Python Programming

01. Getting Started

Evan Chang

About me - Evan Chang

- ~~NCTU~~ NYCU Computer Science Ph.D. Student
- Data Science / ML / DL
- Python / C / C++ / Java
- toosyou.tw@gmail.com

What's Python

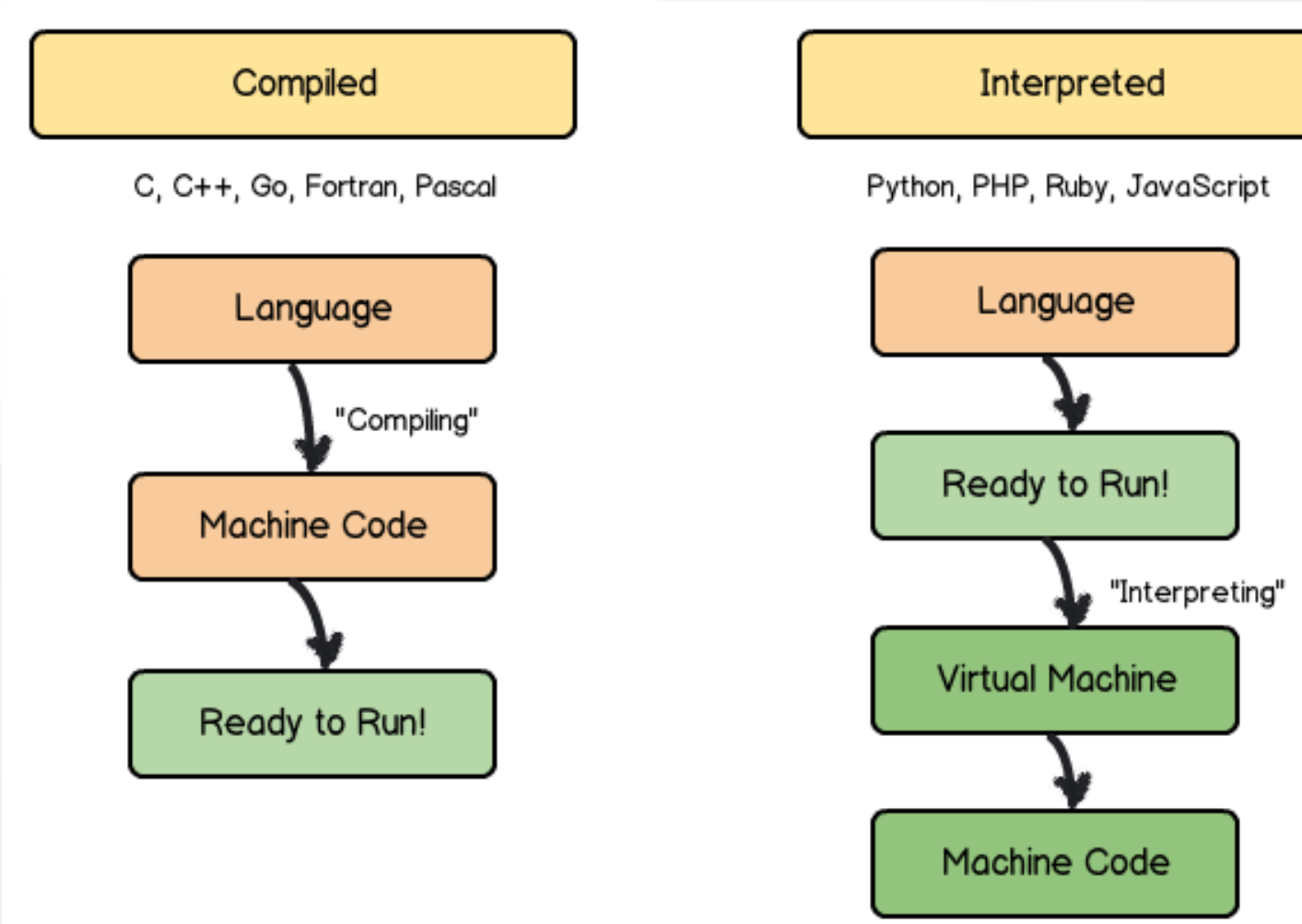
<- **Guido van Rossum** created it in *1989*.

