

程式設計基本概念

Python 語言程式設計

國立臺中教育大學 數位內容科技學系
吳智鴻 教授

Python之父

- ▶ Python程式語言是由Guido Van Rossum在1980年代末所設計的。



```
1 import copy
2 product = ['code','name','color','category','description','quantity']
3 all = []
4 list = []
5
6
7
8 def dataEntry():
9     for i in range(6):
10        print("What is the " + product[i]+ " of the Product?")
11        answer = input()
12        list.append(answer)
13        new_list = copy.deepcopy(list)
14        print(new_list)
15        all.append(new_list)
16        list.clear()
17
18 while True:
19     dataEntry()
20
21 if input() == "end":
22     print(product)
```


Python優點

- ▶ 程式碼簡潔易懂，適合教學的程式語言。
- ▶ 跨平台和相容性非常好，可執行在多種電腦平台與作業系統中，如:UNIX、Windows、MacOS。
- ▶ 記憶體自動回收。讓我們不用在意記憶體管理，而可以專注在邏輯處理。
- ▶ 易於擴充與維護。可以透過C或C++進行功能擴充。
- ▶ 它是開放的原始碼。對於任何開放、免費的東西通常都會有比較好的人氣，這或許也是其成功的原因之一。

Python應用

- ▶ Python語法簡單，應用卻極為廣泛，以下列出較為常見的應用：
- ▶ 數據分析處理
- ▶ Web網頁開發
- ▶ AI人工智慧
- ▶ 資料庫支援
- ▶ GUI視窗介面(wxPython)、Tkinter
- ▶ 遊戲開發(PyGame)

























IEEE Spectrum 2018 程式語言排名

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	98.4
3. C	  	98.2
4. Java	  	97.5
5. C#	  	89.8
6. PHP		85.4
7. R		83.3
8. JavaScript	 	82.8
9. Go	 	76.7
10. Assembly		74.5
11. Matlab		73.1
12. Scala	 	72.4
13. Ruby	 	71.7
14. HTML		70.7
15. Arduino		69.4
16. Shell		66.3

圖片來源: IEEE Spectrum

2021

IEEE Spectrum / Top Programming Languages

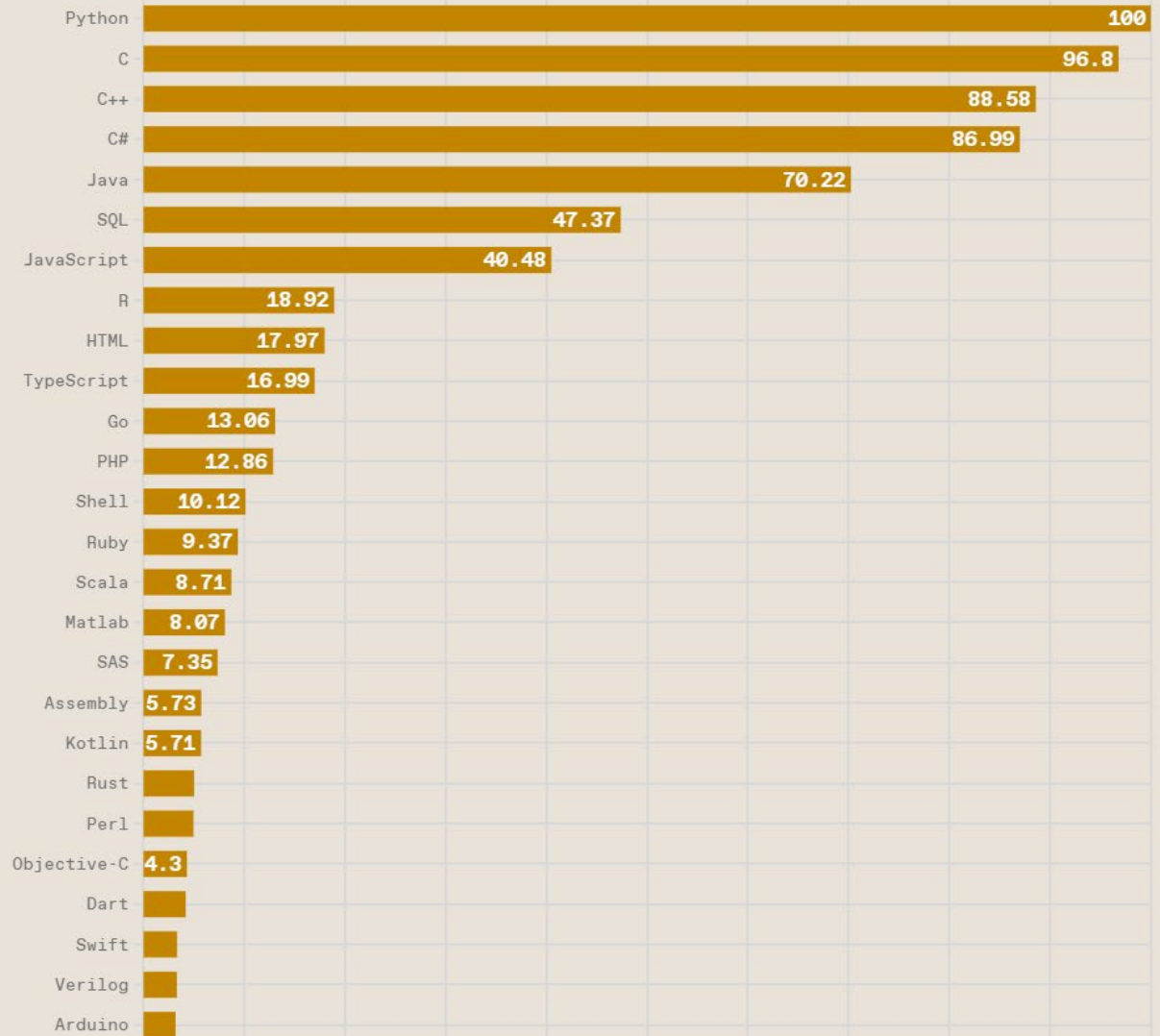
Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	95.4
3	C	  	94.7
4	C++	  	92.4
5	JavaScript		88.1
6	C#	   	82.4
7	R		81.7
8	Go	 	77.7
9	HTML		75.4
10	Swift	 	70.4
11	Arduino		68.4



