

# 110年第2學期 智慧計算實驗

Computational Intelligence Laboratory

人工智慧

物聯網

電腦視覺

網路通訊

東華大學電機工程學系





課程大綱

# COURSE OUTLINE

1

智慧計算(Computational Intelligence)

2

人工智慧(Artificial Intelligence, AI) 理論與實作

3

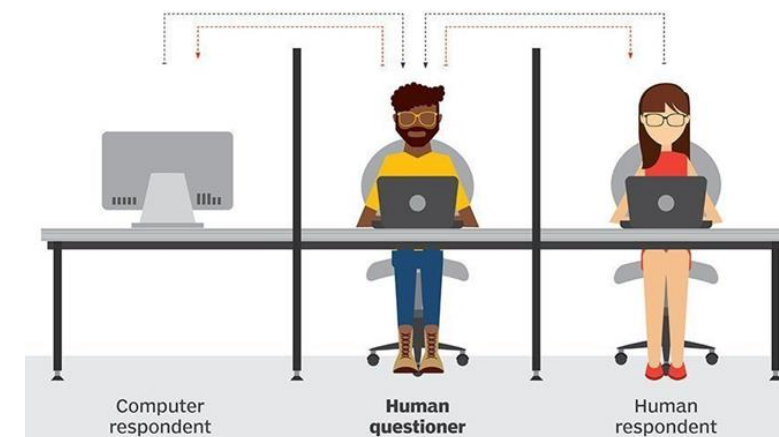
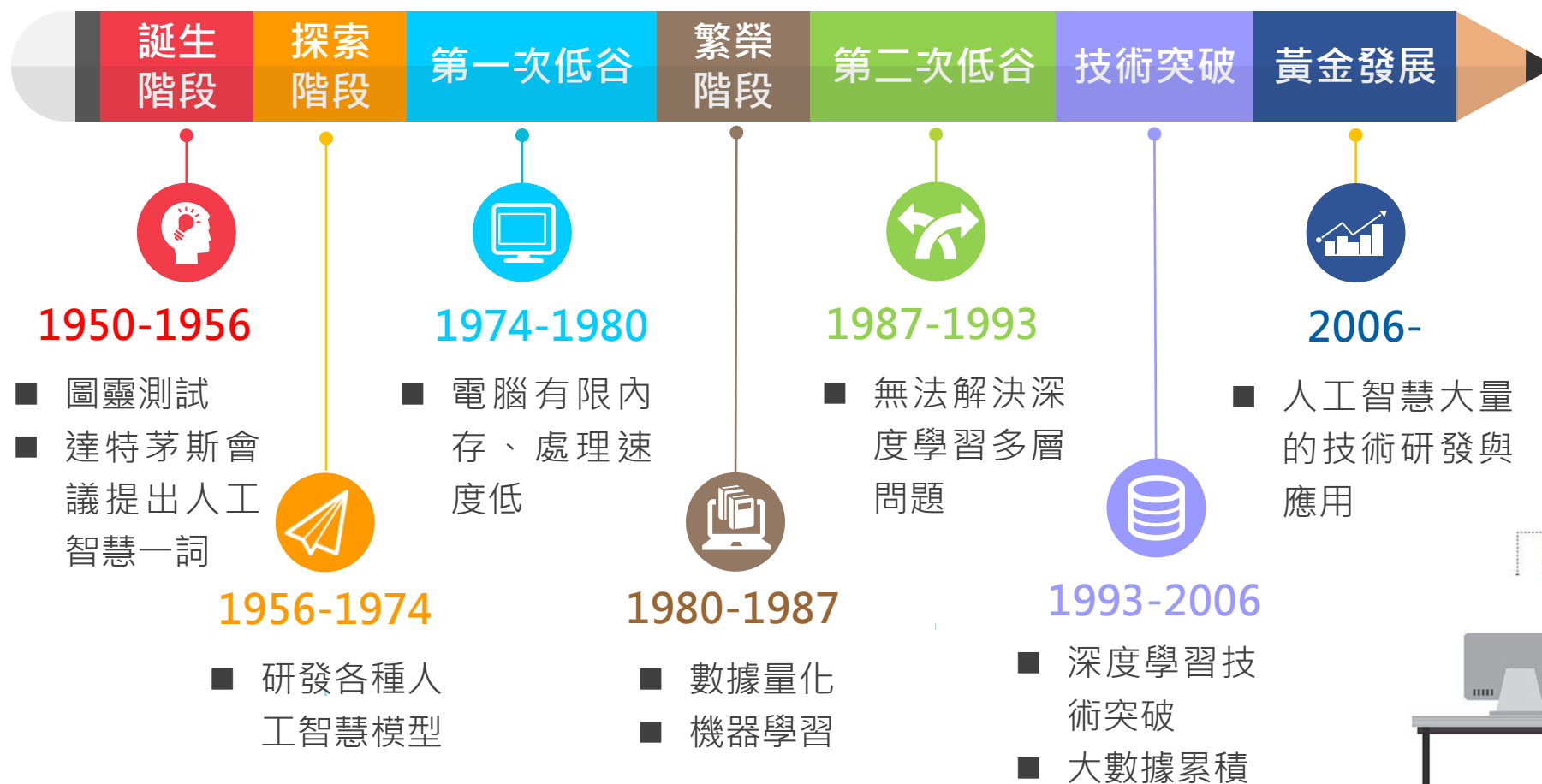
人工智慧在電腦視覺、網絡通訊與物聯網上的最新應用

