

# 變數與 互動運算

- Python 程式語言簡介
- Python 變數與資料型態
- Python 互動運算

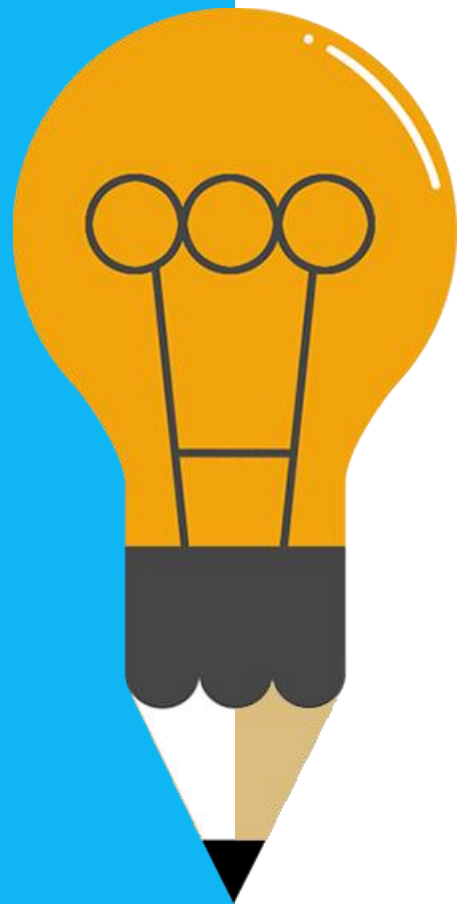


# 學習目標：變數

- **Who(對象)：變數資料**
- **Why(目的緣由)：為什麼需要變數**
  1. 為了讓程式知道使用者所需的資料
  2. 讀取資料來進行運算處理再輸出
- **When(應用時機)/Where(應用場域)：變數資料的應用時機**

人機互動取值：程式/使用者取得資訊的時候
- **What(內容定義)：變數資料可以做甚麼事**
  1. 資料定義
  2. 儲存資料
  3. 資料傳輸
- **How(操作機制)：變數資料的產生運作機制**
  1. 變數定義：變數命名規則與變數內容值指派
  2. 資料型態：數值、字串、容器(串列、字典)
  3. 讀入/輸出資料：input/print