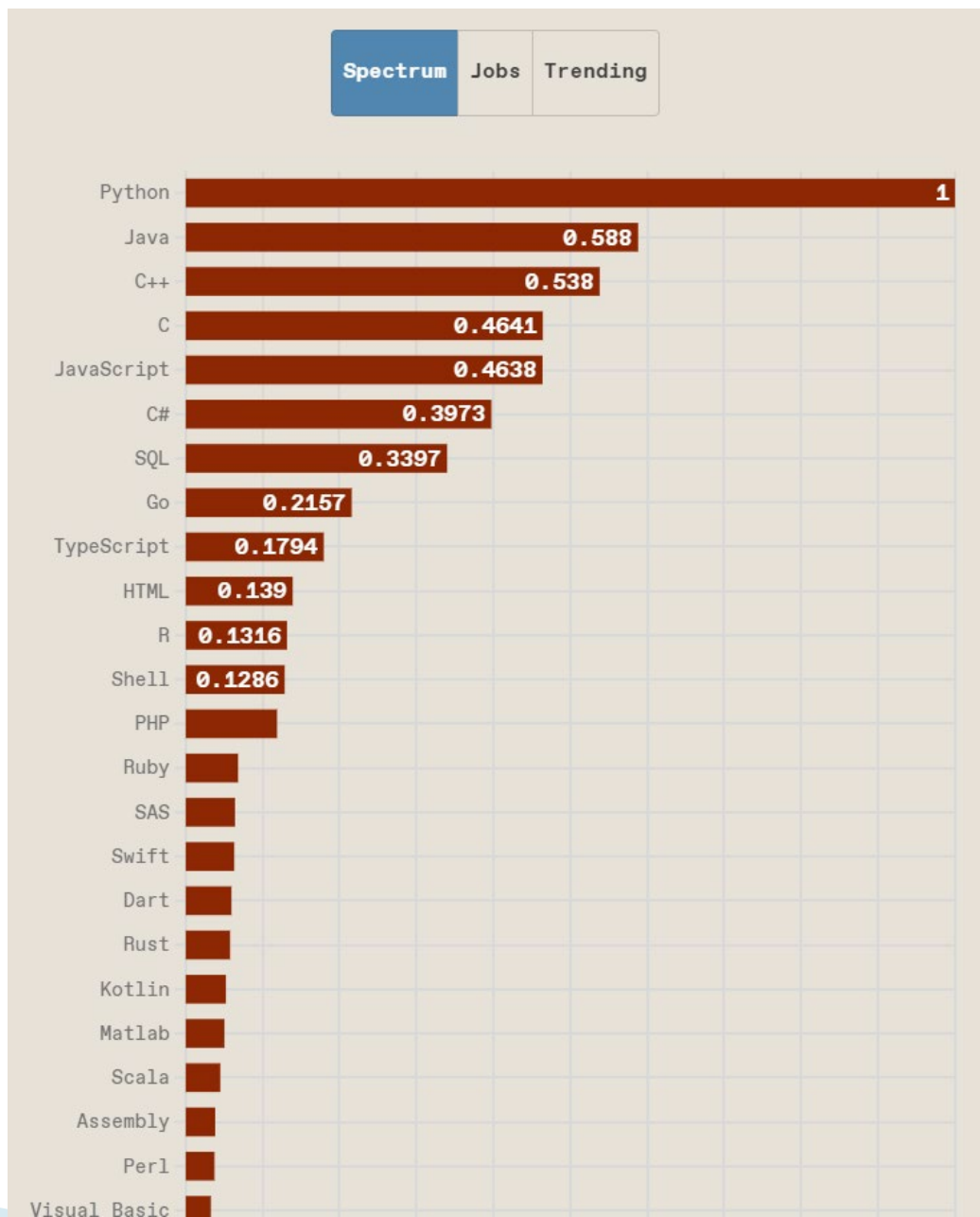
Spectrum Jobs Trending

TAGS
















- PYTHON
- SQL
- TOP PROGRAMMING LANGUAGES
- CODING
- PROGRAMMING LANGUAGES



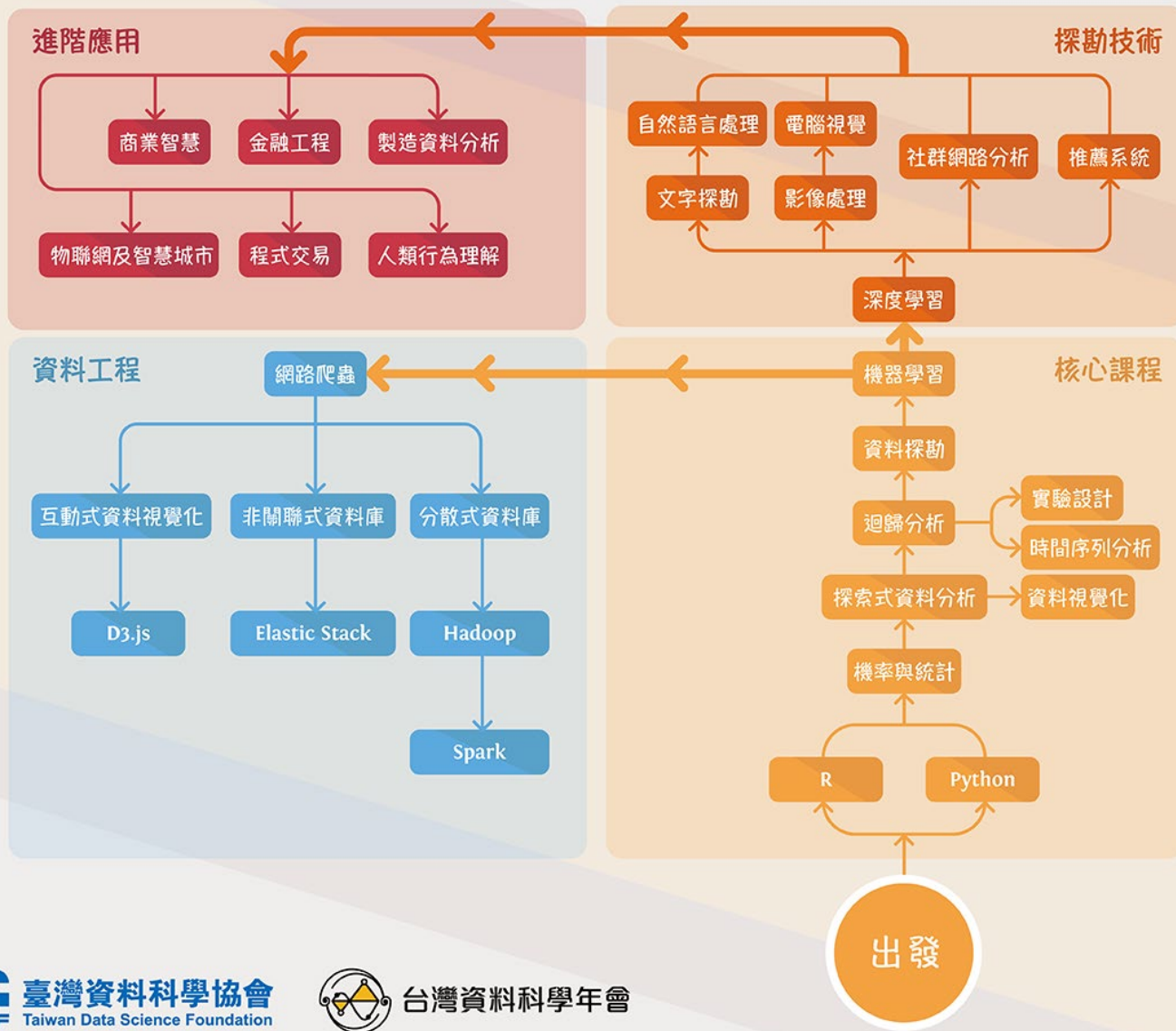
2023



2024

Jan 2024	Jan 2023	Change	Programming Language		Ratings	Change
1	1		 Python		13.97%	-2.39%
2	2		 C		11.44%	-4.81%
3	3		 C++		9.96%	-2.95%
4	4		 Java		7.87%	-4.34%
5	5		 C#		7.16%	+1.43%
6	7	↑	 JS	JavaScript	2.77%	-0.11%
7	10	↑	 PHP		1.79%	+0.40%
8	6	↓	 VB	Visual Basic	1.60%	-3.04%
9	8	↓	 SQL	SQL	1.46%	-1.04%
10	20	↑	 Scratch		1.44%	+0.86%
11	12	↑	 Go		1.38%	+0.23%
12	27	↑	 Fortran		1.09%	+0.64%
13	17	↑	 Delphi/Object Pascal		1.09%	+0.36%
14	15	↑	 MATLAB		0.97%	+0.06%
15	9	↓	 ASM	Assembly language	0.92%	-0.68%

資料科學學習地圖



國立臺中教育大學數位內容科技學系 人工智慧微型學分學程

設置要點

110 年 12 月 28 日 110 學年度第 1 學期校課程委員會通過

- 一、設置目的：依據國立臺中教育大學學分學程設置辦法辦理，為培養學生人工智慧與大數據專長，特訂定本要點。
- 二、本學分學程總學分必須至少選修 6 學分，具體課程請詳見學分學程科目表。

科目代碼	科目名稱	選別	最低學分	時數	開課年級	備註
ADT11010	人工智慧應用 Artificial Intelligence Application	選	3	3	三上	
	大數據分析 Big Data Analysis	選	3	3	三下	

- 三、修讀資格：本校大二（含）以上在學學生均可修習。
- 四、人數限制：每班符合資格者在最低選課人數以上始得開班，並依規定設置上限。
- 五、申請核可程序：有意修讀者，須檢附申請表及歷年成績單，於每學期教務處所規定申請之時間內，向本系提出書面申請，超過每科目上課人數上限，則由數位內容科技學系組成小組審查後擇優錄取。