註

應先修課程：程式設計(一)

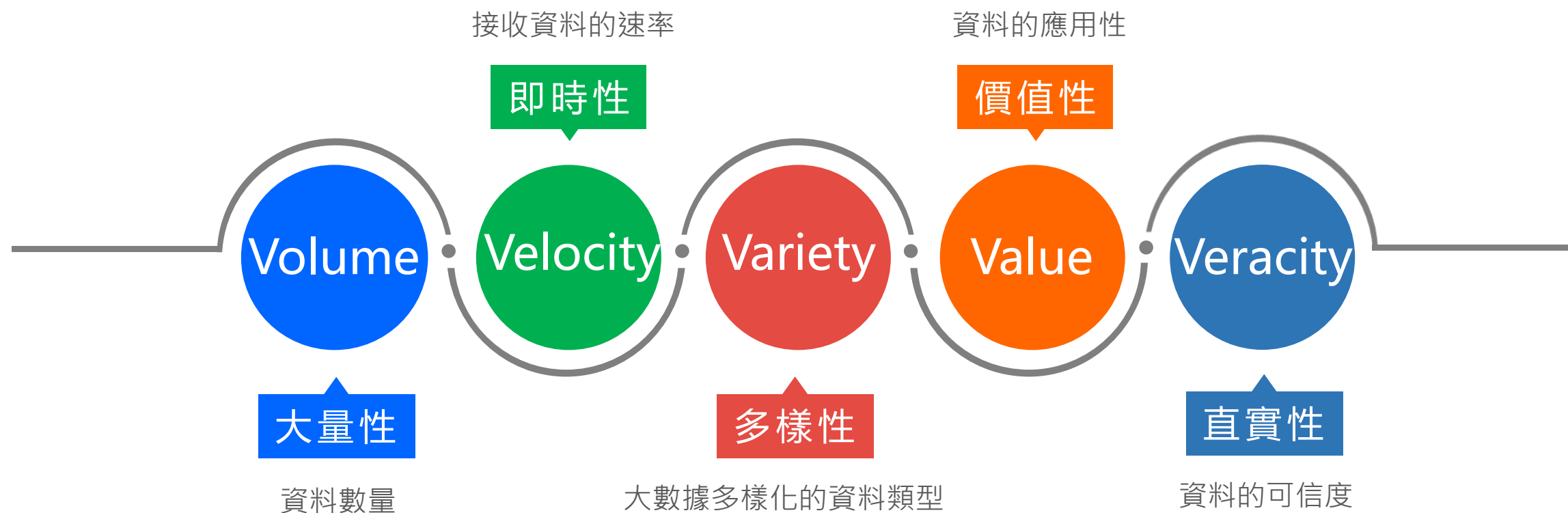


# 01 發展歷程



## 02 大數據

指更龐大且更複雜的資料集，使傳統的資料處理軟體已無力招架



# 03 學習任務

機器學習：透過大量的資料來讓機器(模型)，學習到如何正確預測及判斷

在數學定義上是找到一函數，能讓資料輸入函數得到輸出值，其值與實際值越相近

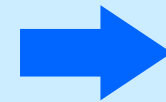
輸入

函數

輸出

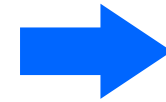
