

# Python 程式設計

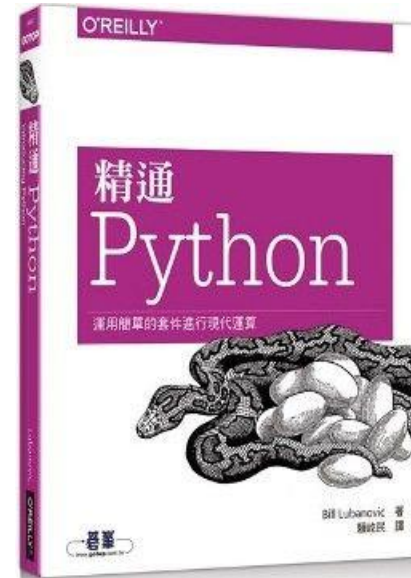
講師：蔡宜璋

yc0815@gmail.com

# 課程綱要

- 01 簡介與程式基礎
- 02 開發環境建置
- 03 基礎語法
- 04 變數、運算、資料型態
- 05 流程控制與迴圈
- 06 字串格式化處理
- 07 自訂函式、內建函式
- 08 結構化資料操作
- 09 字串操作
- 10 類別與物件導向

# Python參考書籍



# Python\_01

## Python 簡介

# Python 可以做什麼？

- 終端機應用 (Console application)
- 桌面程式應用 (Desktop application)
- 網站應用 (Web application)
- 行動應用 (Mobile application)
- 機器學習 (Machine Learning)
- 資料科學 (Data Science)
- 物聯網應用 (IOT application)
- And many more ...



Python之父  
吉多·范羅蘇姆(Guido van Rossum)

# Python 起源

- Python 在1989年聖誕節期間，由創始人吉多·范羅蘇姆( Guido van Rossum) 為了打發聖誕節的閒暇，在荷蘭阿姆斯特丹所開發的。第一個正式版本釋出在1991年。
- 選 Python (大蟒蛇的意思) 作為程式語言的名字，源自於Guido是蒙提·派森的飛行馬戲團(Monty Python's Flying Circus)的愛好者，蒙提·派森的飛行馬戲團是一部由英國六人喜劇團體主演的BBC電視短劇。
- Python 的前身是ABC語言，ABC語言是由 Guido 參加設計的一種教學語言。Guido 認為針對非專業人員所設計的ABC 語言非常優美和強大。
- Guido 認為ABC語言沒有推廣成功，主要是封閉造成。Python 希望能吸引更多UNIX / C的程式設計師加入，因此Python可與其他熱門的程式語言，如：C、C++、Java 進行良好的結合。

# Python 版本

## **Python 0.9.0 - February 20, 1991**

Python 0.9.1 - 1991

Python 0.9.2 - 1991

## **Python 1.0 - January 1994**

Python 1.2 - April 10, 1995

Python 1.3 - October 12, 1995

Python 1.4 - October 25, 1996

Python 1.5 - December 31, 1997

Python 1.6 - September 5, 2000

## **Python 2.0 - October 16, 2000**

Python 2.1 - April 15, 2001

Python 2.2 - December 21, 2001

Python 2.3 - July 29, 2003

Python 2.4 - November 30, 2004

Python 2.5 - September 19, 2006

Python 2.6 - October 1, 2008

Python 2.7 - July 3, 2010

## **Python 3.0 - December 3, 2008**

Python 3.1 - June 27, 2009

Python 3.2 - February 20, 2011

Python 3.3 - September 29, 2012

Python 3.4 - March 16, 2014

Python 3.5 - September 13, 2015

Python 3.6 - December 23, 2016

Python 3.7 - June 27, 2018

## **Python 3.8 – October 19, 2019**





# Python 特色

- 容易學習，開發快速
  - 語法好學易懂，適合初學者，可加速其他語言學習。
  - 可用幾行程式碼，完成複雜的任務。
    - 例如：三行程式碼下載 Youtube 影片。
- 跨平台（跨作業系統）
  - 同樣的程式碼可以在不同作業系統執行。
- 強大的標準函式庫（內建模組）
  - Python內建的標準函式庫，提供廣泛的內建模組，如：文字處理、檔案處理、作業系統、網路通信、W3C格式、多國語系、數學運算、圖形界面、...、等。
- 網路社群資源豐富
  - 提供眾多實用的第三方模組。
  - 社群樂於分享教學，協助新手解決問題。