Python 2.0 *2000 - 2020*
Python 3.x *2008 - now*

What's Python

- **Beautiful** is better than ugly.
- **Explicit** is better than implicit.
- **Simple** is better than complex.
- **Complex** is better than complicated.
- **Readability** counts.

What's Python



What's Python

- Interpreted Language 直譯語言
  - No **Compile** required.
- Dynamic Typed Language 動態語言
  - Type checking in run-time.
  - i.e. type error might occur while executing.

What's Python

- Easy to learn.
- Faster to develop.
- Cross-platform.
- **Free packages!**

Pop Quiz:

Python 2.0 是在哪一年發行的？

Syllabus

1. Environment, Variable, Operation
2. Conditional Statement
3. Loop & Iteration
4. Data Container
5. Functions
6. More Data Container
7. Team Match

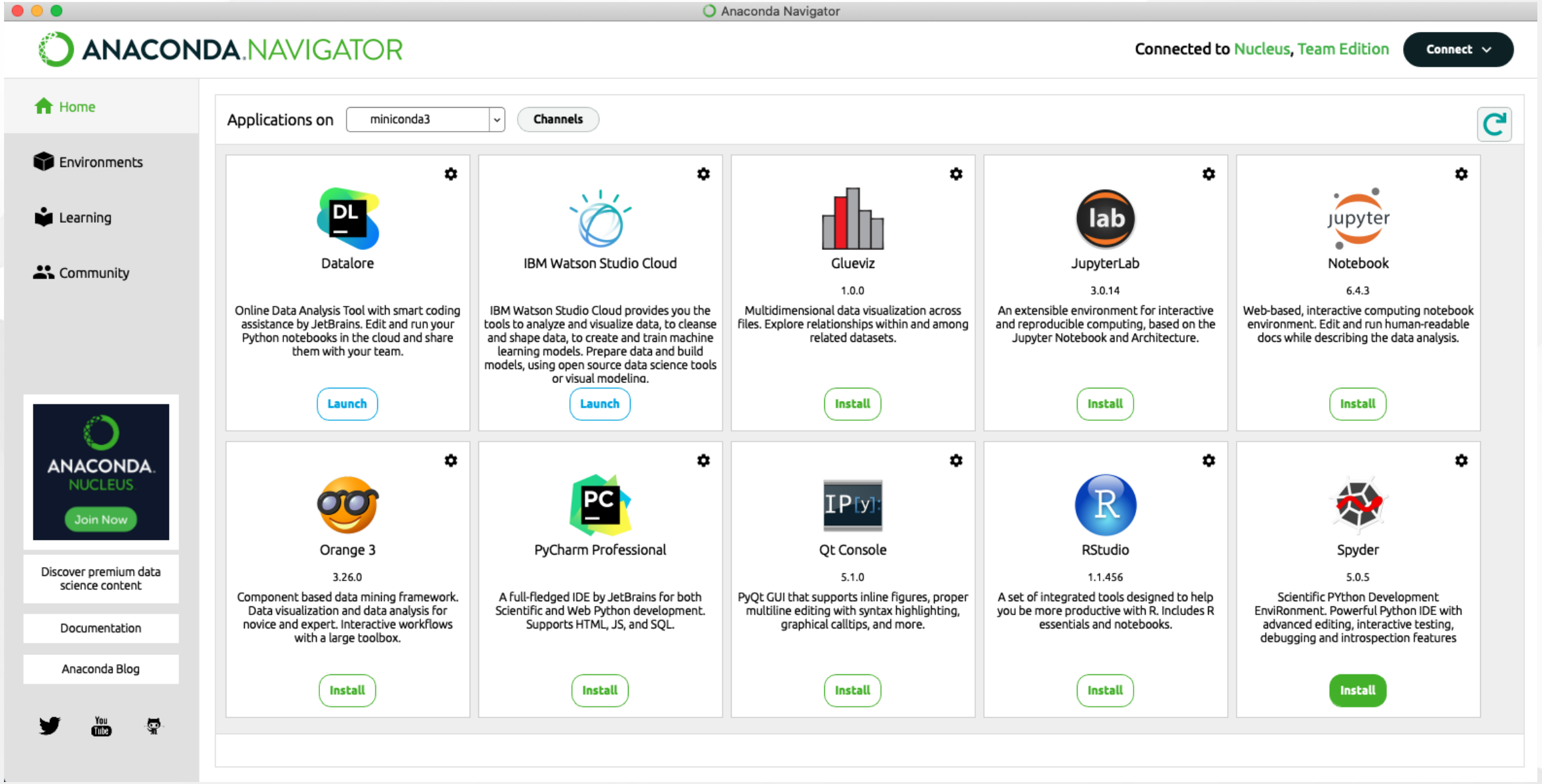
Scoring

- 6 HWs, one for each class.
  - 4 - 5 questions each.
  - Upload the source codes to NYCU E3.
  - TA will help.
- Team match
  - ~30 people a team
  - ~100 questions
  - Easy / Medium / Hard

python™

Installation

- **Anaconda**
  - Your data science toolkit
  - https://www.anaconda.com/products/individual
    - Python 3.9, 64-Bit Graphical Installer (4xx MB)
    -  > `Anaconda3 (64-bit)` >  `Anaconda Navigator`
    -  `Spyder`


Anaconda Navigator
ANACONDA.NAVIGATOR
Connected to Nucleus, Team Edition
Connect
Home
Environments
Learning
Community
Applications on
miniconda3
Channels
Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.
Launch
IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch
Glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install
JupyterLab
3.0.14