# Python 程式語言簡介



1

程式語言 V.S. 語言

2

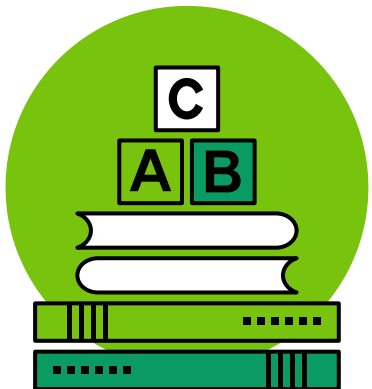
執行程式：語法錯誤、語意錯誤

# 程式語言

學習程式語言的目的是為了和電腦溝通，  
能指揮電腦進行我們指定的運算流程。

程式語言和語言相似，涉及使用單字、文法、語意。

- 詞彙：主詞、動詞、受詞
- 句型：句子的形成規則，也就是文法規則
- 語意：該句子的意思



# 語言 V.S. 程式語言

## 英語

詞彙：英文單字詞彙

➤ 主詞、動詞、受詞

句型：英語文法。

語意：該句子的意思。

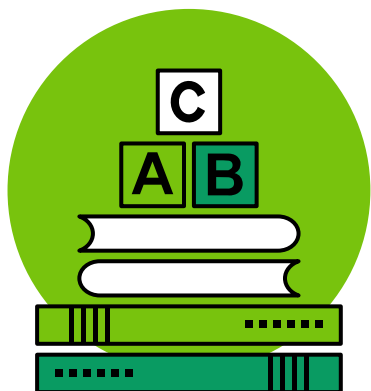
## 程式語言

詞彙：程式所使用的詞彙

➤ 包含保留字、變數

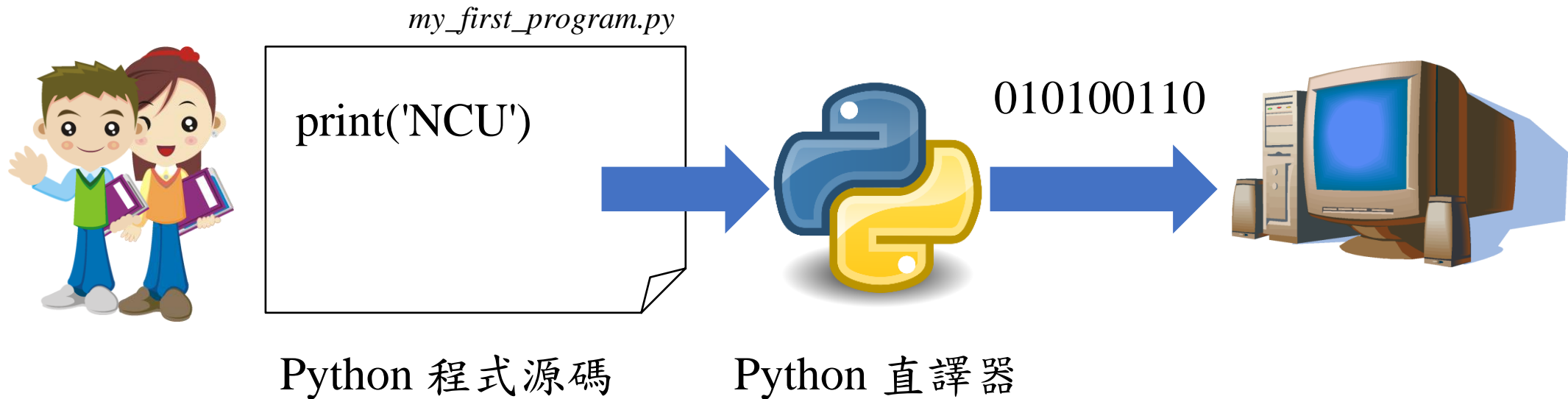
句型：程式語法。

語意：程式的執行結果。



# 執行程式

- 程式源碼 (source code)：依循程式語言語法撰寫可供電腦執行的程式。
- 我們目前學習的程式語言多屬高階語言。
- 透過直譯器將程式源碼翻譯成電腦指令再由電腦執行。



# 翻譯英文

## 正確的例子

翻譯

原文是 英文 ▾



翻譯成 中文(繁體) ▾

翻譯

義大利文 韓文 英文 偵測語言

I went to the beach yesterday.



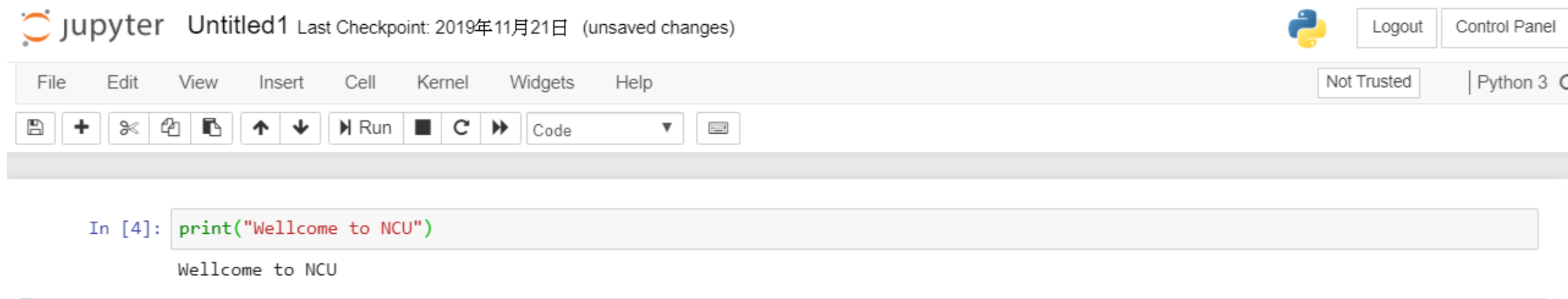
中文(繁體) 英文 中文(簡體)

我昨天去海灘。



# 執行程式

## 正確的例子



使用 Python 直譯器將程式源碼翻譯成電腦運算動作。



# 翻譯英文:語法錯誤

## 英文語法錯誤的例子

翻譯

原文是 英文 ▾



翻譯成 中文(繁體) ▾

翻譯

義大利文 韓文 英文 偵測語言

I go to the beach yesterday.



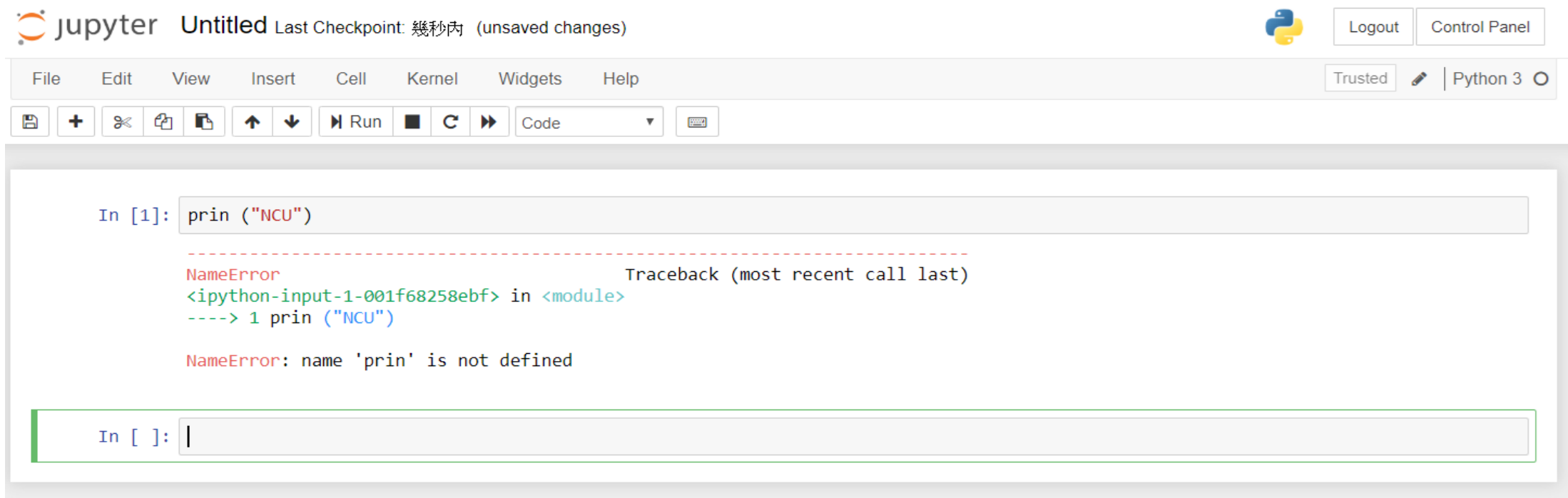
中文(繁體) 英文 中文(簡體)