# Life is short, you need Python.

- Python 經典的第三方套件

Web框架：Django、Flask、Pyramid、web2py

GUI開發：PyQt、Tkinter、wxPython

App開發：kivy

爬蟲：Scrapy、selenium、beautifulsoup

資料科學：numpy、scipy、matplotlib、pandas

機器學習：keras、tensorflow



- Python 官方套件網站：Pypi

- Python 第三方套件推薦網站

- [https://python.libhunt.com/categories?fbclid=IwAR3NRxa8sCBaK-ndsx6D5fkYbQOny1Ya2qRlpD-Z6cRJZi\\_0kJ-Ff4xKLZw](https://python.libhunt.com/categories?fbclid=IwAR3NRxa8sCBaK-ndsx6D5fkYbQOny1Ya2qRlpD-Z6cRJZi_0kJ-Ff4xKLZw)

- [https://awesome-python.com/?fbclid=IwAR0IKGm67\\_Vw3fjcW1Ns9foSupGvHPOfeKenKKEBIXeppZzqMs5h-bF\\_BaM](https://awesome-python.com/?fbclid=IwAR0IKGm67_Vw3fjcW1Ns9foSupGvHPOfeKenKKEBIXeppZzqMs5h-bF_BaM)

# 知名企業都在用Python

The popular YouTube video sharing system is largely written in Python



Google makes extensive use of Python in it's web search system



Dropbox storage service codes both its server and client software primarily in Python



The Raspberry Pi single-board computer promotes Python as its educational language



Raspberry Pi

## COMPANIES USING PYTHON



BitTorrent peer-to-peer file sharing system began its life as a Python Program



NASA uses Python for specific Programming Task



The NSA uses Python for cryptography and intelligence analysis

**NETFLIX**

Netflix and Yelp have both documented the role of Python in their software infrastructures

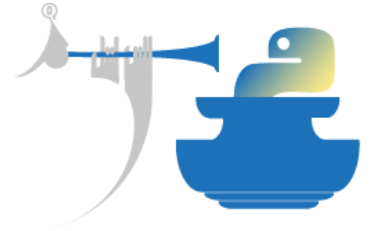
# 程式語言受歡迎排名

<https://www.tiobe.com/tiobe-index/>



Apr 2020	Apr 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.73%	+1.69%
2	2		C	16.72%	+2.64%
3	4	▲	Python	9.31%	+1.15%
4	3	▼	C++	6.78%	-2.06%
5	6	▲	C#	4.74%	+1.23%
6	5	▼	Visual Basic	4.72%	-1.07%
7	7		JavaScript	2.38%	-0.12%
8	9	▲	PHP	2.37%	+0.13%
9	8	▼	SQL	2.17%	-0.10%
10	16	▲	R	1.54%	+0.35%
11	19	▲	Swift	1.52%	+0.54%
12	18	▲	Go	1.36%	+0.35%
13	13		Ruby	1.25%	-0.02%
14	10	▼	Assembly language	1.16%	-0.55%
15	22	▲	PL/SQL	1.05%	+0.26%
16	14	▼	Perl	0.97%	-0.30%
17	11	▼	Objective-C	0.94%	-0.57%
18	12	▼	MATLAB	0.93%	-0.36%
19	17	▼	Classic Visual Basic	0.83%	-0.23%
20	27	▲	Scratch	0.77%	+0.28%