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Install
Notebook
6.4.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Install
ANACONDA. NUCLEUS
Join Now
Discover premium data science content
Documentation
Anaconda Blog
Orange 3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install
PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Install
Qt Console
5.1.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Install
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install
Spyder
5.0.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Install

C:/Users/TestUser/Documents/Spyder - Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\TestUser\Downloads\16740156420fb678bb5ba67c3ee3aae4-551407feca17b0f67bb9f85687f4db8d1b953678\16740156420fb678bb5ba67c3ee3aae4-551407feca17b0f67bb9f85687f4db8d1b953678

Project explorer

Spyder
Data
spyder
.github
conda.recipe
continuous_integration
doc
img_src
requirements
rope_profiling
scripts
spyder
app
tests
__init__.py
cli_options.py
mac_stylesheet.qss
mainwindow.py
restart.py
start.py
tour.py
config
defaults
fonts
images
locale
plugins
tests
utils
widgets
windows
workers
__init__.py
dependencies.py
interpreter.py
otherplugins.py
pil_patch.py
py3compat.py
pyplot.py
requirements.py
spyder_breakpoints
spyder_io_dcm
spyder_io_hdf5
spyder_profiler
spyder_pylint
.checkignore
.ciocheck
.ciocopyright
.codecov.yml
.coveragerc
.gitignore
.pep8speaks.yml
.project
.travis.yml
Announcements.md
appveyor.yml

Editor - C:\Users\TestUser\Downloads\16740156420fb678bb5ba67c3ee3aae4-551407feca17b0f67bb9f85687f4db8d1b953678\16740156420fb6...