回歸

PM<sub>2.5</sub>、CO、SO<sub>2</sub>等



呼吸道感染人數

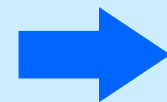
分類



貓 (0) / 狗 (1)

類別

分類  
定位



影像的像素值

目標物的類別及其位置

PART

1

# 人工智慧安裝篇

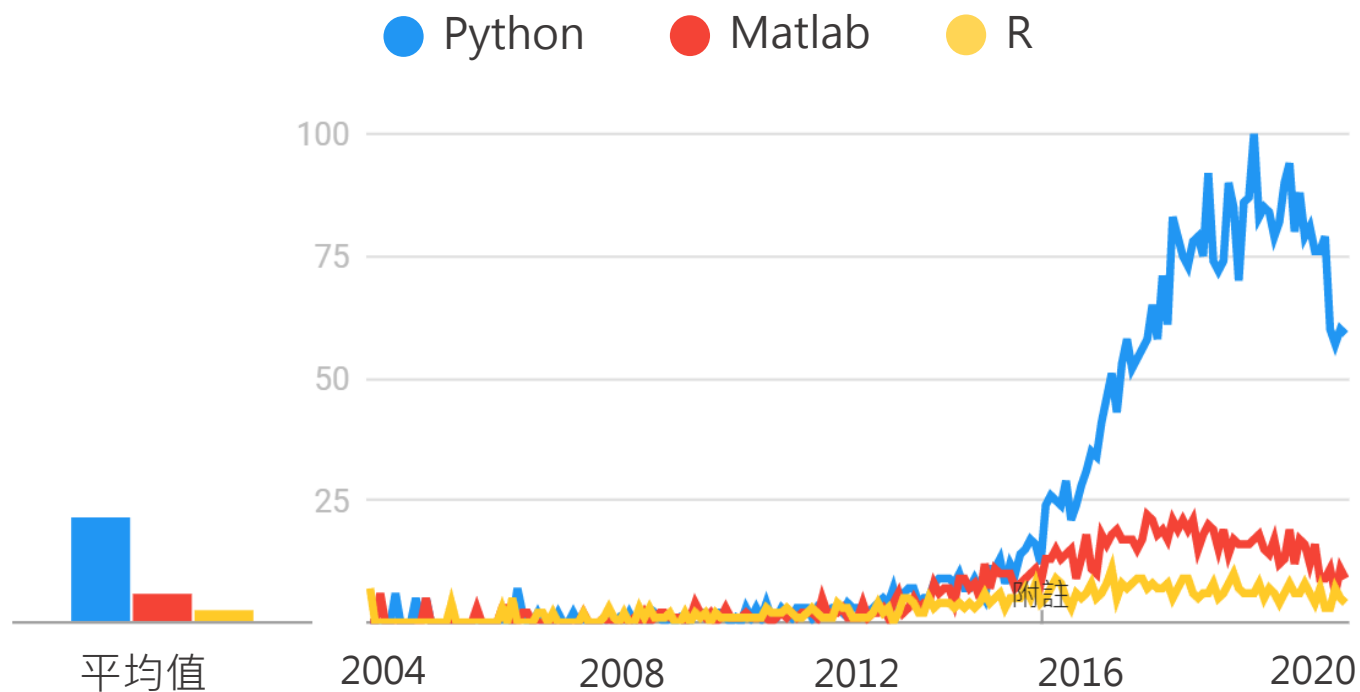
- 編程語言
- 深度學習框架
- 本機安裝Python
- Jupyter Notebook 自動補上語法
- 安裝套件