# The Zen of Python



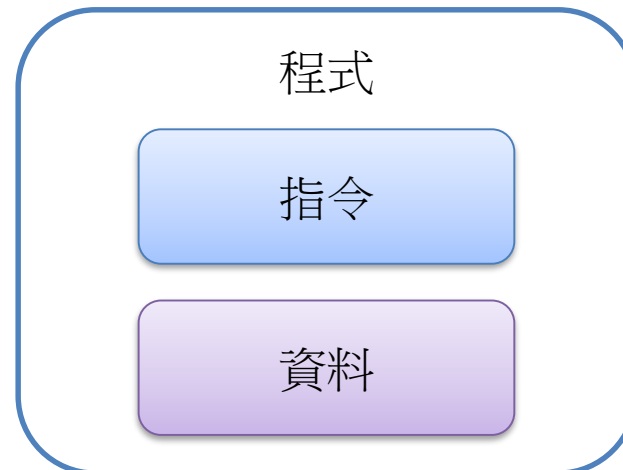
```
>>> import this
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

優美優於醜陋，  
明瞭優於隱晦；  
簡單優於複雜，  
複雜優於凌亂，  
扁平優於嵌套，  
稀疏優於稠密，  
可讀性很重要！  
即使實用比純粹更優，  
特例亦不可違背原則。  
錯誤絕不能悄悄忽略，  
除非它明確需要如此。  
面對不確定性，  
拒絕妄加猜測。  
任何問題應有一種，  
且最好只有一種，  
顯而易見的解決方法。  
儘管這方法一開始並非如此直觀，  
除非你是荷蘭人。  
做優於不做，  
然而不假思索還不如不做。  
很難解釋的，必然是壞方法。  
很好解釋的，可能是好方法。  
命名空間是個絕妙的主意，  
我們應好好利用它。

