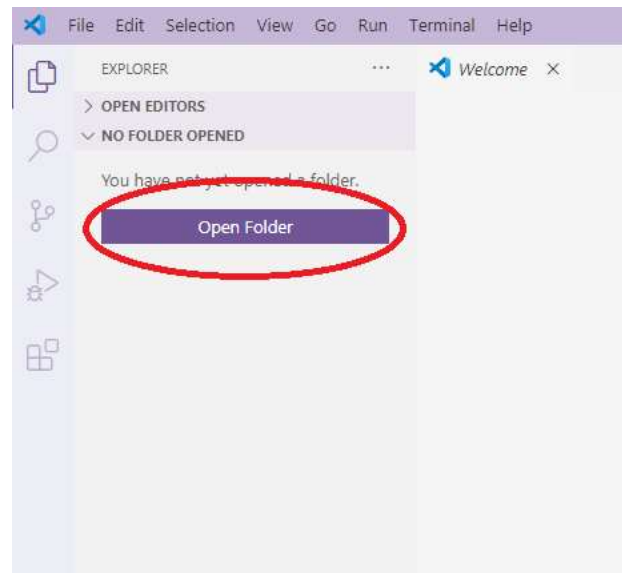
Use of VSCode

- Launch VS Code, you should see interface like the screenshot below



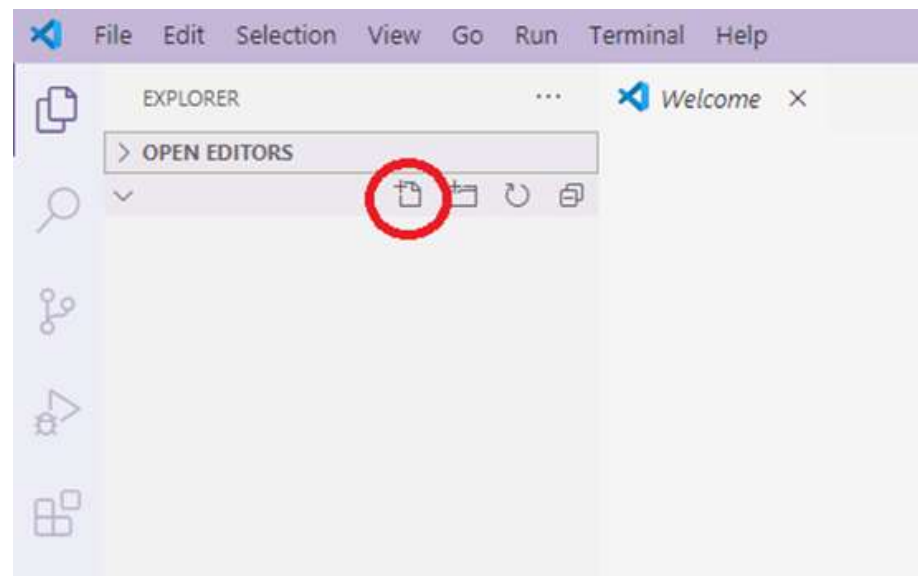
Use of VSCode

- Before creating a Python script (.py file), you need to tell VS Code where to save the file
- Click on **Open Folder**, or **Ctrl+K Ctrl+O** (**Command + O** in Mac), and choose a destination



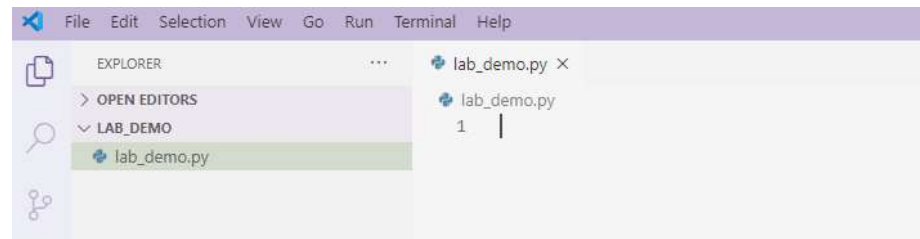
Use of VSCode

- **Move your cursor** to the title bar and click the **first icon** to create a new file



Use of VSCode

- In addition to file name, you need to specify the format
- It will be **.py** for Python script. After that, press **Enter** and you will see a file created
- This is where you do your work