temp.py | interpolation.py | __init__.py | umd_helper.py | umd_main.py | README.md

```
6
7  import pylab
8  from numpy import cos, linspace, pi, sin, random
9  from scipy.interpolate import splprep, splev
10
11 # %% Generate data for analysis
12
13 # Make ascending spiral in 3-space
14 t = linspace(0, 1.75 * 2 * pi, 100)
15
16 x = sin(t)
17 y = cos(t)
18 z = t
19
20 # Add noise
21 x += random.normal(scale=0.1, size=x.shape)
22 y += random.normal(scale=0.1, size=y.shape)
23 z += random.normal(scale=0.1, size=z.shape)
24
25
26 # %% Perform calculations
27
28 # Spline parameters
29 smoothness = 3.0  # Smoothness parameter
30 k_param = 2  # Spline order
31 nests = -1  # Estimate of number of knots needed (-1 = maximal)
32
33 # Find the knot points
34 knot_points, u = splprep([x, y, z], s=smoothness, k=k_param, nests=-1)
35
36 # Evaluate spline, including interpolated points
37 xnew, ynew, znew = splev(linspace(0, 1, 400), knot_points)
38
39
40 # %% Plot results
41
42 # TODO: Rewrite to avoid code smell
43 pylab.subplot(2, 2, 1)
44 data, = pylab.plot(x, y, 'bo-', label='Data with X-Y Cross Section')
45 fit, = pylab.plot(xnew, ynew, 'r-', label='Fit with X-Y Cross Section')
46 pylab.legend()
47 pylab.xlabel('x')
48 pylab.ylabel('y')
49
50 pylab.subplot(2, 2, 2)
51 data, = pylab.plot(x, z, 'bo-', label='Data with X-Z Cross Section')
52 fit, = pylab.plot(xnew, znew, 'r-', label='Fit with X-Z Cross Section')
53 pylab.legend()
54 pylab.xlabel('x')
```

Outline

interpolation.py
Generate data for analysis
Perform calculations
Plot results
imputerNan
Queue
__init__
appendleft
pop
with open(data_path + output_file_n...
with open(data_path + output_file_n...
with open(data_path + output_file_n...
print_file
Example Bitzer class
DataSet
__init__
prepare_dataset
Serie
something
Dataframe
something
foo
__init__
spam
with open(file) as f:
Base
Derived
for r, bar in zip(radii, bars):
with np.load(filename) as dem:

Variable explorer

Name	Type	Size	Value
array_int8	int8	(2, 3)	Min: -7 Max: 6
array_uint32	uint32	(2, 2, 3)	Min: 1 Max: 7
bars	container.BarContainer	20	BarContainer object of matplotlib.conta…
df	DataFrame	(3, 2)	Column names: bools, ints
filename	str	1	C:\ProgramData\Anaconda3\lib\site-packa…
list_test	list	2	[Dataframe, Numpy array]
nrows	int	1	344
r	float64	1	7.611082589334796
radii	float64	(20,)	Min: 0.4983036638535687 Max: 9.856848974942551
region	tuple	2	(slice, slice)
rgb	float64	(45, 45, 4)	Min: 0.0 Max: 1.0
series	Series	(1,)	Series object of pandas.core.series mod…
test_none	NoneType	1	NoneType object of builtins module

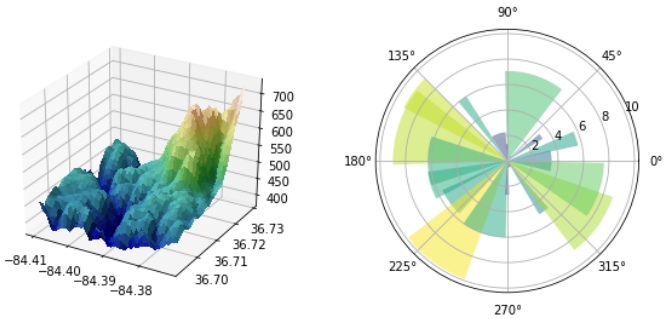
Variable explorer | Help | File explorer | Find in files | Breakpoints | Static code analysis | Profiler | Online help