# 程式設計基礎

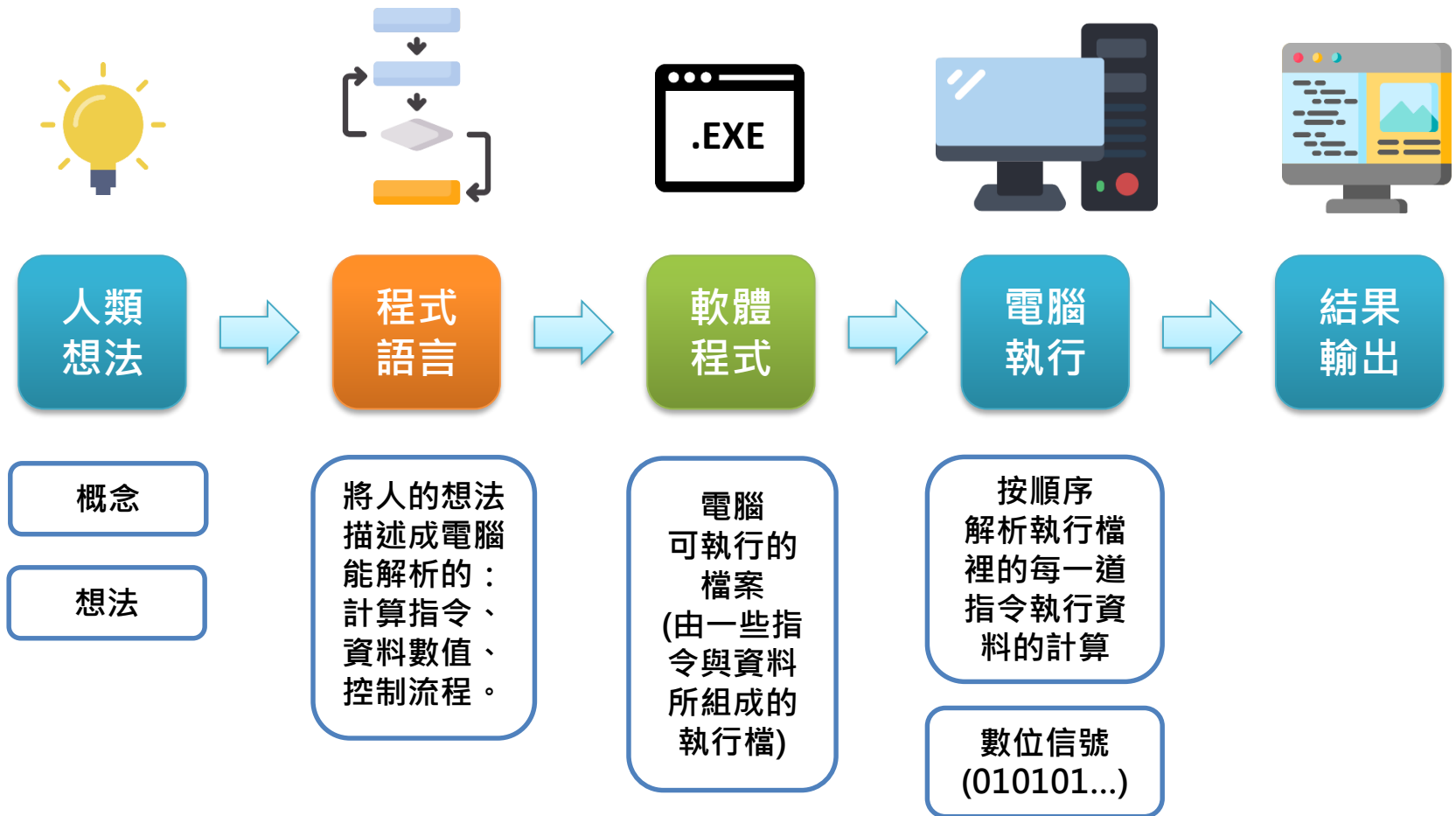
# 程式

- 指令
  - 執行計算的最小單位(例如：加、減、乘、除、比較、載入、分支跳躍)
  - 二進位格式，指令長度為**32**位元或**64**位元
  - 電腦中央處理器(CPU)指令集當中的指令(如：x86 指令集)
- 資料
  - 文字符號(字元)、數值(有號數、無號數、浮點數)
  - 指令計算時資料須載入記憶單元(如：CPU暫存器或快取記憶體)
- 程式
  - 由指令與資料所組成
  - 透過特定程式語言編寫程式碼
  - 程式可在電腦上執行
    - 以執行檔方式執行
    - 以程式碼直接執行



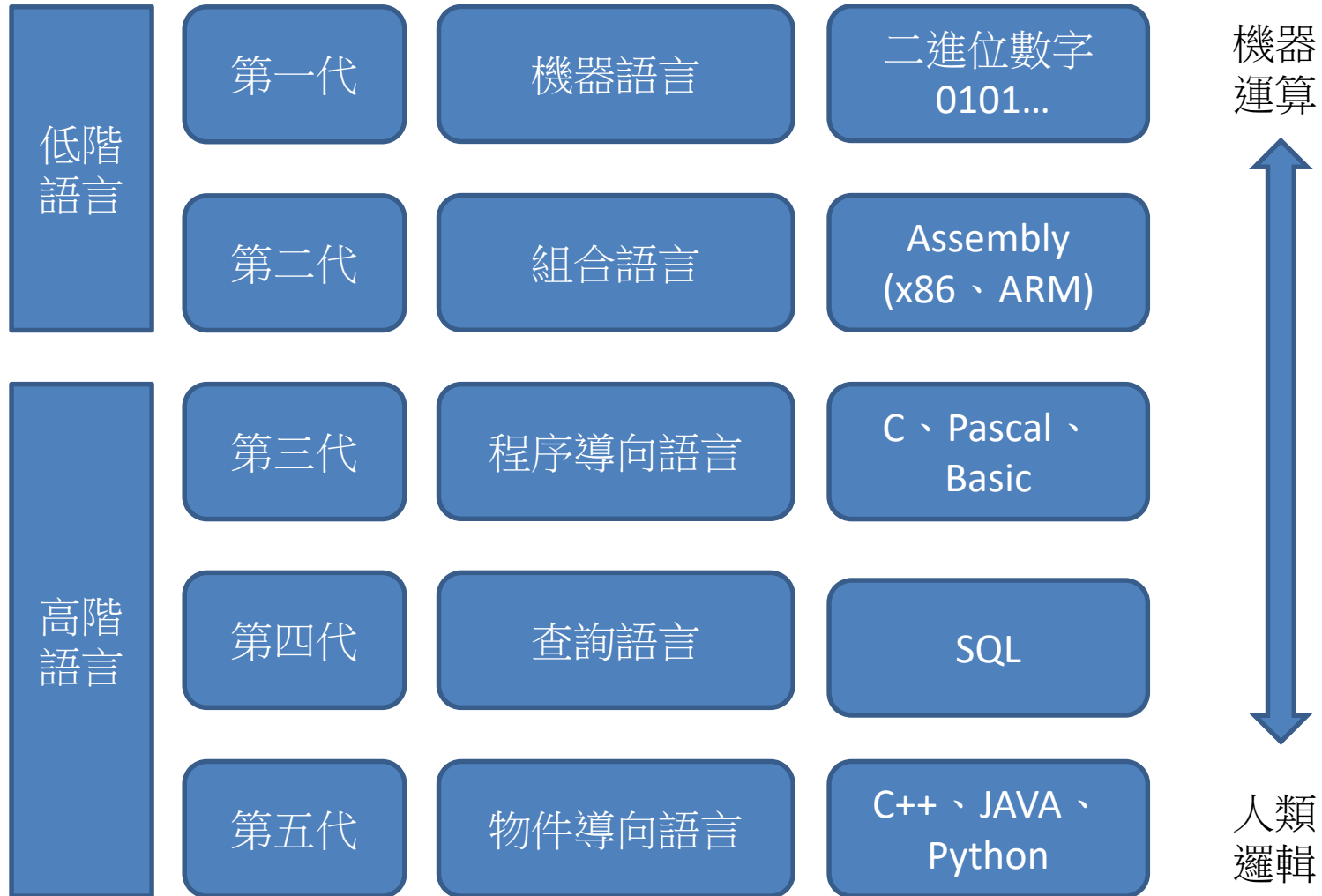
# 程式語言

- 程式語言(programming language)是指電腦能解析的語言
- 將人的想法，實現成電腦可執行的軟體程式(Software program)