Use of VSCode - Illustration

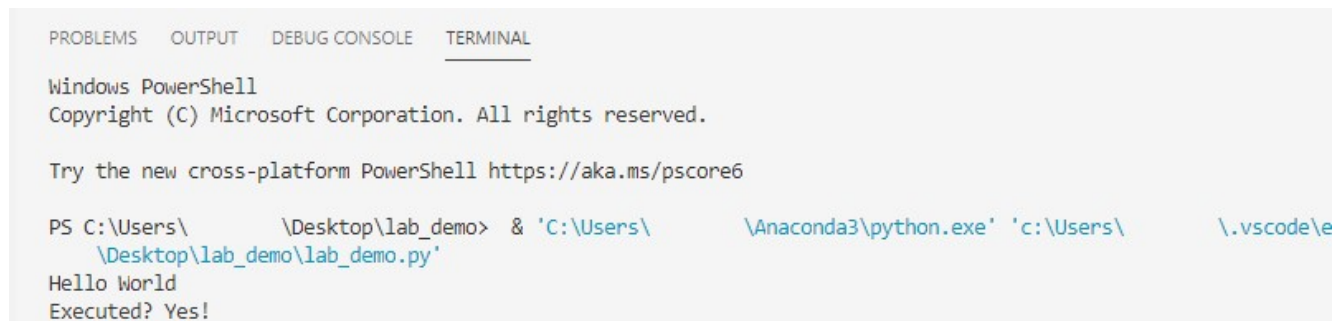
- What is going on in the code?



```
lab_demo.py •
lab_demo.py > ...
1  a = 3
2  b = "Hello" # This is a comment
3  # Comment also
4  print("Hello World")
5  c = "ABCDEFGH"
6  print("Executed? Yes!") # Comment will not be executed
7
```

Use of VSCode - Illustration

- Press **F5** (**fn + F5** in Mac), select **Python File Debug the currently active Python file** as debug configuration
- Things we included in the print functions are printed out



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

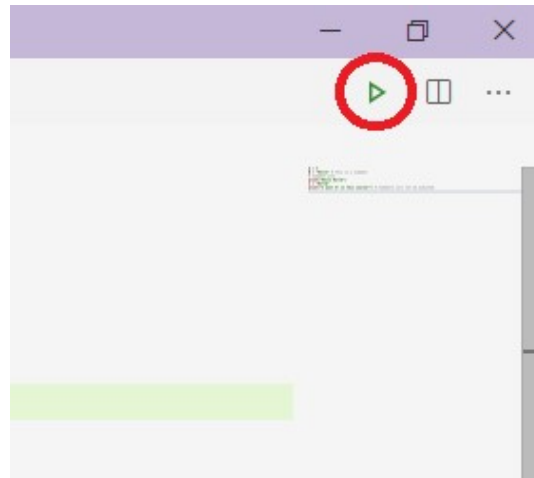
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\          \Desktop\lab_demo> & 'C:\Users\          \Anaconda3\python.exe' 'c:\Users\          \.vscode\e
\Desktop\lab_demo\lab_demo.py'
Hello World
Executed? Yes!
```

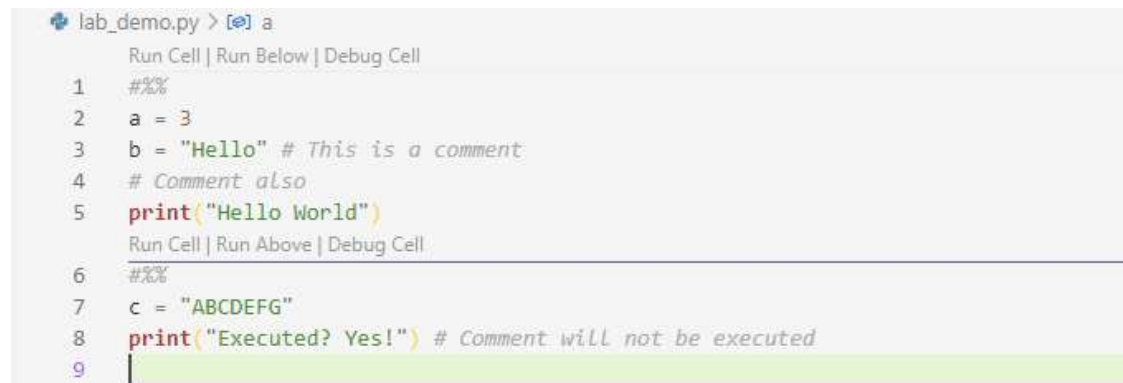
Use of VSCode - Illustration

- Alternatively, you can click the button on the **top-right** corner to run the script



Use of VSCode - Illustration

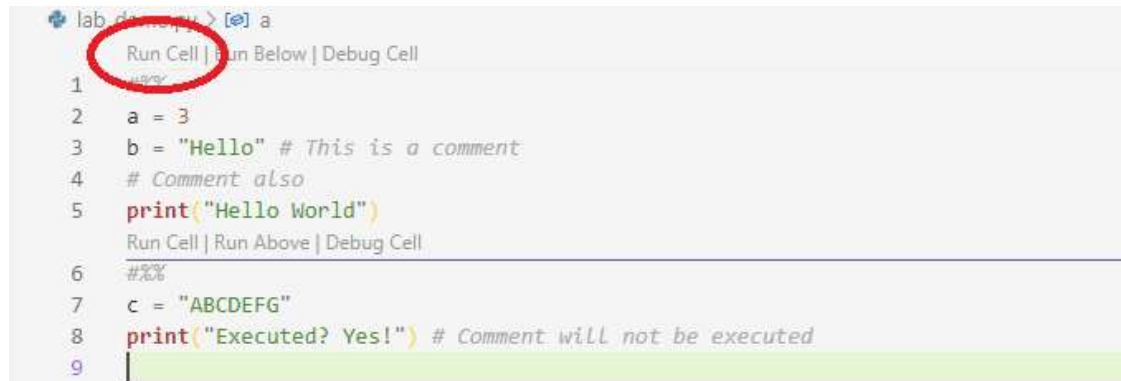
- What to do when you only want to run particular lines only?
- **Code Cell & Interactive Window**