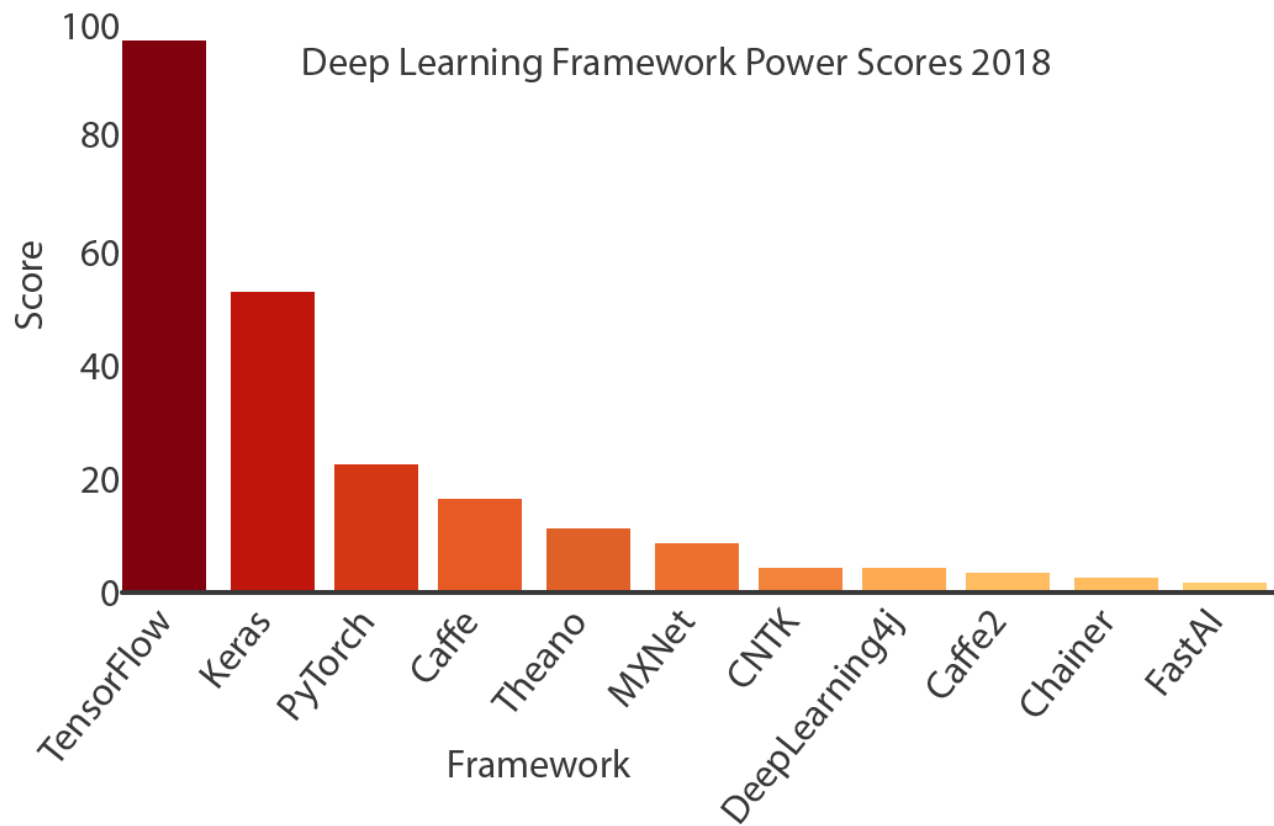
# 01 編程語言

## 01 人工智慧安裝篇

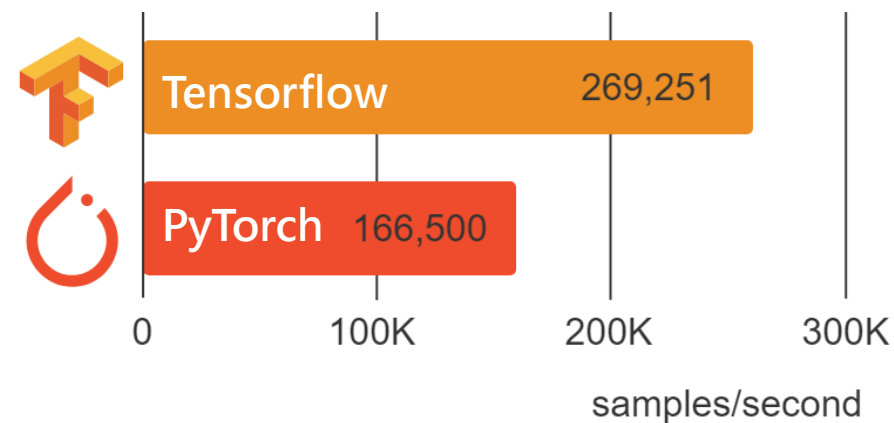
Google Trends 全球搜索詞彙：Machine and Deep Learning



## 02 深度學習框架



## 01 人工智慧安裝篇



	Tensorflow	PyTorch
Training time	01:01:54	01:40:06
Epoch time (seconds/epoch)	3.714	6.006
Training speed (samples/second)	269,251	166,500



## 03 本機安裝Python

- 從Anaconda官網中選擇下載電腦系統支援的安裝檔案(目前Anaconda的最高Python版本為3.9)  
<https://www.anaconda.com/products/individual>
- 執行安裝檔，環境變數需要打勾文字成紅色，其餘都按下一步即可
- 安裝完成後，將Anaconda以右鍵以「系統管理身分執行」
- 由於考量本教材使用的套件大多只支援**Python3.6**，所以創建Python 3.6的環境：
  1. Environments中在base(root)根環境，以左鍵點擊三角形選擇執行「Open Terminal」執行命令提示字元(CMD)
  2. 在CMD中輸入 `conda create -n py36 python=3.6 anaconda`，開始創建名稱為py36與Python版本為3.6的環境
  3. 安裝完後，在Environments中點擊py36的環境，並轉換到「Home」，點擊安裝程式編譯器Jupyter Notebook「install」
  4. 完成後Environments中在py36環境，以左鍵點擊三角形選擇執行「Open with Jupyter Notebook」可以開始撰寫程式



