

# Python 課程資訊

## 大綱

1. 講師介紹
2. Python 介紹
3. 環境安裝
4. 基本語法、變數與資料型態
5. 運算子
6. 條件式
7. 迴圈
8. 函式
9. 物件導向程式
10. 常用基本模組
11. 檔案與例外處理
12. 實際應用-網路爬蟲
13. 實際應用-資料科學計算
14. 實際應用-GUI程式(+Parallel Programming)

## 講師介紹

- 本名: 趙俊凱 (Compeador)
- 海洋大學資訊工程學系四年級
- YTP 3!
- 核心智慧資安發展處實習生
- 海大網路發展協會(IND)社長

## Python 介紹

### Why Python?

- 完全看需求與喜好
- 語言特性?
- 套件完整度?
- 社群?

### Python 的功能

- General-Purpose
- 解譯式語言 (Interpreted language)

- 可被設計為廣知的程式設計範式 (programming paradigm)
  - 程序式 (Procedural)
  - 物件導向 (Object-Oriented)
  - 函式化 (Functional)
- 語法簡潔, 動態資料形態

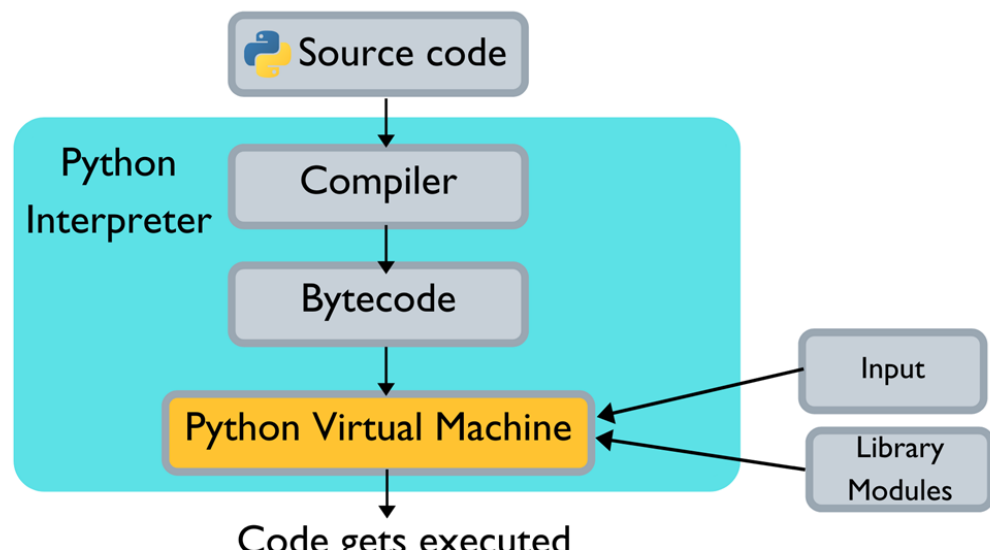
## 編譯 vs 解譯

### 編譯

- 將程式轉換為可執行的二進位檔 (binary executables)
- 執行速度快
- 需等編譯完畢後才能看到結果
- 撰寫時資料通常為固定形別

### 解譯

- 解譯器逐行執行, 可立即看到結果
- 執行速度較慢
  - 其實還是要把code變成機器碼
- 撰寫時資料不需明確定義型別



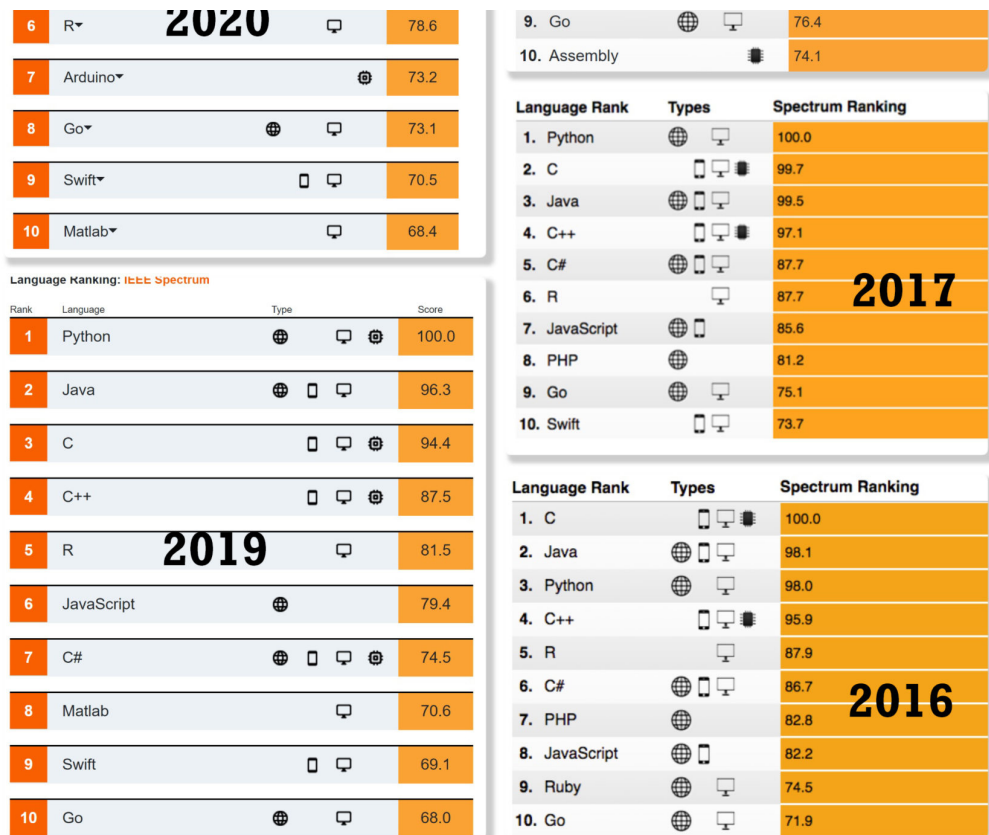
## Top (popular) programming languages by IEEE