# 程式語言的分類與演進



# 軟體程式分類與架構

應用程式

文書編輯  
軟體

程式開發  
軟體

網路應用  
軟體

資料庫  
軟體

用戶自製  
軟體

系統程式

作業系統  
(Windows / Linux / Mac OS)

管理執行中的應用程式  
(Process)

管理硬體的驅動程式  
(Driver)

電腦硬體

中央  
處理器

主記  
憶體

儲存  
裝置

I/O  
裝置

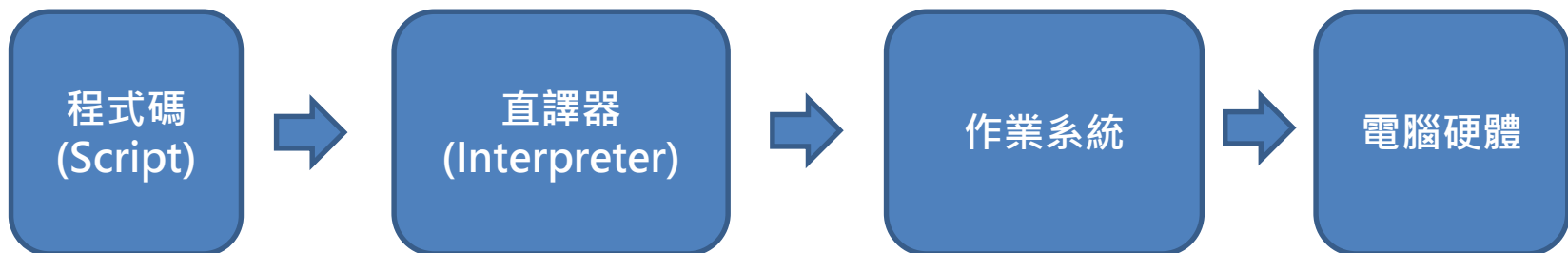
其他  
裝置

# 程式碼的執行

- 編譯式語言(程式碼編譯成執行檔)：C、C++



- 直譯式語言(不用編譯)：Linux Shell, Python, Perl, Ruby, Java, JavaScript, PHP



腳本

直譯器本身就是一種程式(執行檔)  
直譯器會讀取使用者寫的腳本來執行

# 編譯式語言的執行(以C為例)

程式碼