## 04 Jupyter Notebook 自動補上語法

在 py36 環境中開啟命令提示字元執行下列指令：

### ■ 安裝 nbextensions

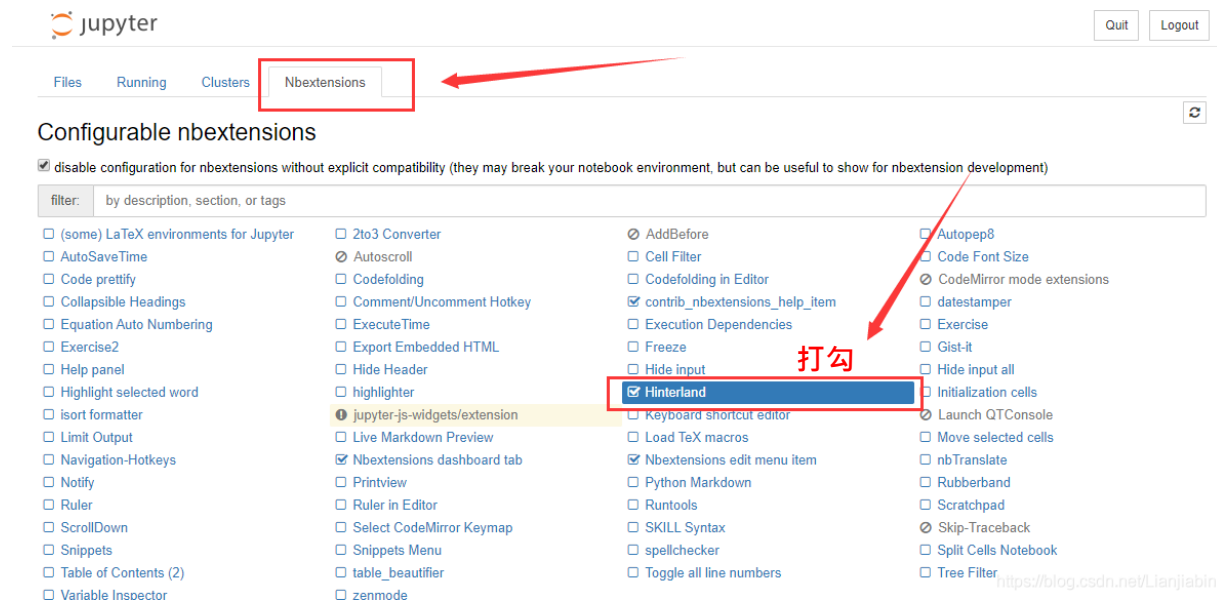
- ✓ `pip install jupyter_contrib_nbextensions`
- ✓ `jupyter contrib nbextension install --user`

### ■ 安裝 nbextensions\_configurator

- ✓ `pip install jupyter_nbextensions_configurator`
- ✓ `jupyter nbextensions_configurator enable --user`

■ 完成上述安裝後，請重新啟動 Jupyter Notebook，在彈出的Home裡面中，能看到增加了 Nbextensions 標籤頁，在這勾選Hinterland即啟動語法自動補全

- 補充：若要程式碼的左邊顯示行數，點擊「View」中的「Toggle Line Number(切換行數)」，預設為關閉



## 05 安裝套件

兩種安裝方式：

1. 可至PyPI網站（ <https://pypi.org/> ）上尋找套件名稱，並複製 `pip install package_name` 輸入至該環境的命令提示字元中



- 補充：pip 默認將 Python 包安裝到系統目錄，有時會需要系統管理訪問權限。  
添加 `-user` 指令是不需要任何特殊權限，`pip install package_name --user`
- 2. 在該環境中選擇「Not Installed」，對預安裝的套件點擊方框，並選擇「Apply」來安裝。或是使用 `conda install package_name`
- 補充：安裝方式 pip 與 conda 在環境中的依賴方面有所不同
  1. pip安裝時會在遞歸的串行循環中安裝依賴項。可能無法確保同時滿足所有套件的依賴性。如果較早安裝的套件與稍後安裝的套件具有不兼容的依賴性版本，則可能導致破壞環境
  2. conda使用可確保滿足環境中安裝的所有套件的所有要求。此檢查可能需要額外的時間，但有助於防止創建破壞環境

PART

2

# 資料處理

- 純文字文件
- 常見傳遞資料格式
- 影像
- 影片與攝影機串流



# 01 純文字文件

## ☑ 步驟：開啟 — 寫入/讀取 — 關閉

1. 開啟 `file_obj = open(file, mode= "r" )`

r 開啟檔案只供讀取，為預設值

w 開啟檔案供寫入，如果原先檔案有內容，其內容將被覆蓋

a 開啟檔案供寫入，如果原先檔案有內容，新寫入的資料將附加在後面

x 開啟一個新的檔案供寫入，如果所開啟的檔案已經存在則會產生錯誤

2. 讀檔 `file_obj.readlines()` 一次讀一行，以列表呈現

`file_obj.read()` 一次讀全部

3. 寫檔 `file_obj.write(str)`

`print(str, file=file_obj)`

4. 關閉 `file_obj.close()` 每次開啟檔案，請必要執行關閉

若使用with as 開啟檔案，以下程式執行完會自動關閉 `with open(file, mode= "r" ) as file_obj:`

## 02 常見傳遞資料格式

✓ Pandas 是一個資料處理與資料分析常用的開源套件 ( <https://pandas.pydata.org/docs/index.html> )