IPython console

Console 1/A | Console 2/A | Custom Name/A

00:34:13

```
...:
...: ls = LightSource(270, 45)
...: # To use a custom hillshading mode, override the built-in shading
...: # in the rgb colors of the shaded surface calculated from "shade".
...: rgb = ls.shade(z, cmap=cm.gist_earth, vert_exag=0.1, blend_mode='soft')
...: surf = ax.plot_surface(x, y, z, rstride=1, cstride=1, facecolors=rgb,
...:                        linewidth=0, antialiased=False, shade=False)
...:
...: plt.show()
```



In [12]:

IPython console | History log | Internal console

Permissions: RW | End-of-lines: LF | Encoding: UTF-8 | Line: 26 | Column: 4 | Memory: 49 % | CPU: 15 %

Spyder

- **IDE** for python
  - **I**ntegrated **D**evelopment **E**nvironment
    - 整合開發環境
  - Where to edit and run codes

Hello, World!

1.  New File
2. `print('Hello, World!')`
3. Ctrl+s to save the program (you could name it `hello_world.py`)
4. Run 
5. See the output in the bottom right panel!
6. ~~Now you know python~~

How not to write code - comments

- Single-line comments start with `#`
- Multiple lines comments start and end with `'''` or `"""`
- Computer won't see comments. Write whatever you want!

```
# This is a single line comment
'''This is a
    multiple lines comment
'''
""" so is this one """
```

Constants

- `'Hello, World!'` is a `str`, string 字串 **constant**
  - `'str'`, `"str"`, `'''str'''` and `"""str"""` are all the same in python
  - Try it!

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

Constants

- But how to print single/double quote?
  - `print("single quote:'")` and `print('double quote:"')` works
  - `print('\'')` and `print("\"")` also works
- Special Characters
  - Tab: `\t`
  - Newline: `\n`
  - Beep: `\a`

Constants

- Numerical **constants**
  - `int`, Integer 整數 constant: `420`
  - `float`, Float 浮點數 constant: `199.87`
  - `7e10` means 7×10^{10}
  - Try it!

```
print(123)
print(321.1234567)
print(9.99e9)
```

Arithmetic

- `+ - * /`: `print(1 + 2 / 3)` -> `1.6666666666666665`
- `//` integer division 整數除法: `print(10 // 3)` -> `3`
- `%` modulus 取餘數: `print(10 % 3)` -> `1`
- `**` power: `print(2 ** 10)` -> `1024`

Operator Precedence Rule

1. Parenthesis `( )`
2. Power `**`
3. Multiplication, Division, Modulus `*/%`
4. Addition & Subtraction `+-`
5. Left to right

Pop Quiz:

```
x = 3 + 4 ** 2 / 4 % ( 2 + 1 )
```

Variables

- Object with a name that stores data
  - You could choose the name, ~~choose wisely.~~
  - Variables can be reassigned too.
  - Use single `=` to assgin values to variables.

```
a = 'Hello,'
b = 69
c = 10
c = b / c
print(a, b, c) # Hello, 69 6.9
```

    - what's the value of `c`?

Variable Names

- Must start with a letter or underscore `_`
- Only letters, numbers and underscore `_`
  - ❌: `2people`, `#sign`, `varable.123`
- Case sensitive 大小寫有差
  - `Sign`, `sign` and `SiGn` are all different variables
- No python keywords

Python Keywords

```
False   def     if      raise
None    del     import  return
True    elif    in      try
and     else    is      while
as      except  lambda  with
assert  finally nonlocal        yield
break   for     not
class   from    or
continue        global  pass
```

`print`

- Print multiple things at once.

```
a = 420
b = 10
c = a / b
print(a, 'divided by', b, '=', c) # 420 divided by 10 = 42.0
```

  - `a`, `'divided by'`, `b`, `'='` and `c` are arguments 引數 for `print`
  - Note that they are separated by a space.