```
// hello.c
#include <stdio.h>
int main() {
    printf("Hello, world!\n");
    return 0;
}
```

編譯

```
gcc -o hello hello.c
```

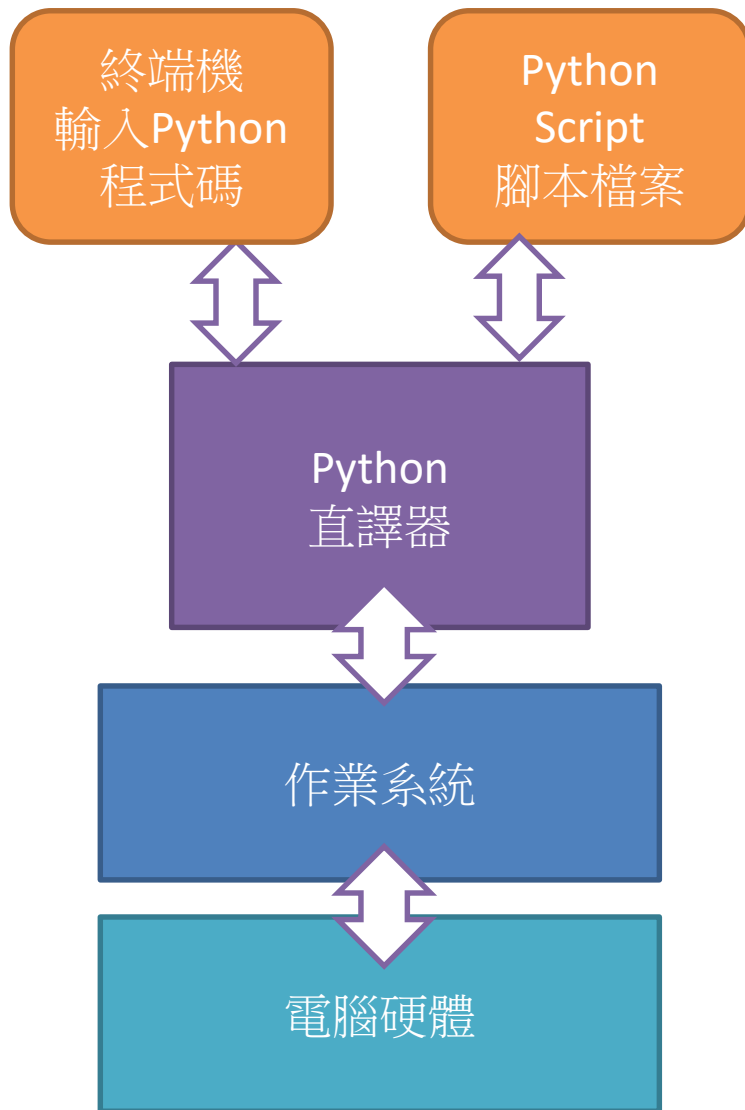
執行

```
./hello
```

輸出

```
Hello, world!
```

# 直譯式語言的執行(以Python為例)



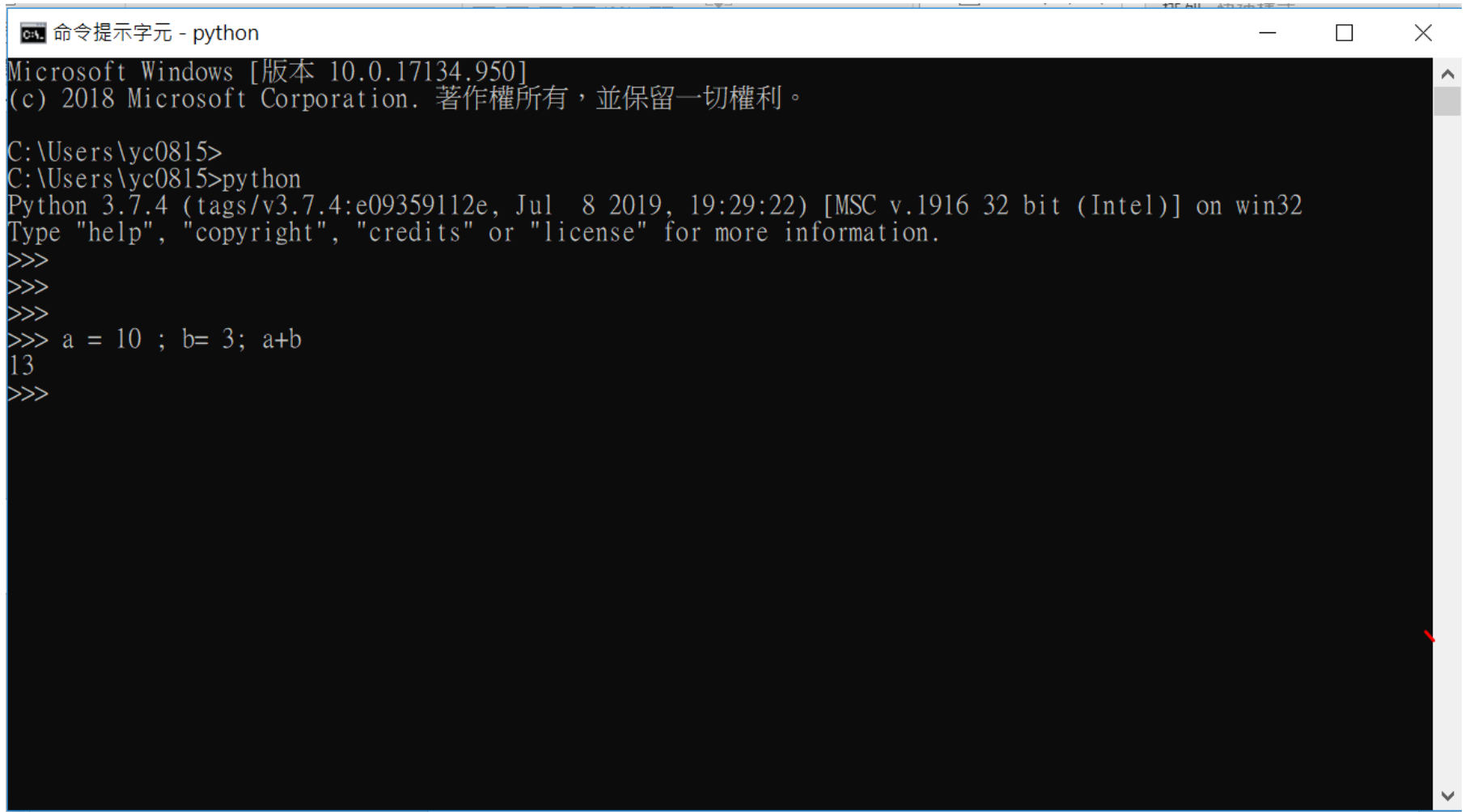
工作管理員

檔案(F) 選項(O) 檢視(V)

處理程序 效能 應用程式歷程記錄 開機 使用者 詳細資料 服務

名稱	狀態	13% CPU	78% 記憶體	0% 磁碟	0% 網路
應用程式 (12)					
> Google Chrome (118)		2.8%	5,326.0 ...	0.1 MB/秒	0 Mbps
> Microsoft Edge (52)		0.1%	155.9 MB	0 MB/秒	0 Mbps
> Microsoft PowerPoint (2)		0.1%	173.4 MB	0 MB/秒	0 Mbps
> Microsoft Word		0%	20.1 MB	0 MB/秒	0 Mbps
▼ mysql.exe (2)		0%	0.3 MB	0 MB/秒	0 Mbps
mysql.exe		0%	0.1 MB	0 MB/秒	0 Mbps
主控台視窗主機		0%	0.2 MB	0 MB/秒	0 Mbps
> PyCharm (3)		0%	63.1 MB	0 MB/秒	0 Mbps
▼ Python (32 位元) (2)		0%	9.8 MB	0 MB/秒	0 Mbps
Python (32 位元)		0%	2.4 MB	0 MB/秒	0 Mbps
主控台視窗主機		0%	7.4 MB	0 MB/秒	0 Mbps
> Windows 命令處理程式 (3)		0%	9.5 MB	0 MB/秒	0 Mbps