1. 給予資料與欄位字串，建立一DataFrame

```
df = pd.DataFrame(data, columns = [str])
```

2. 呈現DataFrame資料：前n筆資料(n預設為5)

```
df.head(n=5)
```

後n筆資料(n預設為5)

```
df.tail(n=5)
```

3. 從DataFrame中取得一欄位的資料：

```
data_column = df[str]
```

4. 儲存DataFrame資料：

- csv `df.to_csv(path, index = bool)`
- excel `df.to_excel(path, index = bool, sheet_name = str)`
- json `df.to_json(path)`
- xml `df.to_xml(path)`

index為第一欄位編號是否存取

sheet\_name 為工作表的名稱

備註：pandas 版本為 1.3.0 以上才能使用

## 02 常見傳遞資料格式

### 5. 讀取資料：

■ csv `df = pd.read_csv(path)`

■ excel `df = pd.read_excel(path, engine='openpyxl')`

openpyxl 支持較新的試算表格式

■ json `df = pd.read_json(path)`

■ xml `df = pd.read_xml(path)`

### 6. 新建欄位

`df[str] = data`

單欄新增，若str是以存在的欄位，其資料內容則會被取代

`df.insert(index,str,data)`

單欄新增，index為插入的欄位位置

`df = df.assign(str1 = data1, str2 = data2, ...)`

多欄新增

### 7. 新建資料(列)

`df2 = df2.append(df1)`

多列新增，df1的欄位名稱要與df2相同，此方法較多限制

# 03 影像



處理影像的套件常見的如下五個：

	OpenCV (cv2)	Matplotlib	Scipy
讀取資料	<code>cv2.imread(path)</code>	<code>matplotlib.image.imread(path)</code>	<code>scipy.misc.imread(path)</code>
資料型別	<code>numpy.ndarray</code>	<code>numpy.ndarray</code>	<code>numpy.ndarray</code>
顯示影像	<code>cv2.imshow(Title, img)</code>	<code>matplotlib.pyplot.imshow(img)</code> <code>matplotlib.pyplot.matshow(img)</code> <code>matplotlib.pyplot.show()</code>	<code>scipy.misc.imshow(img)</code>
儲存影像	<code>cv2.imwrite(path, img)</code>	<code>matplotlib.pyplot.imsave(path, img)</code>	<code>scipy.misc.imsave(path, img)</code>

	PIL	Tensorflow (tf), Keras
讀取資料	<code>PIL.Image.open(path)</code>	<code>tf.keras.preprocessing.image.load_img(path)</code>
資料型別	PIL	PIL
顯示影像	<code>img.show()</code>	<code>img.show()</code>
儲存影像	<code>img.save(path)</code>	<code>tf.keras.preprocessing.image.save_img(path, img)</code>



## 03 影像

1. PIL轉為陣列，可用兩種方法：`tf.keras.preprocessing.image.img_to_array(img)`，`numpy.array(img)`

2. OpenCV讀取影像的彩色通道為BGR。顯示影像也為BGR。

若要轉換彩色通道時，可以使用 `dst = cv2.cvtColor( src, code)`

- `dst` 表示輸出影像，與原始輸入影像具有同樣的資料類型與深度
- `src`表示原始輸入影像
- `code`是色彩空間轉換碼，常見的如右表 與 網站 [ [連結](#) ]：

補充：色彩空間的通道表示如下

- RGB 與 BGR：為光的三原色即紅、綠、藍色
- HSV：H為色調，S為飽和度，V為亮度
- YUV：又稱為YCrCb，Y為亮度，UV 表示色度，即色調與飽和度，分別用Cr和Cb來表示