- 六、證書發給：學生須修滿規定之 6 學分且成績及格者，檢具歷年成績單正本，向本系提出申請核發微型學分學程證明書，經系主任審核後，簽請院長、教務長、校長同意後核發，修畢部份學分者，其成績載於畢業總成績單，不另發證明。
- 七、本要點如有未盡事宜，依本校相關辦法辦理之。
- 八、本要點經系課程委員會及系務會議通過，循序送院及校課程委員會審議通過，並經教務會議備查，陳請校長核定後實施，修正時亦同。

許多AI技術串接均使用Python

Openai with DALLE3 function call



Documentation API reference Examples

Forum Help

Search CTRL K

GET STARTED

Overview
Introduction
Quickstart
Models
Tutorials

CAPABILITIES

Text generation
Function calling
Embeddings
Fine-tuning
Image generation
Introduction
Usage
Language-specific tips
Vision
Text-to-speech
Speech-to-text
Moderation

Usage

Generations

The [image generations](#) endpoint allows you to create an original image given a text prompt. When using DALL-E 3, images can have a size of 1024×1024, 1024×1792 or 1792×1024 pixels.

By default, images are generated at `standard` quality, but when using DALL-E 3 you can set `quality: "hd"` for enhanced detail. Square, standard quality images are the fastest to generate.

You can request 1 image at a time with DALL-E 3 (request more by making parallel requests) or up to 10 images at a time using DALL-E 2 with the `n` parameter.