> Windows 檔案總管 (2)		2.2%	72.6 MB	0.1 MB/秒	0 Mbps
> 工作管理員		1.2%	42.6 MB	0 MB/秒	0 Mbps

# Python透過終端機與直譯器互動



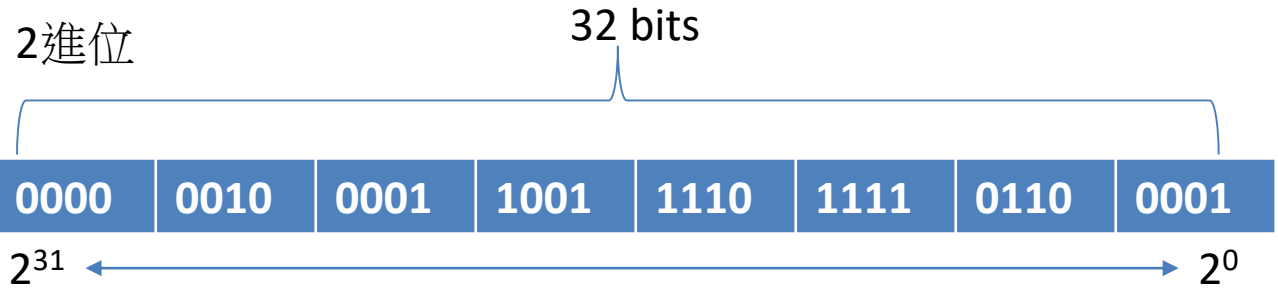
```
命令提示字元 - python
Microsoft Windows [版本 10.0.17134.950]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\yc0815>
C:\Users\yc0815>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> a = 10 ; b= 3; a+b
13
>>>
```

# 電腦的數字系統

- 2進位
- 16進位
- 10進位

二進位	十進位	十六進位
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F



16進位

0	2	1	9	E	F	6	1
---	---	---	---	---	---	---	---

10進位表示 2,223,969

# 電腦的文字編碼

- **ASCII 美國資訊交換標準碼** ( American Standard Code for Information Interchange )，ASCII 碼早期使用 7 位元編碼，後來則為了相容性改為使用 **8 個位元編碼**，因此可以產生  $2^8 = 256$  種組合，目前 ASCII 只定義了前 128 碼 ( 包括英文大小寫字母、數字與常用符號等 )，後面的 128 碼則多由軟體廠商自行定義相關的控制字元與特殊符號。
- **BIG-5 大五碼**，台灣目前使用最為普遍的中文碼是由行政院委託民間編制的大五碼 ( BIG-5 碼 )，是使用 16 個位元來編碼。而中國大陸和新加坡則普遍使用國標碼 ( GB 碼 ) 來處理簡體中文。
- **Unicode 萬國碼** 對世界上大部分的文字系統進行了整理、編碼，是由 IBM、微軟、昇陽、蘋果等大公司共同制訂與發展出來的一套編碼方式，目前已經收錄超過 13 萬個字元，解決字元編碼在電腦上不同語系無法相容的問題。