```
lab_demo.py > [a]
Run Cell | Run Below | Debug Cell
1  ###
2  a = 3
3  b = "Hello" # This is a comment
4  # Comment also
5  print("Hello World")
Run Cell | Run Above | Debug Cell
6  ###
7  c = "ABCDEFGH"
8  print("Executed? Yes!") # Comment will not be executed
9
```

Use of VSCode - Illustration

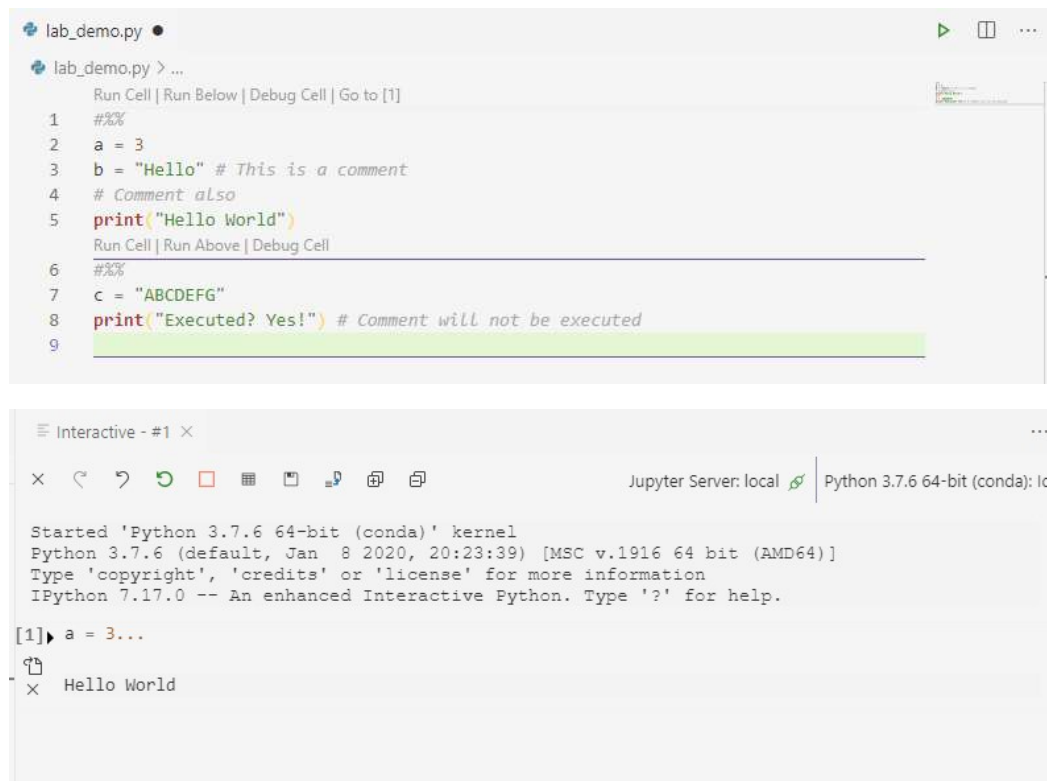
- Click **Run Cell** button to run the cell
- Try **Ctrl + Enter** / **Shift + Enter**



The screenshot shows a Jupyter Notebook cell in VS Code. The cell's toolbar at the top contains three buttons: 'Run Cell', 'Run Below', and 'Debug Cell'. The 'Run Cell' button is circled in red. Below the toolbar, the code in the cell is as follows:

```
1 user
2 a = 3
3 b = "Hello" # This is a comment
4 # Comment also
5 print("Hello World")
6 Run Cell | Run Above | Debug Cell
7 %%
8 c = "ABCDEFGH"
9 print("Executed? Yes!") # Comment will not be executed
```

Use of VSCode - Illustration



The image shows a screenshot of the Visual Studio Code (VS Code) interface. The top panel displays a Python file named `lab_demo.py`. The code contains two code cells. The first cell has the following code:

```
1  #%%
2  a = 3
3  b = "Hello" # This is a comment
4  # Comment also
5  print("Hello World")
```

The second cell has the following code:

```
6  #%%
7  c = "ABCDEFGF"
8  print("Executed? Yes!") # Comment will not be executed
9
```

The bottom panel shows the Jupyter Notebook output for the first cell. It displays the following text:

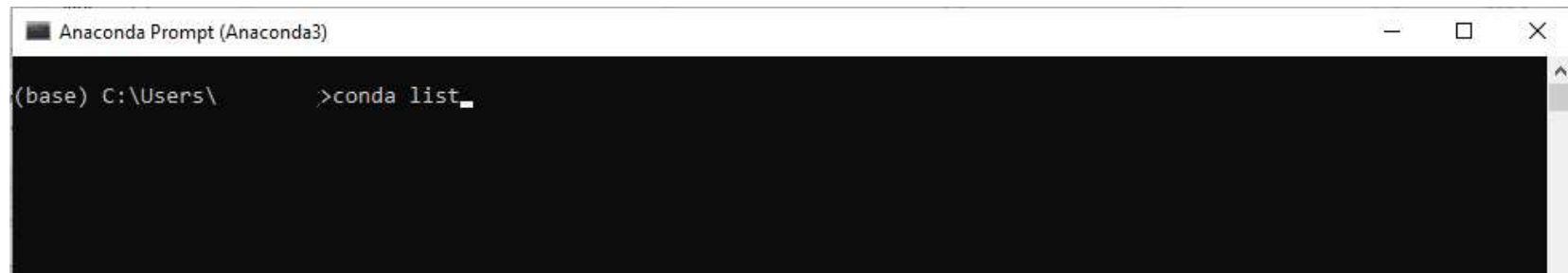
```
Started 'Python 3.7.6 64-bit (conda)' kernel
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.17.0 -- An enhanced Interactive Python. Type '?' for help.

[1] a = 3...
Hello World
```

Working with Virtual Environment

Working with Virtual Environment

- **Windows User:** Launch **Anaconda Prompt (Anaconda3)**
- **Mac User:** Launch **Terminal**
- Try **conda list** and press **Enter**
- What happens?



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\ >conda list
```


Working with Virtual Environment

- Check Python version: **python --version**
- What environments do you have in your computer? **conda info --envs**

```
(base) C:\Users\  
# conda environments:  
#  
?
```

Working with Virtual Environment

- Command to create an environment:
- **conda create --name environment name python(=version number)**
- By default, the command will create an environment with an updated **python version**
- If you need an older version, specify it
- Try to create one by yourself

```
(base) C:\Users\ >conda create --name labdemo python_
```

Working with Virtual Environment

- Type **y** and press **Enter** to proceed

```
Anaconda Prompt (Anaconda3) - conda create --name labdemo python

pip-20.3.3      | py39haa95532_0      | 1.8 MB
python-3.9.1    | h6244533_2          | 16.4 MB
setuptools-52.0.0 | py39haa95532_0      | 725 KB
sqlite-3.33.0   | h2a8f88b_0          | 809 KB
tzdata-2020f    | h52ac0ba_0          | 113 KB
vc-14.2         | h21ff451_1          | 8 KB
vs2015_runtime-14.27.29016 | h5e58377_2        | 1007 KB
wheel-0.36.2    | pyhd3eb1b0_0        | 33 KB
wincertstore-0.2 | py39h2bfff1b_0      | 15 KB
-----
Total:          | 25.9 MB

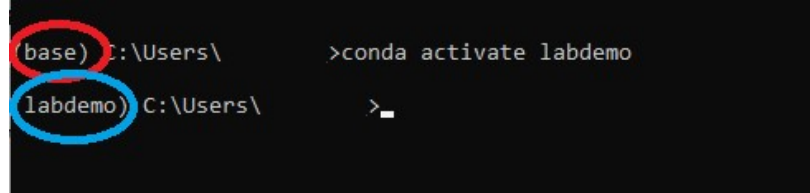
The following NEW packages will be INSTALLED:

ca-certificates  pkgs/main/win-64::ca-certificates-2021.1.19-haa95532_0
certifi          pkgs/main/win-64::certifi-2020.12.5-py39haa95532_0
openssl          pkgs/main/win-64::openssl-1.1.1i-h2bfff1b_0
pip              pkgs/main/win-64::pip-20.3.3-py39haa95532_0
python           pkgs/main/win-64::python-3.9.1-h6244533_2
setuptools       pkgs/main/win-64::setuptools-52.0.0-py39haa95532_0
sqlite           pkgs/main/win-64::sqlite-3.33.0-h2a8f88b_0
tzdata          pkgs/main/noarch::tzdata-2020f-h52ac0ba_0
vc               pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime  pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel            pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
wincertstore     pkgs/main/win-64::wincertstore-0.2-py39h2bfff1b_0
zlib             pkgs/main/win-64::zlib-1.2.11-h62dcd97_4

Proceed ([y]/n)? y_
```

Working with Virtual Environment

- After the creation of environment, we are still working with the base environment.
- Activate environment: **conda activate environment name**

A terminal window with a black background and white text. The first line shows the prompt '(base) C:\Users\' followed by the command '>conda activate labdemo'. The second line shows the prompt '(labdemo) C:\Users\' followed by a cursor. A red circle highlights the '(base)' prompt, and a blue circle highlights the '(labdemo)' prompt.

```
(base) C:\Users\ >conda activate labdemo
(labdemo) C:\Users\ >_
```