Generate an image

python Copy

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 response = client.images.generate(
5     model="dall-e-3",
6     prompt="a white siamese cat",
7     size="1024x1024",
8     quality="standard",
9     n=1,
10 )
11
12 image_url = response.data[0].url
```



What is new with DALL-E 3

Explore what is new with DALL-E 3 in the OpenAI Cookbook

遊戲開發(多媒體概論)

- ▶ UNITY
 - C#

網頁程式設計

- ▶ 網頁前端
 - HTML
 - CSS
 - JavaScript
- ▶ 網頁後端
 - PHP
 - 資料庫 SQL

Python 好處

- ▶ Python 邁向人工智慧的基礎
- ▶ 資料科學家的基礎

程式設計的三大面向

INPUT

- 輸入
- 鍵盤
- 滑鼠

PROCESS

- 處理
- 加減乘除運算
- 資料結構
 - 變數
 - 陣列
- 程式邏輯
 - 迴圈

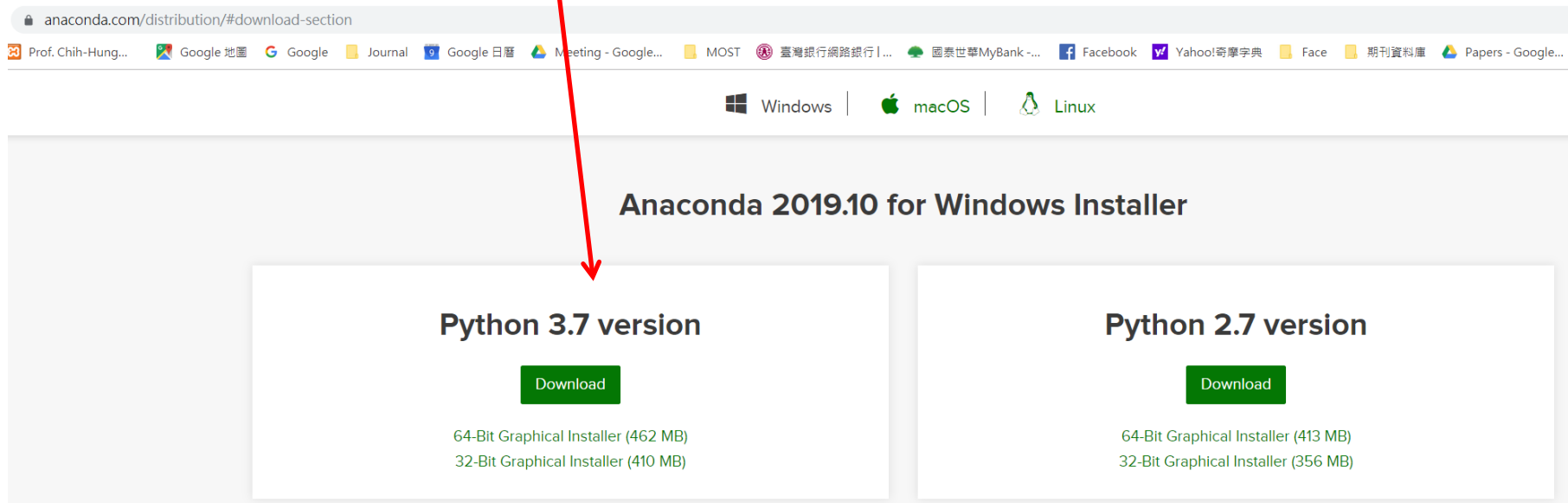
OUTPUT

- 螢幕顯示
- 文字檔
- 資料庫

Python語言 開發工具

▶ Python

- 免費的Python語言開發環境
- 從Anaconda網站下載(Python 3.7, 64bit)

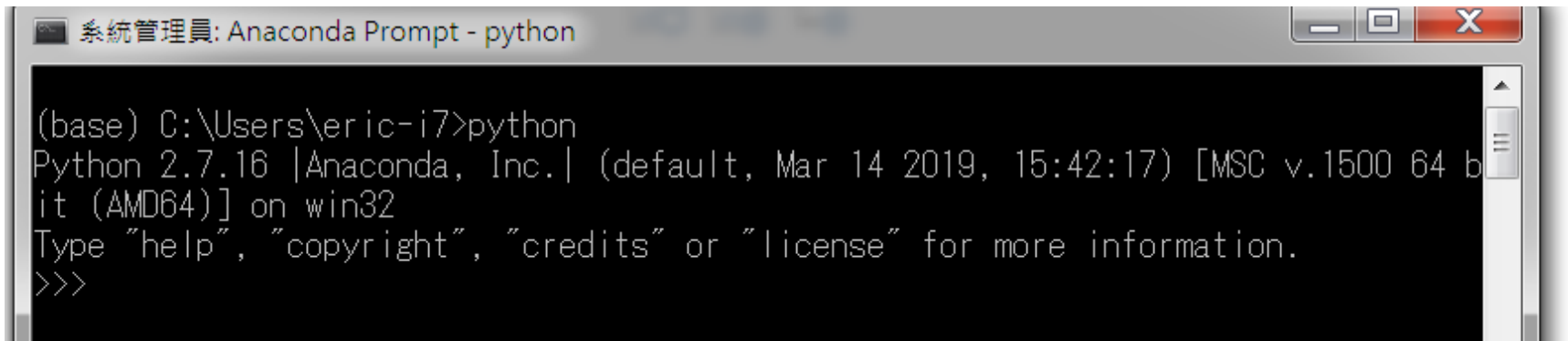


The screenshot shows the Anaconda website's download section. The browser address bar displays 'anaconda.com/distribution/#download-section'. The page title is 'Anaconda 2019.10 for Windows Installer'. Below the title, there are two main download options:

- Python 3.7 version**: Includes a green 'Download' button and lists '64-Bit Graphical Installer (462 MB)' and '32-Bit Graphical Installer (410 MB)'. A red arrow points to this section.
- Python 2.7 version**: Includes a green 'Download' button and lists '64-Bit Graphical Installer (413 MB)' and '32-Bit Graphical Installer (356 MB)'.

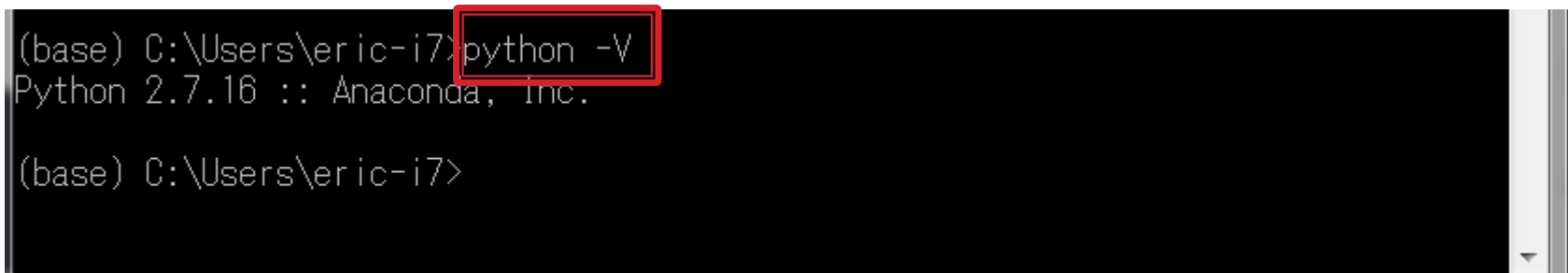
Navigation links for Windows, macOS, and Linux are visible above the download options.

命令列執行Python



```
系統管理員: Anaconda Prompt - python
(base) C:\Users\eric-i7>python
Python 2.7.16 |Anaconda, Inc.| (default, Mar 14 2019, 15:42:17) [MSC v.1500 64 bit
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