Rank	Language	Type	Score
1	Python	☉ 📱 🖥️	100.0
2	Java	☉ 📱 🖥️	95.3
3	C	📱 🖥️ ☉	94.6
4	C++	📱 🖥️ ☉	87.0
5	JavaScript	☉	79.5

Language Rank	Types	Spectrum Ranking
1. Python	☉ 📱 🖥️	100.0
2. C++	📱 🖥️ ☉	99.7
3. Java	☉ 📱 🖥️	97.5
4. C	📱 🖥️ ☉	96.7
5. C#	☉ 📱 🖥️	89.4
6. PHP	☉	84.9
7. R	🖥️	82.9
8. JavaScript	☉ 📱	82.6

2018



Source: [spectrum.ieee.org/static/interactive-the-top-programming-languages-2020](https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2020)

## Python 2?

已經停止支援，除非有歷史包袱，否則一律建議使用 Python3

## 安裝 Python

### 安裝虛擬環境 Anaconda

請到 <https://www.anaconda.com/products/individual#Downloads> 下載

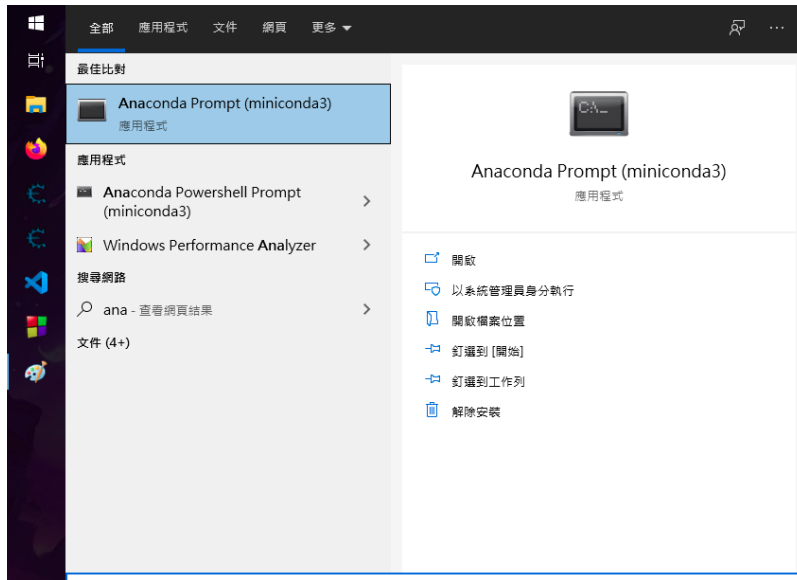
- Anaconda
  - 包含大量常用套件,吃空間
  - 許多函式庫與相容問題不需再自己安裝解決
- Miniconda
  - 僅有核心套件,乾淨
  - 需自行安裝套件,若有版本衝突需自行解決

### 若直接在 Windows 原生環境安裝 Python

- 若有先前安裝的軟體需要 Python (或後續安裝), 可能因衝突卡套件、函式庫版本抓錯 (主要是Python 2/3 衝突)
- 如果都是用 Python 3 的套件應該不會有問題

- Linux 沒有這個問題 :D

## 開啟 Conda shell



## 套件管理工具

在 Conda 虛擬環境內兩者都有提供，一般來說端看個人習慣

- Pip
  - `pip install <package-name>`
  - 套件庫: PyPi
- Conda
  - `conda install -c anaconda <package-name>`
  - 套件庫: Anaconda/packages

## Python IDE

### Jupyter Notebook

- 程式碼可分好幾段, 可結合文字與圖片敘述 (Markdown)
- 適合規模較小的程式
- 安裝: `pip install notebook` 或 `conda install -c conda-forge notebook`
- 執行: `jupyter notebook`

### Visual Studio Code

- 適合任何規模
- 擴充功能多
- 安裝: <https://code.visualstudio.com/>

## PyCharm (JetBrain)

- 專門開發 Python 的正式 IDE, 適合任何規模
- 內建功能多
- Professional (Education) / Community
- 安裝: <https://www.jetbrains.com/pycharm/>

## 建立虛擬環境

針對不同的Python模組或專案，可建立各自虛擬環境，避免因版本問題互相衝突。

### 建立新的虛擬環境

- `conda create -n myenv`
- `conda create -n myenv python=3.8.5`
- `conda create -n myenv python=3.8.5 pip`

### 切換/停用虛擬環境

- `conda activate myenv`
- `conda deactivate myenv`

### 更改名稱

先複製新名稱後刪掉舊的

- `conda create -n new_name --clone old_name`
- `conda remove -n old_name --all`

## 確認 Python 版本

- `python --version`

In [1]:

```
import sys
print(sys.version_info)
print(sys.version)
```

```
sys.version_info(major=3, minor=8, micro=5, releaselevel='final', serial=0)
3.8.5 (default, Sep  4 2020, 07:30:14)
[GCC 7.3.0]
```

Python 語言的特性可以很快上手使用程式表達概念。但是還是要花時間才能精熟。

**恭喜! 基本環境已經設定完成!**