Working with Virtual Environment

- Try **conda list** and **python --version**
- Any difference compared to the base environment you are using?
- Try **conda install ipykernel**

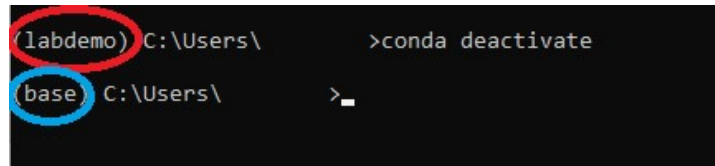
```
(labdemo) C:\Users\ >conda list
# packages in environment at C:\Users\ \Anaconda3\envs\labdemo:
#

Python 3.7.4

(labdemo) C:\Users\ >python --version
Python 3.7.4
```

Working with Virtual Environment

- If you do not want to work in this environment anymore, you can simply type **conda deactivate** to go back to the base environment
- Delete environment: **conda env remove --name environment name**

A terminal window with a dark background. The first line shows the prompt '(labdemo) C:\Users\' followed by the command '>conda deactivate'. The second line shows the prompt '(base) C:\Users\' followed by '>_'. The prompt '(labdemo)' is circled in red, and the prompt '(base)' is circled in blue.

```
(labdemo) C:\Users\ >conda deactivate
(base) C:\Users\ >_
```

Virtual Environment and VS Code for Python Programming

(Quick Start)

By James S. H. Kwok

Virtual Environment

Required	Optional
Step 1: Go and launch Terminal	
	Step 2: change to the destination folder cd << destination>>
	Step 3: Show all virtual environment conda env list
Step 4: Create a virtual environment conda create -n isom3400 python=3.8	
Step 5: Activate the virtual environment conda activate isom3400	
	Step 6: Show all package conda list
Step 7: Install a package conda install pandas	
	Step 8: Exit the virtual environment conda deactivate

VS Code

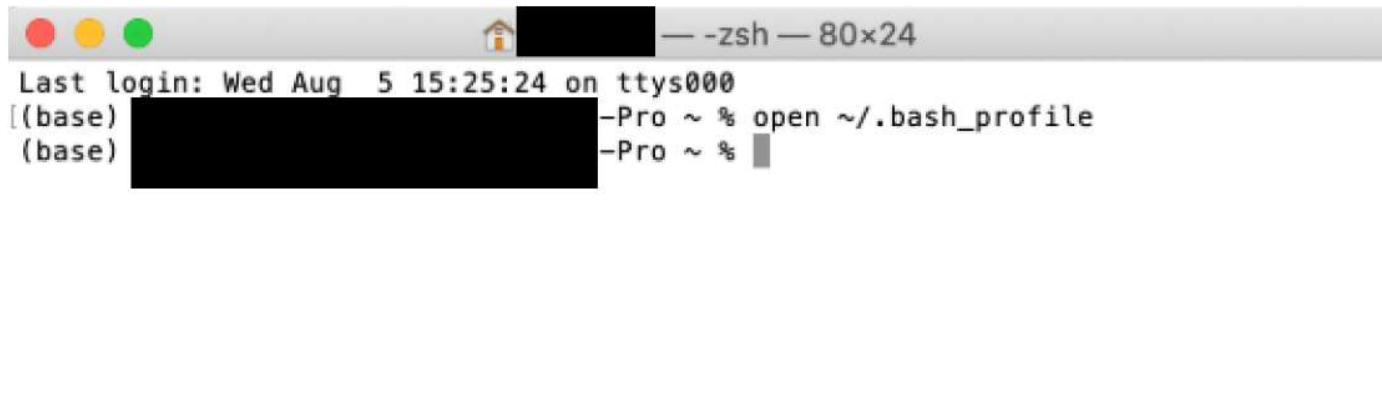
Step 1: Launch VS Code start code
Step 2 (optional): Change to working directory File – Open... - << destination directory >>
Step 3: New a py file New File Or Click “New File” icon, and enter “isom3400_variable.py”
Step 4: Use #%% to separate sections of code #%% print("Hello World!")

Working with Virtual Environment

- **Mac User:** The following slides are for students who have updated terminal to **zsh** and **command not found conda** is shown

Working with Virtual Environment

- **Mac User:** Type **open ~/.bash_profile** in the terminal

A screenshot of a macOS terminal window. The title bar shows standard macOS window controls (red, yellow, green buttons) and a home icon followed by a black redaction box, then the text "— -zsh — 80x24". The terminal content shows the login message "Last login: Wed Aug 5 15:25:24 on ttys000". Below this, the prompt "[(base)]" is followed by a black redaction box. The next line shows the prompt "(base) " followed by the command "open ~/.bash_profile" and a cursor. The final line shows the prompt "(base) " followed by a black redaction box and the prompt "-Pro ~ %".

```
— -zsh — 80x24
Last login: Wed Aug 5 15:25:24 on ttys000
[(base) ]
(base)
(base) open ~/.bash_profile
(base) -Pro ~ %
```