`print` arguments

- Pass `sep` argument to use different separator.

```
print(10, 20, 30, sep=', ') # 10, 20, 30
print(10, 20, 30, sep='\t') # 10	20	30
```

- `end` argument to change the ending string.

```
print(10, 20, 30, end='')
print(40, 50, 60)
# 10 20 3040 50 60
# why?
```

`input(prompt_str)`

- Ask the user to input something to the console
  - End with an `Enter`

```
name = input('Tell me your name:')
age = input('And your age:')
print("You're", name, "(", age, ")") # You're Evan ( 18 )
```

  - However

```
print(age + 1) # TypeError: can only concatenate str (not "int") to str
```

    - Why?

Type Conversion

- `type` will return what type the variable is

```
print(type(age)) # <class 'str'>
```

  - Since `input` return `str`
- Use `int()` to convert object to Integer

```
age = int(age) # convert to Integer
print(age + 1, type(age)) # 19 <class 'int'>
```

Type Conversion

- Convert it right after `input`

```
age = int(input('Tell me your age:'))
print(age + 1)
```

  - ⚠️ Conversion might fail if decimals or texts are inputted.
- Convert to float -> `float()`
- Convert to string -> `str()`
- You get the idea.

Type Conversion

- Integer conversion won't round up
  - `int(3.999)` -> `3`
  - `int(-3.999)` -> `-3`
- `str` convert numbers to strings of number
  - `str(-42)` -> `'-42'`
  - `str(3.999)` -> `'3.999'`

Arithmetic Assignment Operators

- Combine arithmetic operations with assignments
  - `a = a + b` -> `a += b`
  - `a = a - b` -> `a -= b`
  - `a = a * b` -> `a *= b`
  - `a = a / b` -> `a /= b`
  - `a = a % b` -> `a %= b`
  - `a = a ** b` -> `a **= b`
  - `a = a // b` -> `a //= b`

Pop Quiz:

「12345679」猜四個字

Arithmetic Assignment Operators

- Combine arithmetic operations with assignments
  - ⚠️ The followings are not valid syntax.

```
a += a += 1
a += (b += 1)
```