查看python版本：`python -V`



```
(base) C:\Users\eric-i7>python -V
Python 2.7.16 :: Anaconda, Inc.

(base) C:\Users\eric-i7>
```

AI models in python

explainability, fairness, bias, ethics

AIF360

deon

Aequitas

Skater

visualization

Seaborn

Altair

Bokeh

Shapely

Rasterio

Pydot

Matplotlib

plotnine

Plotly

Cartopy

Geopandas

analysis, modeling

Airflow

Rasa

AllenNLP

scikit-learn

Keras

PyTorch

StatsModels

Theano

TensorFlow

PyMC3

SciPy

Gensim

NetworkX

data representation

Pandas

NumPy

datasketch

Modin

spaCy

NLTK

RDFlib

data access

SQLAlchemy

Pillow

BeautifulSoup

network resources

PyArrow

Scrapy

Requests

Flasgger

Istio

application frameworks

JupyterLab

Dask

PyWren

Ray

Flask

PySpark

Project Jupyter

Jupyter Enterprise Gateway

Gunicorn

package management

Pip

Conda

AI開發環境介紹

人工智慧

開發環境工具介紹

吳智鴻 教授

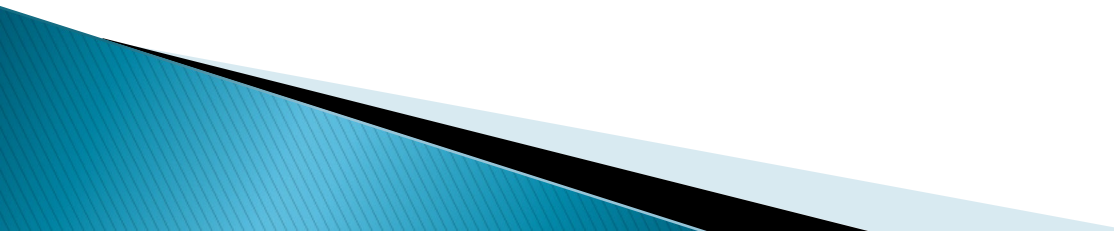
國立臺中教育大學 數位內容科技學系

EMAIL:CHWU@MAIL.NTCU.EDU.TW

WEBSITE: 120.108.221.55/PROFCHWU/DCTAI

2019/09/25

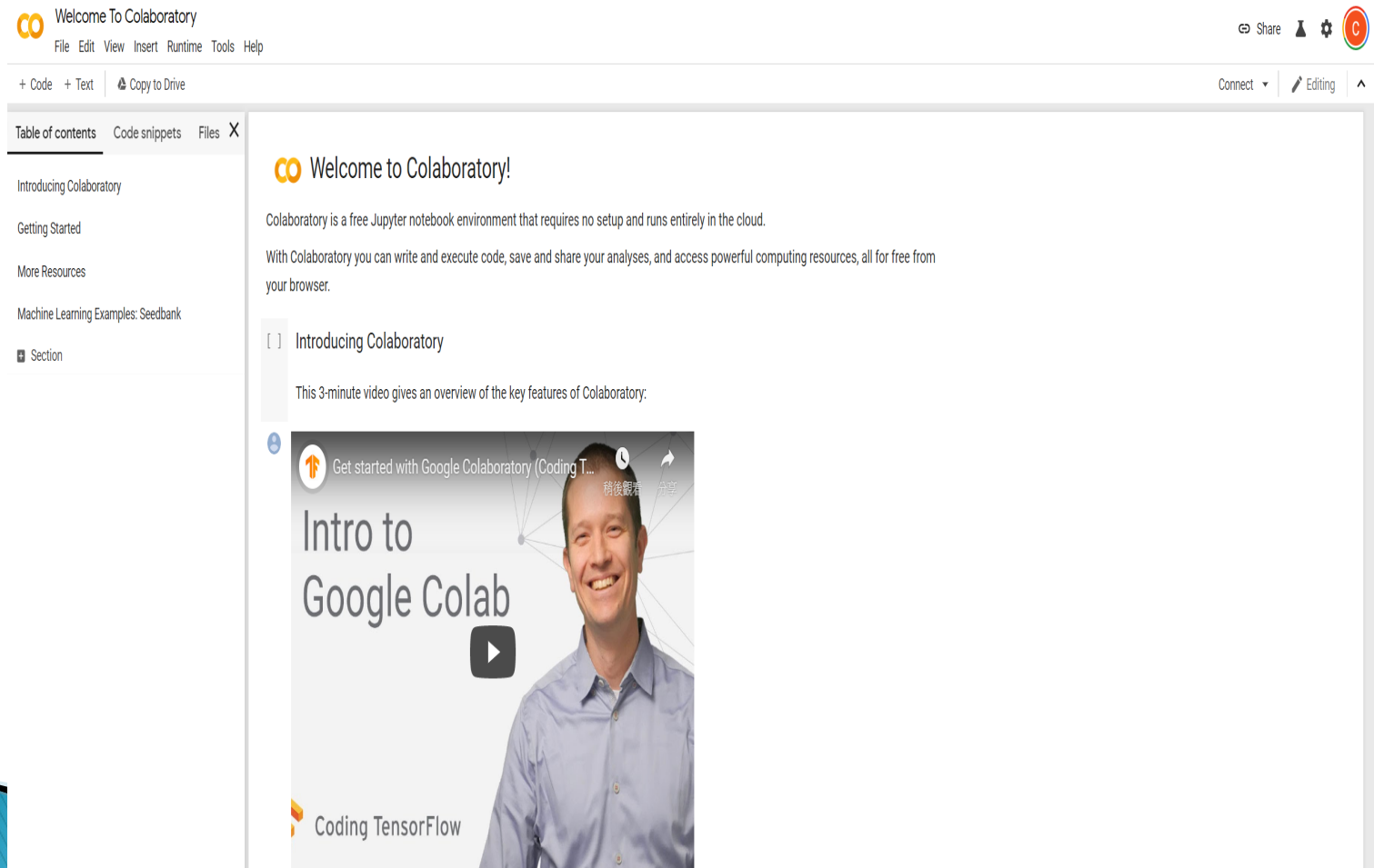
可用的開發工具

- ▶ Spyder
 - ▶ Python直譯器
 - ▶ Jupyter notebook
 - ▶ VS code
 - ▶ Google Colab
- 

線上開發工具

<https://colab.research.google.com/notebooks/welcome.ipynb>

▶ Google Colab



The screenshot shows the Google Colaboratory interface. At the top, there is a navigation bar with the Google Colab logo, the text "Welcome To Colaboratory", and a menu with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side of the navigation bar, there are icons for "Share", "Settings", and a user profile icon. Below the navigation bar, there are buttons for "+ Code", "+ Text", and "Copy to Drive". On the right side of this bar, there are "Connect" and "Editing" options. The main content area is divided into a left sidebar and a main panel. The sidebar contains a "Table of contents" section with links to "Introducing Colaboratory", "Getting Started", "More Resources", "Machine Learning Examples: Seedbank", and a "Section" button. The main panel displays the "Welcome to Colaboratory!" message, followed by a paragraph explaining that Colaboratory is a free Jupyter notebook environment that runs in the cloud. Below this, there is a section titled "Introducing Colaboratory" with a sub-heading "This 3-minute video gives an overview of the key features of Colaboratory:". A video player is embedded, showing a video titled "Get started with Google Colaboratory (Coding T..." with a play button and a "稍後觀看" (Watch later) button. The video thumbnail features a man in a blue shirt and the text "Intro to Google Colab" and "Coding TensorFlow".

Google Colab優點

- ▶ 免費
- ▶ 提供GPU/TPU 運算
- ▶ 什麼前置安裝都不需要，請打開瀏覽器，登入 **Google** 帳號，就可開始
- ▶ 學習A.I. ...