Working with Virtual Environment

- **Mac User: Search & copy the following code** (started with **# added by Anaconda3** and ended with **#<<< conda init <<<**) and close the window

```
.bash_profile
PATH="/Library/Frameworks/Python.framework/Versions/2.7/bin:${PATH}"
export PATH

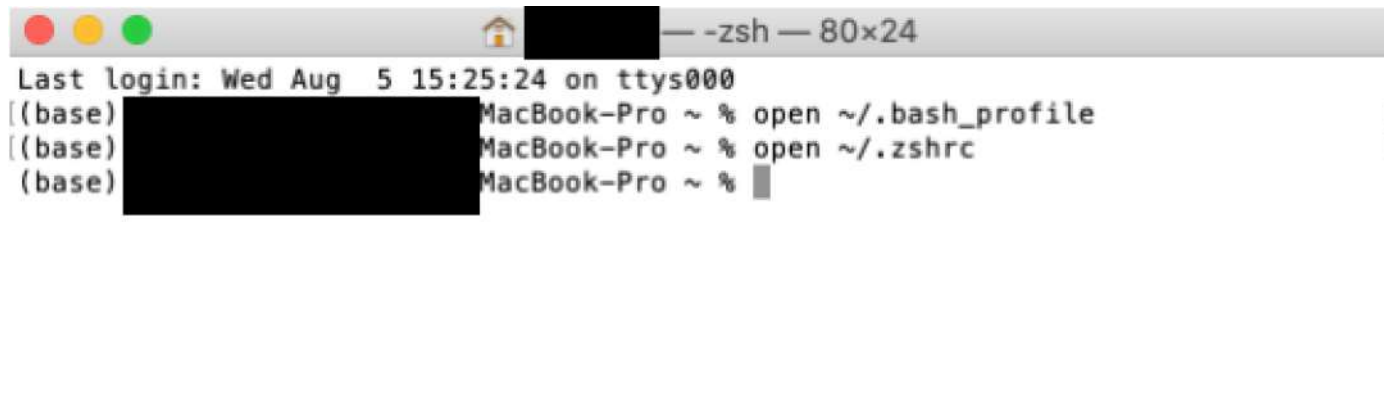
# Setting PATH for Python 2.7
# The original version is saved in .bash_profile.pysave
PATH="/Library/Frameworks/Python.framework/Versions/2.7/bin:${PATH}"
export PATH

# Setting PATH for Python 3.6
# The original version is saved in .bash_profile.pysave
PATH="/Library/Frameworks/Python.framework/Versions/3.6/bin:${PATH}"
export PATH

# added by Anaconda3 2018.12 installer
# >>> conda init >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$(CONDA_REPORT_ERRORS=false '/Users/████████/anaconda3/bin/conda' shell.bash
hook 2> /dev/null)"
if [ $? -eq 0 ]; then
    \eval "$__conda_setup"
else
    if [ -f "/Users/████████/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/Users/████████/anaconda3/etc/profile.d/conda.sh"
        CONDA_CHANGEPS1=false conda activate base
    else
        \export PATH="/Users/████████/anaconda3/bin:${PATH}"
    fi
fi
unset __conda_setup
# <<< conda init <<<
```

Working with Virtual Environment

- **Mac User:** Back to the terminal and type **open ~/.zshrc**

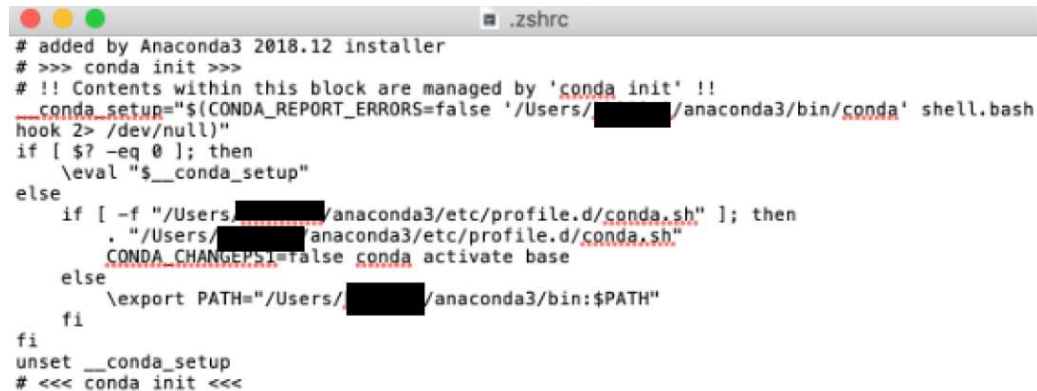


```

Last login: Wed Aug  5 15:25:24 on ttys000
[(base) MacBook-Pro ~ % open ~/.bash_profile
[(base) MacBook-Pro ~ % open ~/.zshrc
(base) MacBook-Pro ~ %
```