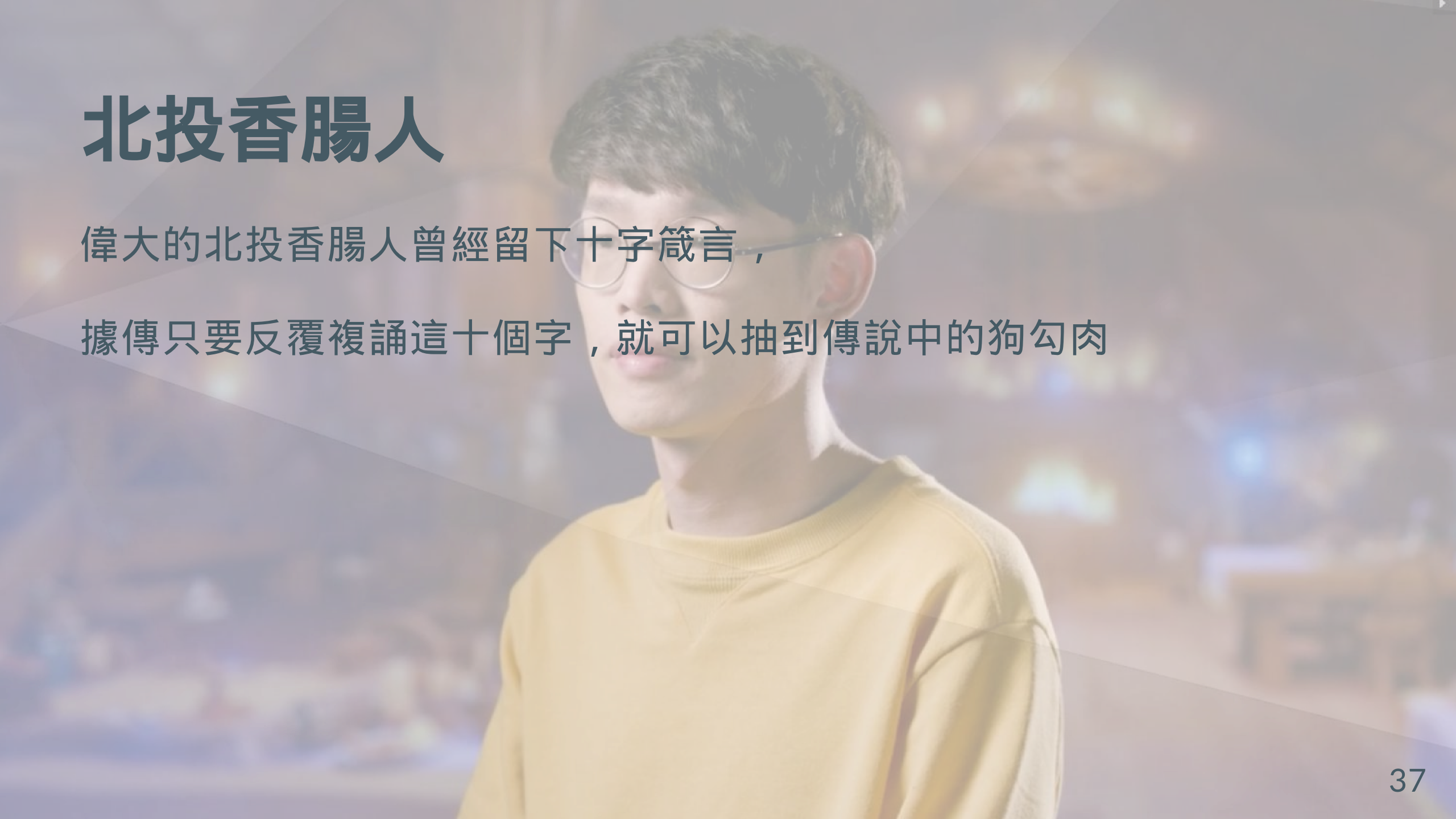
Exercises

北投香腸人

北投香腸人

偉大的北投香腸人曾經留下十字箴言，

據傳只要反覆複誦這十個字，就可以抽到傳說中的狗勾肉



1. 亂講

- 今天你覺得老師上課在亂講，所以你要印一個「亂講」
- 打開 Spyder ，使用 `print` 印出 `亂講`
  - 中文會壞掉的話可以用英文

2. 冷靜

- 因為你反駁老師，老師太激動了，
- 請輸入老師的名字，後面要他「冷靜」

1. 使用 `input` 輸入老師名字，存入變數 `teeechearrrr`
2. 使用 `print` 印出 `{teeechearrrr} 冷靜`
    - e.g. `TOZY 冷靜`

3. 真假

- 你同學跟你講溫度都用華氏，你只好回「真假」再用電腦轉成攝氏度
- 寫一個程式將輸入的華氏溫度 F 轉換成攝氏度 C 印出來
  - $C = \frac{5}{9} \times (F - 32)$

```
輸入：50
輸出：真假 10 度喔
```

4. 有料

- 北投香腸人的朋友 TOZY 最近想買狗牌電動機車
- 但他只有四種硬幣（50, 10, 5, 1）可以付款
- 幫他用最少的硬幣付款吧
- 請輸入一個整數的價格，印出最有料的付法

```
輸入：12699
輸出：50 * 253
      10 * 4
       5 * 1
       1 * 4
```

5. 中計

- 北投香腸人買了一片三角形批薩，發現面積好像怪怪的
- 輸入三個邊長，請用海龍公式印出面積

$$\triangle = \sqrt{s(s-a)(s-b)(s-c)}, s = \frac{a+b+c}{2}$$

```
輸入　：3↵　4↵　5↵
輸出：6　阿中計
```

- 提示

Acknowledgment

- Prof. Chang-Chieh Cheng. National Yang Ming Chiao Tung University, Taiwan
- Python for Everybody