- ▶ 未來晉升中高階開發者，也可無痛轉換成Jupyter 繼續開發
- ▶ 與**Google Drive** 連動，方便存取與備份以外，也方便與他人協作
- ▶ 允許執行命令視窗指令如pip，夠方便

資料來源：我想學 Python + A.I.，該從何入手？

Google Colab缺點

- ▶ 最多 12 小時的連續運算
- ▶ 每次重新開啟 Colab，都要重新連動 Drive
- ▶ 如果要使用 Colab 無預設安裝好的特定模組，那每次開啟 Colab 時都要重新安裝一遍
- ▶ 與 Google Drive 連動是優點也是缺點，像是資料容量過大的話可能就放不進去 Google Drive

Example #1

輸入與輸出

Example #1

輸入身高、體重

並印出BMI (體重 / 身高 (公尺) ^2)

```
In [1]: #Example#1  
height = int(input('請輸入身高 cm :'))  
weight = int(input('請輸入體重 kg :'))
```

請輸入身高 cm :180

請輸入體重 kg :70

```
In [2]: mybmi = weight / ((height/100)**2)  
print('My BMI is %d' % mybmi)
```

My BMI is 21

if判斷式

```
if運算式:  
    述句  
    述句..  
  
elif運算式:  
    述句  
    述句..  
  
else:  
    述句  
    述句..
```

```
score = 87  
  
if 90 <= score <= 100:      #如果if為真  
    print('A')              #執行此處  
elif score >= 80:           #如果if為假而這裡為真  
    print('B')              #執行此處  
elif score >= 70:           .  
    print('C')              .  
elif score >= 60:           .  
    print('D')              .  
else:                        #若上述皆假  
    print('F')              #執行此處
```

Example#2 bmi

- ▶ 輸入姓名
- ▶ 輸入身高、體重
- ▶ 計算BMI
- ▶ 並列出建議

	身體質量指數(BMI) (kg/m ²)	腰圍 (cm)
體重過輕	BMI < 18.5	-
正常範圍	18.5 ≤ BMI < 24	-
異常範圍	過重：24 ≤ BMI < 27 輕度肥胖：27 ≤ BMI < 30 中度肥胖：30 ≤ BMI < 35 重度肥胖： BMI ≥ 35	男性：≥ 90公分 女性：≥ 80公分

Example #2

BMI計算程式

▶ 輸入資料

```
In [1]: #Example#2  
name = str(input('請輸入姓名')) #輸入姓名  
height = int(input('請輸入身高 cm :'))  
weight = int(input('請輸入體重 kg :'))
```

請輸入姓名吳老師
請輸入身高 cm :172
請輸入體重 kg :72

▶ 計算

```
In [2]: print('Your name is %s' % name)  
mybmi = weight / ((height/100)**2)  
print('My BMI is %d' % mybmi)
```

Your name is 吳老師
My BMI is 24

▶ 判斷

```
In [3]: if mybmi < 18.5:  
        print('過瘦')  
        if mybmi >=18.5 and mybmi < 24:  
            print('正常')  
            if mybmi >=24 and mybmi < 27:  
                print('過重')  
                if mybmi >=27 and mybmi < 30:  
                    print('輕度肥胖')  
                    if mybmi >=30 and mybmi < 35:  
                        print('中度肥胖')  
                        if mybmi >=35:  
                            print('重度肥胖')
```

過重

迴圈結構

- ▶ 可讓程式重複執行
- ▶ 有分兩種類型
- ▶ For迴圈
 - 進入迴圈前先判斷條件是否符合
- ▶ while迴圈

- 又分兩類

```
Do {  
.....  
} While (條件)
```

(至少迴圈內容會被執行一次)

```
For (i=1; i<=9; i++)
```

```
{
```

```
    迴圈內容
```

```
    此部分程式會被重複執行9次
```

```
}
```

```
While (條件)
```

```
{ .....
```

```
}
```

(迴圈內容可能不會被執行)

Example #3

九九乘法表 需有雙重迴圈概念

```
# Example3 99乘法表
```

```
for i in range(2,10):  
    for j in range(2,10):  
        s= i*j  
        print ('%d * %d = %2d ' %(i, j , s))
```

函數語法

```
range(start, stop[, step])
```

range(開始值, 結束值 (不包括), 每次累加值)