Working with Virtual Environment

- **Mac User: Paste** the code you copied and close the window

A screenshot of a text editor window titled ".zshrc" showing the Anaconda3 2018.12 installer's configuration code for the shell. The code is a bash script that sets up the conda environment. It includes comments and conditional logic to source the conda profile script and activate the base environment. The user's home directory is represented by a redacted black box.

```
# added by Anaconda3 2018.12 installer
# >>> conda init >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$(CONDA_REPORT_ERRORS=false '/Users/[REDACTED]/anaconda3/bin/conda' shell.bash
hook 2> /dev/null)"
if [ $? -eq 0 ]; then
    \eval "$__conda_setup"
else
    if [ -f "/Users/[REDACTED]/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/Users/[REDACTED]/anaconda3/etc/profile.d/conda.sh"
        CONDA_CHANGEPS1=false conda activate base
    else
        \export PATH="/Users/[REDACTED]/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda init <<<
```

Working with Virtual Environment

- **Mac User: Restart** terminal and you should be able to use conda command now

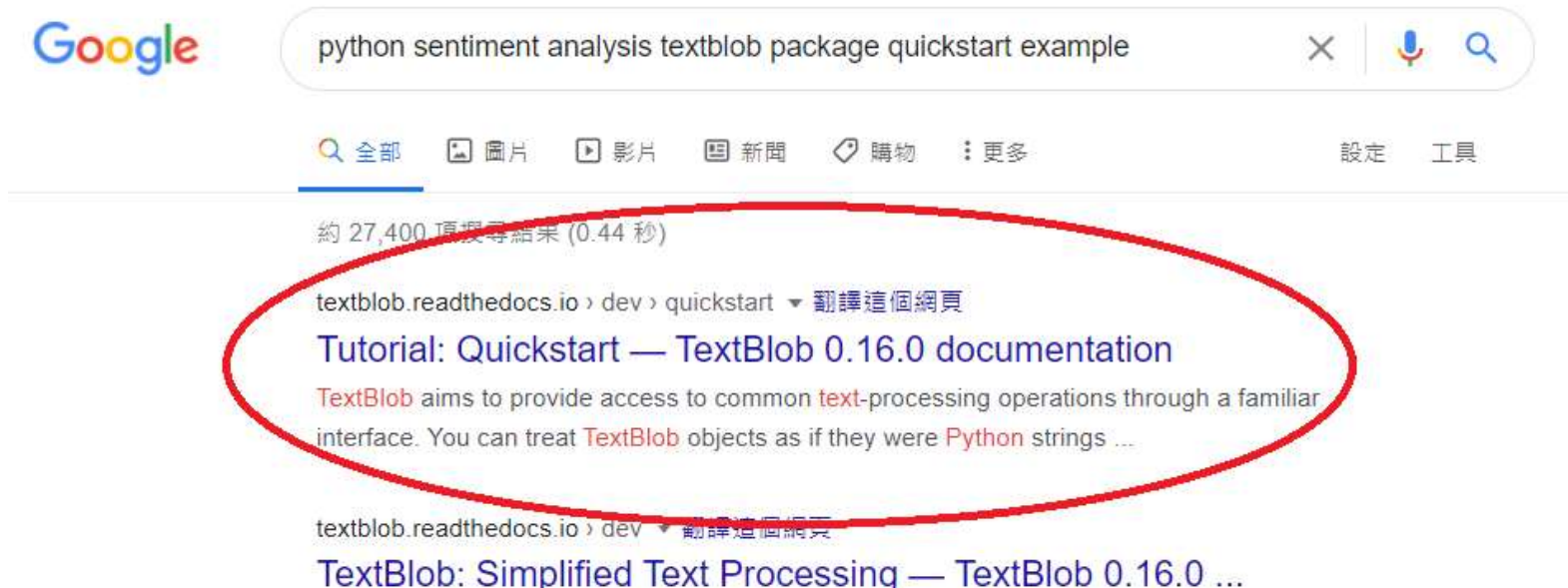
Python Program: Example

Python Program: Example - Objective

- Install a package
- Use functions in the package
- Test example given in the website

Python Program: Example

- Search **python sentiment analysis textblob package quickstart example** in Google
- Click the first one



Python Program: Example

- You need to use a package named **textblob** but it is not installed
- Click the link under tag **Useful Links**

TextBlob Star 7,530

TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more.