說明	轉換碼
RGB與BGR轉換	<code>cv2.COLOR_BGR2RGB</code>
	<code>cv2.COLOR_RGB2BGR</code>
RGB/BGR與灰階轉換	<code>cv2.COLOR_BGR2GRAY</code>
	<code>cv2.COLOR_RGB2GRAY</code>
RGB/BGR與HSV轉換	<code>cv2.COLOR_BGR2HSV</code>
	<code>cv2.COLOR_RGB2HSV</code>
	<code>cv2.COLOR_HSV2BGR</code>
	<code>cv2.COLOR_HSV2RGB</code>
RGB/BGR與YUV轉換	<code>cv2.COLOR_BGR2YUV</code>
	<code>cv2.COLOR_RGB2YUV</code>
	<code>cv2.COLOR_YUV2BGR</code>
	<code>cv2.COLOR_YUV2RGB</code>

## 04 影片與攝影機串流

### 1. 初始化 `cap = cv2.VideoCapture(ID)`

初始化攝影機並存至`cap`變數中，其中ID為攝影機的ID號。預設值為-1，表示隨機選取一個攝影機。設為0為第一個攝影機，設為1為第二個攝影機，依此類推。ID也可以給予影片的位置來初始化預播放的影片

### 2. 捕捉畫面 `hasFrame, img = cap.read()`

`hasFrame` 表示捕捉是否成功，如果成功則該值為`True`，不成功為`False`。`img` 是傳回的捕捉到的畫面，如果沒有畫面被捕捉，則該值為空

### 3. 播放視訊時每一畫面的持續停留時間，該參數的單位為ms

`cv2.waitKey(1)`

### 4. 釋放 `cap.release()`

### 5. 銷毀視窗 `cv2.destroyAllWindows()`

```
1 cap = cv2.VideoCapture(0)
2 while True:
3     hasFrame, img = cap.read()
4     cv2.imshow("Frame", img)
5     if cv2.waitKey(1) & 0xFF == ord('q'):
6         break
7 cap.release()
8 cv2.destroyAllWindows()
```

作業

1. (20%) 打印三角形聖誕樹 ( 右圖 ) ，使用for迴圈打印，並儲存至純文字文件：

- 樹葉為底十個 \* 字號，依序減二，最高為兩個 \* 字號且皆置中
- 樹幹為高寬兩個 \*

```
  **
 ****
*****
*****
*****
  **
  **
```

2. (20%) 表格 ( 左圖 ) 再新增欄位為 BMI，其數值為身高與體重所計算的，型別為浮點數取小數點兩位 ( 右圖 )

	age	city	height	weight	sex	SBP	DBP
0	23	Japan	175	68	M	120	85
1	18	Taiwan	168	55	F	114	90
2	30	USA	173	75	M	145	75
3	25	Taiwan	158	50	F	110	78



	age	city	height	weight	sex	SBP	DBP	BMI
0	23	Japan	175	68	M	120	85	22.20
1	18	Taiwan	168	55	F	114	90	19.49
2	30	USA	173	75	M	145	75	25.06
3	25	Taiwan	158	50	F	110	78	20.03



3. (20%) 右圖影像轉為陣列裁減至左上角(140, 220)至右下角(450, 520)的矩形，並用Matplotlib顯示其裁減影像並儲存

# 作業

4. (20%) 以OpenCV讀取影像，不能使用 `cv2.cvtColor()` 而是以陣列取代方式來實現BGR轉RGB通道，並用Matplotlib顯示
5. (20%) 讀取影像並調整影像亮度提高20%與降低20%，並用Matplotlib顯示
6. (20%) 開啟攝影機：
  - 在畫面中繪製紅色正方形在畫面左上角位置(100, 100) 到 右下角位置(200, 200)
  - 將純紅色正方形改為透明度60%的紅色