Ex. Range(0,5)是 [0,1,2,3,4] 不包括5

```
2 * 1 = 2  
2 * 2 = 4  
2 * 3 = 6  
2 * 4 = 8  
2 * 5 = 10  
2 * 6 = 12  
2 * 7 = 14  
2 * 8 = 16  
2 * 9 = 18  
3 * 1 = 3  
3 * 2 = 6  
3 * 3 = 9  
3 * 4 = 12  
3 * 5 = 15  
3 * 6 = 18  
3 * 7 = 21  
3 * 8 = 24  
3 * 9 = 27  
4 * 1 = 4  
4 * 2 = 8  
4 * 3 = 12  
4 * 4 = 16  
4 * 5 = 20  
4 * 6 = 24  
4 * 7 = 28  
4 * 8 = 32  
4 * 9 = 36  
5 * 1 = 5  
5 * 2 = 10  
5 * 3 = 15  
5 * 4 = 20  
5 * 5 = 25  
5 * 6 = 30  
5 * 7 = 35  
5 * 8 = 40  
5 * 9 = 45  
6 * 1 = 6  
6 * 2 = 12  
6 * 3 = 18  
6 * 4 = 24  
6 * 5 = 30  
6 * 6 = 36
```

Example #3b

九九乘法表 輸出不換行

```
# Example3b 99乘法表 (不換行)
```

```
for i in range(2,10):  
    for j in range(2,10):  
        s= i*j  
        print ('%d * %d = %2d ' %(i, j , s) )
```

```
2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18  
3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27  
4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36  
5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45  
6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54  
7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63  
8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72  
9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

解答在後面

Example #3b 解答

九九乘法表 输出不换行

```
# Example3b 99乘法表 (不换行)
```

```
for i in range(2,10):  
    for j in range(2,10):  
        s= i*j  
        print ('%d * %d = %2d ' %(i, j , s), end="")  
    print()
```

```
2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18  
3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27  
4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36  
5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45  
6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54  
7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63  
8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72  
9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

Example 4

- ▶ 試著用While迴圈來完成99乘法表

解答在後面

Example4解答

#Example 4 While 迴圈99乘法表

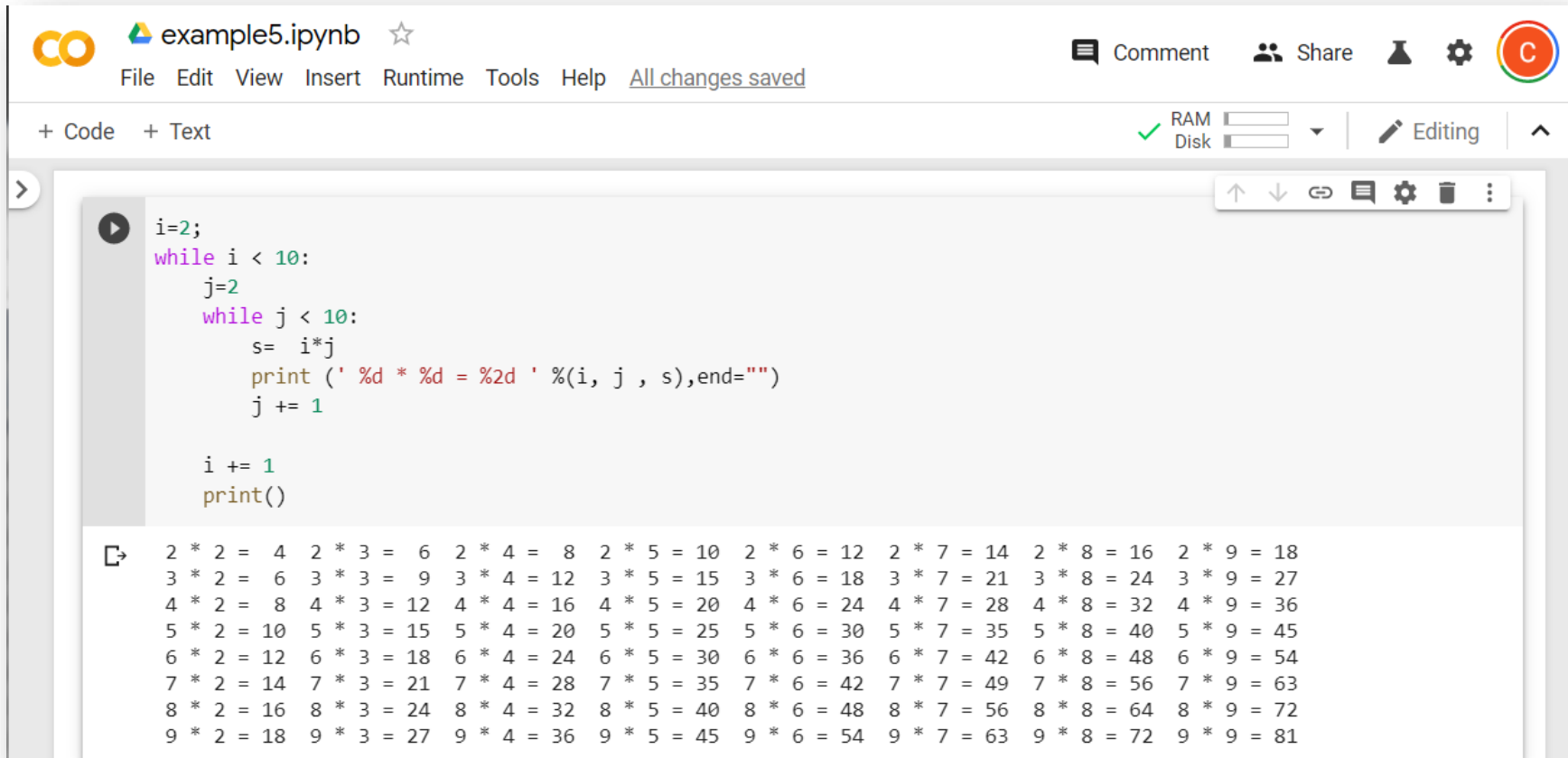
```
i=2;
while i < 10:
    j=2
    while j < 10:
        s= i*j
        print (' %d * %d = %2d ' %(i, j , s),end="")
        j += 1
    i += 1
print()
```