Useful Links

- [TextBlob @ PyPI](#)
- [TextBlob @ GitHub](#)
- [Issue Tracker](#)
- [Table of Contents](#)

Create a TextBlob

First, the import.

```
>>> from textblob import TextBlob
```

Let's create our first **TextBlob**.

```
>>> wiki = TextBlob("Python is a high-level, general-purpose programming language.")
```

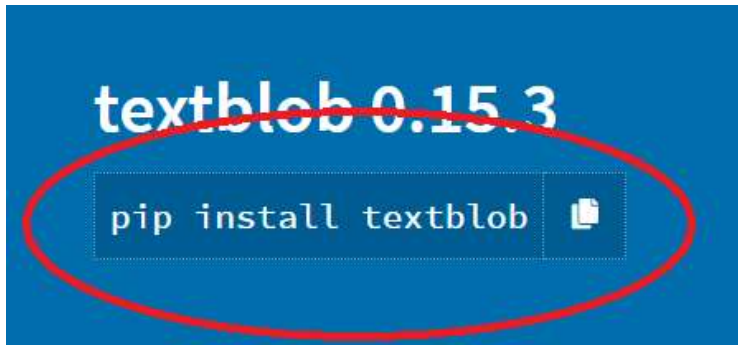
Part-of-speech Tagging

Part-of-speech tags can be accessed through the **tags** property.

```
>>> wiki.tags
[('Python', 'NNP'), ('is', 'VBZ'), ('a', 'DT'), ('high-level', 'JJ'), ('general-pu
```

Python Program: Example

- Install textblob to the environment



```
(labdemo) C:\Users\ >pip install textblob
Collecting textblob
  Using cached textblob-0.15.3-py2.py3-none-any.whl (636 kB)
Collecting nltk>=3.1
  Using cached nltk-3.5-py3-none-any.whl
Collecting click
  Using cached click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting joblib
  Using cached joblib-1.0.0-py3-none-any.whl (302 kB)
Collecting regex
  Using cached regex-2020.11.13-cp39-cp39-win_amd64.whl (270 kB)
Collecting tqdm
  Using cached tqdm-4.56.0-py2.py3-none-any.whl (72 kB)
Installing collected packages: tqdm, regex, joblib, click, nltk, textblob
Successfully installed click-7.1.2 joblib-1.0.0 nltk-3.5 regex-2020.11.13 textblob-0.15.3 tqdm-4.56.0
```

Python Program: Example

- Which parts to illustrate?



The screenshot shows the TextBlob documentation page. A red oval highlights the 'Create a TextBlob' section, which includes the following content:

Natural Language Processing.

Create a TextBlob

First, the import.

```
>>> from textblob import TextBlob
```

Let's create our first [TextBlob](#).

```
> wiki = TextBlob("Python is a high-level, general-purpose programming language.")
```

Below the highlighted section, the 'Part-of-speech Tagging' section is visible.

TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more.

Useful Links

Python Program: Example

- Which parts to illustrate?

amatization
rdNet Integration
rdLists
lling Correction
: Word and Noun Phrase
quences
sin
tJobs Are Like Python
ng!
rams
: Start and End Indices of
tences
Next Steps
ted Topics

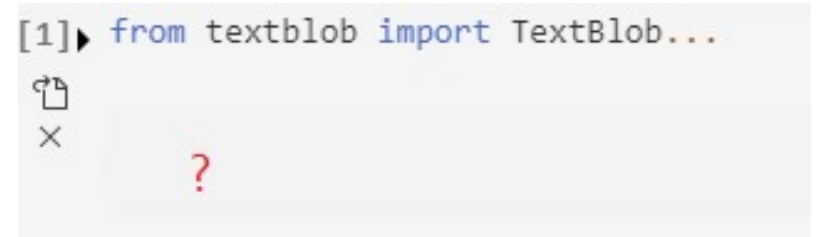
Sentiment Analysis

The [sentiment](#) property returns a namedtuple of the form `Sentiment(polarity, subjectivity)`. The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

```
>>> testimonial = TextBlob("Textblob is amazingly simple to use. What great fun!")  
>>> testimonial.sentiment  
Sentiment(polarity=0.39166666666666666, subjectivity=0.4357142857142857)  
>>> testimonial.sentiment.polarity  
0.39166666666666666
```

Python Program: Example

- Try the first one – Simple copy & paste
- **from textblob import TextBlob**
- **wiki = TextBlob("Python is a high-level, general purpose programming language.")**
- What happens in the interactive window?
- Can you tell the type of **wiki**, i.e. what is being created?



```
[1]: from textblob import TextBlob...
```

```
Run Cell | Run Above | Debug Cell | Go to [1]
#%%
from textblob import TextBlob

wiki = TextBlob("Python is a high-level, general-purpose programming language.")
```

Python Program: Example

- Try the second one
- **testimonial = TextBlob("Textblob is amazingly simple to use. What great fun!")**
- **testimonial.sentiment**

Run Cell | Run Above | Debug Cell | Go to [2]

#%%

```
testimonial = TextBlob("Textblob is amazingly simple to use. What great fun!")
```

```
testimonial.sentiment
```

```
[2] ▶ testimonial = TextBlob("Textblob is amazingly simple to use. What great fun!")...
```



?

Python Program: Example

- How about the last one?
- **testimonial.sentiment.polarity**
- Can you give a simple explanation on the codes being executed?

```
Run Cell | Run Above | Debug Cell | Go to [3]  
#%%  
testimonial.sentiment.polarity
```

```
[3] testimonial.sentiment.polarity
```



?

Take away

- Use of VSCode
- Virtual Environment: Setup and implement
- Python Program: Example

End