- **UTF-8** ( 8-bit Unicode Transformation Format ) 是一種針對 Unicode 的可變長度字元編碼，也是一種字首碼。可以用來表示 Unicode 標準中的任何字元，且其編碼中的第一個位元組仍與 ASCII 相容，這使得原來處理 ASCII 字元的軟體無須或只須做少部份修改，即可繼續使用。因此，它逐漸成為電子郵件、網頁及其他儲存或傳送文字優先採用的編碼。



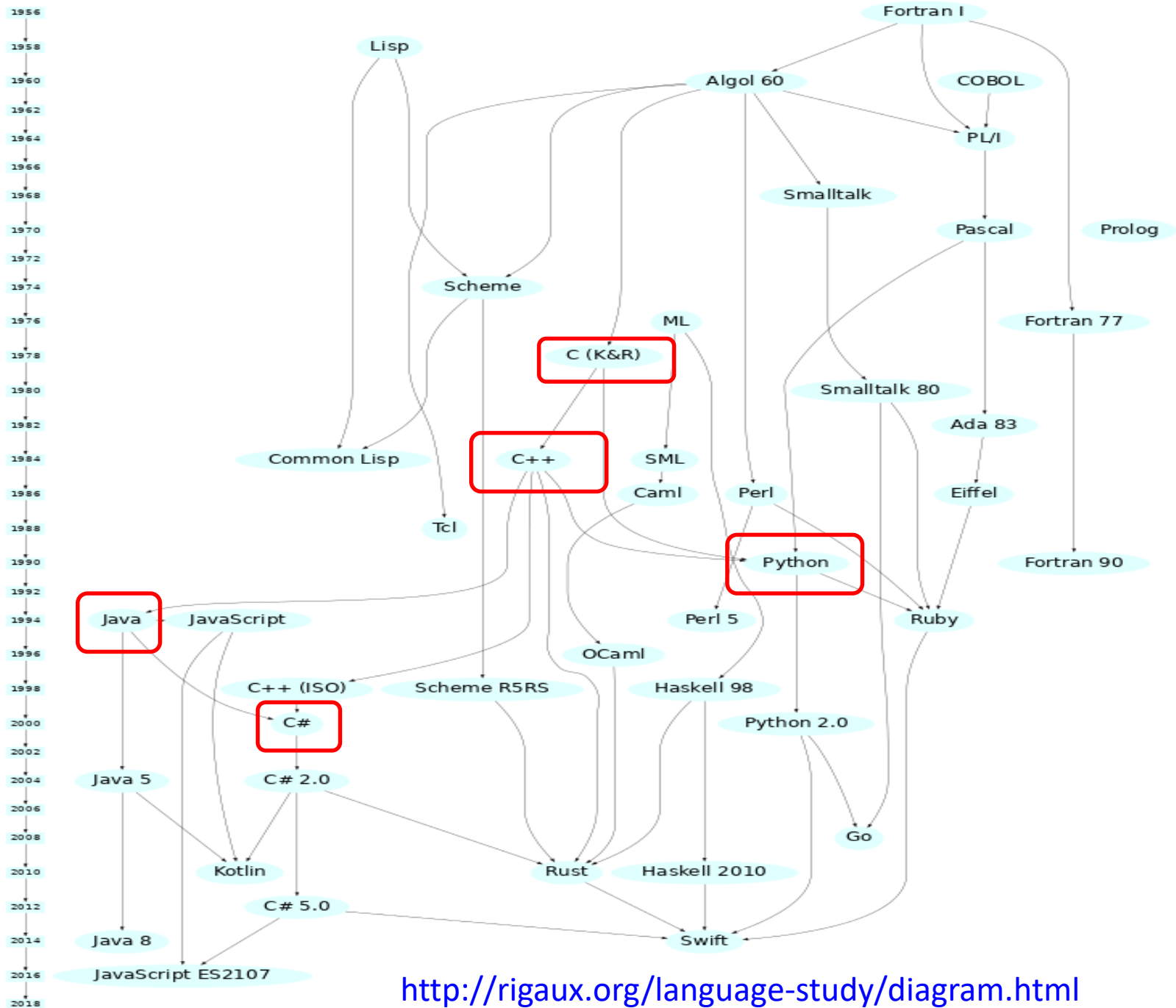
# ASCII 編碼表

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

# 單元練習 01

- Python是直譯還是編譯語言?
- Python是程序導向還是物件導向?
- ASCII編碼 65是什麼字元?

# 補充資料



# 有號數與無號數

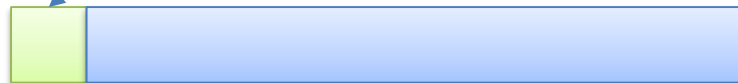
- 整數
  - 正數(自然數)、零、負數
- 無號數：正數、零，使用二進位表示
- 有號數：正數、零、負數，使用2的補數表示

『 3 』，二進位表示為: 0011

『 -3 』的 1 的補數 表示為: 1100

『 -3 』的 2 的補數 即為: 1101

Sign bit 符號位元



十進制	2 的補數
$+ 2^{n-1}-1$	0111 1111 1111 ... 1111
...	...
$+ 2$	0000 0000 0000 ... 0010
$+ 1$	0000 0000 0000 ... 0001
0	0000 0000 0000 ... 0000
$- 1$	1111 1111 1111 ... 1111
$- 2$	1111 1111 1111 ... 1110
$- 3$	1111 1111 1111 ... 1101
...	...
$- 2^{n-1}$	1000 0000 0000 ... 0000