2 * 2 = 4	2 * 3 = 6	2 * 4 = 8	2 * 5 = 10	2 * 6 = 12	2 * 7 = 14	2 * 8 = 16	2 * 9 = 18
3 * 2 = 6	3 * 3 = 9	3 * 4 = 12	3 * 5 = 15	3 * 6 = 18	3 * 7 = 21	3 * 8 = 24	3 * 9 = 27
4 * 2 = 8	4 * 3 = 12	4 * 4 = 16	4 * 5 = 20	4 * 6 = 24	4 * 7 = 28	4 * 8 = 32	4 * 9 = 36
5 * 2 = 10	5 * 3 = 15	5 * 4 = 20	5 * 5 = 25	5 * 6 = 30	5 * 7 = 35	5 * 8 = 40	5 * 9 = 45
6 * 2 = 12	6 * 3 = 18	6 * 4 = 24	6 * 5 = 30	6 * 6 = 36	6 * 7 = 42	6 * 8 = 48	6 * 9 = 54
7 * 2 = 14	7 * 3 = 21	7 * 4 = 28	7 * 5 = 35	7 * 6 = 42	7 * 7 = 49	7 * 8 = 56	7 * 9 = 63
8 * 2 = 16	8 * 3 = 24	8 * 4 = 32	8 * 5 = 40	8 * 6 = 48	8 * 7 = 56	8 * 8 = 64	8 * 9 = 72
9 * 2 = 18	9 * 3 = 27	9 * 4 = 36	9 * 5 = 45	9 * 6 = 54	9 * 7 = 63	9 * 8 = 72	9 * 9 = 81

Example5

- ▶ 試著在Google Colab中完成

Example5



The screenshot shows a Jupyter Notebook interface. At the top, there is a header with the Colab logo, the filename "example5.ipynb", and a star icon. Below this is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", followed by the text "All changes saved". On the right side of the header, there are icons for "Comment", "Share", a flask icon, a gear icon, and a circular icon with a "C". Below the header, there is a toolbar with "+ Code" and "+ Text" on the left, and "RAM", "Disk", and "Editing" on the right. The main area contains a code cell with the following Python code:

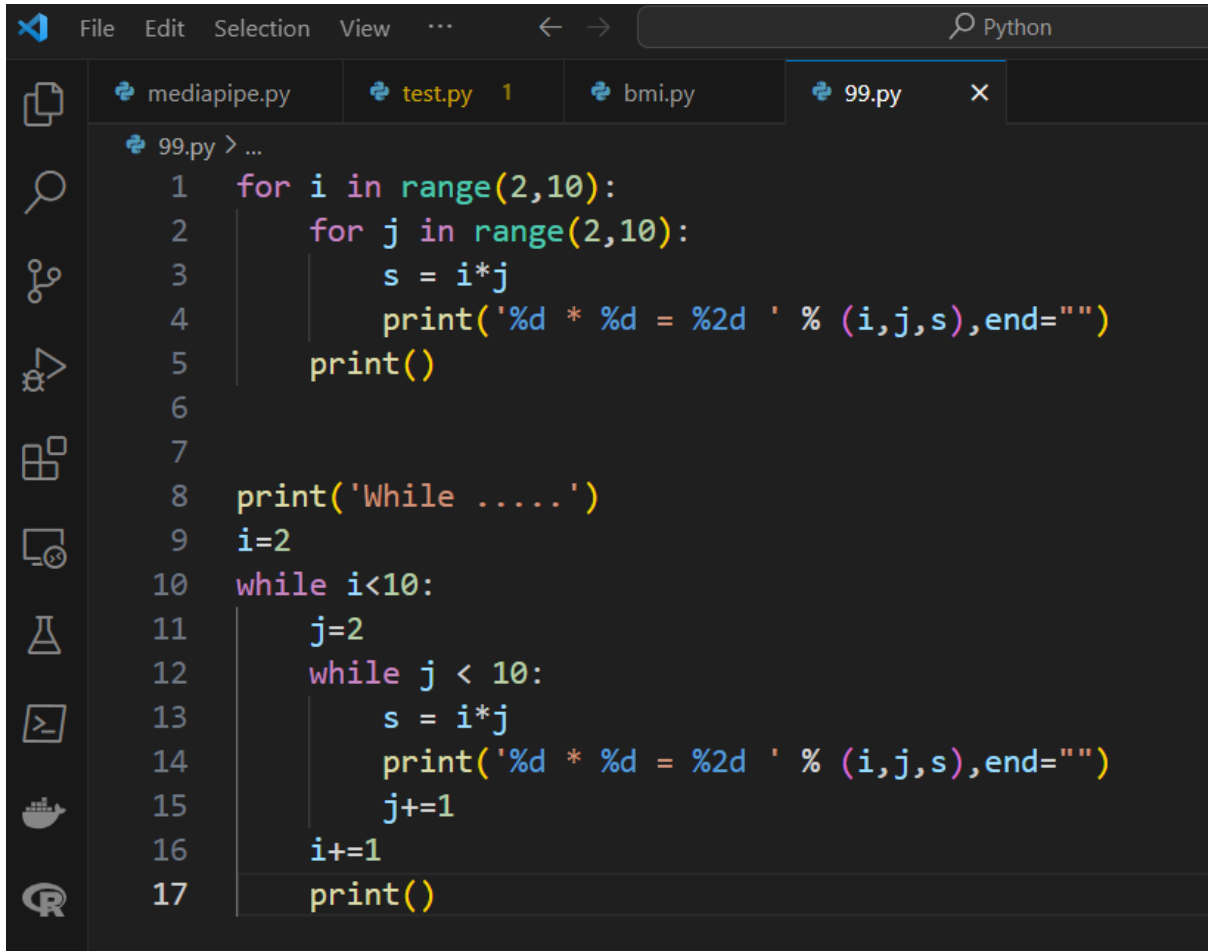
```
i=2;
while i < 10:
    j=2
    while j < 10:
        s= i*j
        print (' %d * %d = %2d ' %(i, j , s),end="")
        j += 1

    i += 1
    print()
```

The output of the code cell is a 9x9 grid of multiplication results, displayed as a single line of text with a copy icon on the left:

```
2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18
3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27
4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36
5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45
6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54
7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63
8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72
9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

VS Code



```
File Edit Selection View ... Python
mediapipe.py test.py 1 bmi.py 99.py x
99.py > ...
1 for i in range(2,10):
2     for j in range(2,10):
3         s = i*j
4         print('%d * %d = %2d ' % (i,j,s),end="")
5     print()
6
7
8 print('While .....')
9 i=2
10 while i<10:
11     j=2
12     while j < 10:
13         s = i*j
14         print('%d * %d = %2d ' % (i,j,s),end="")
15         j+=1
16     i+=1
17 print()
```


Results

```
e C:/Users/eric/Documents/Python/99.py
2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18
3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27
4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36
5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45
6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54
7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63
8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72
9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
while .....
2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18
3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27
4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36
5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45
6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54
7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63
8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72
9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
PS C:\Users\eric\Documents\Python>
```