我去海邊昨天。



# 執行程式：語法錯誤

## 程式語法錯誤的例子



The image shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by 'Untitled' and 'Last Checkpoint: 幾秒內 (unsaved changes)'. On the right, there are 'Logout' and 'Control Panel' buttons. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar are 'Trusted', a pencil icon, and 'Python 3'. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area shows a code cell with the input 'In [1]: prin ("NCU")'. Below the input, a red error message is displayed: 'NameError' followed by a traceback showing the error occurred in the current cell at line 1. The message states 'NameError: name 'prin' is not defined'. Below the error message is an empty code cell with the prompt 'In [ ]: |'.

```
In [1]: prin ("NCU")

-----
NameError                                Traceback (most recent call last)
<ipython-input-1-001f68258ebf> in <module>
----> 1 prin ("NCU")

NameError: name 'prin' is not defined

In [ ]: |
```

錯誤的程式語法無法執行程式，並且會出現錯誤訊息。

# 翻譯英文：語意錯誤

- 正確的英文句子被翻譯成正確的中文句子

偵測語言	中文	英文	↕	↔	英文	中文 (繁體)	中文 (簡體)	↕
I am a student.			×		我是學生。			☆

- 
- 正確的英文文法句子被翻譯成怪異的中文句子（語意錯誤）

偵測語言	中文	英文	↕	↔	英文	中文 (繁體)	中文 (簡體)	↕
I am a dog.			×		我是一隻狗。			☆

# 執行程式：語意錯誤

程式源碼：程式語法正確但程式邏輯錯誤

```
In [4]: a = 10  
b = 20  
c = a - b  
print("a + b = c")  
print("a = ",a," , b = ",b," , a + b = ",c)
```

```
a + b = c  
a = 10 , b = 20 , a + b = -10
```

程式語法正確但程式邏輯錯誤直譯後執行結果不符預期。（語意錯誤）

```
In [5]: a = 10  
b = 20  
c = a + b  
print("a = ",a," , b = ",b," , a + b = ",c)
```

```
a = 10 , b = 20 , a + b = 30
```

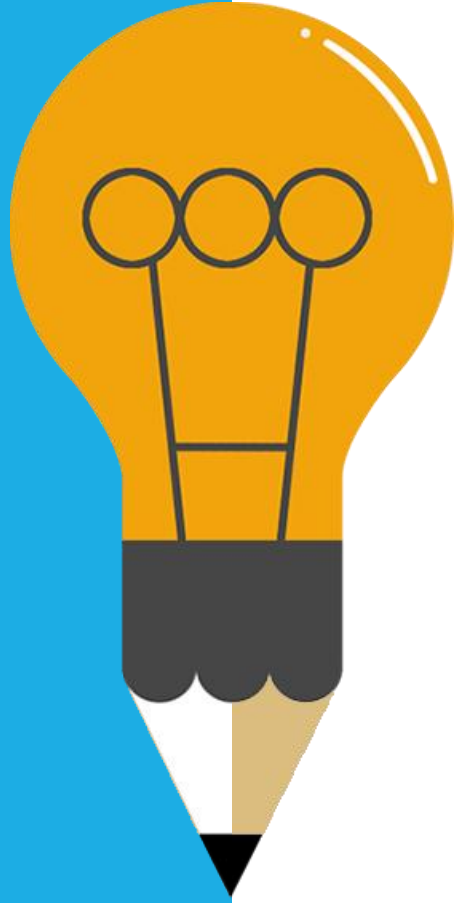
程式語法與邏輯均正確直譯後得到符合預期的結果。

# 進入Python 程式世界

- Python 程式語言簡介
- **Python 變數與資料型態**
- Python 變數運算



# 變數與資料型態



1

變數命名

2

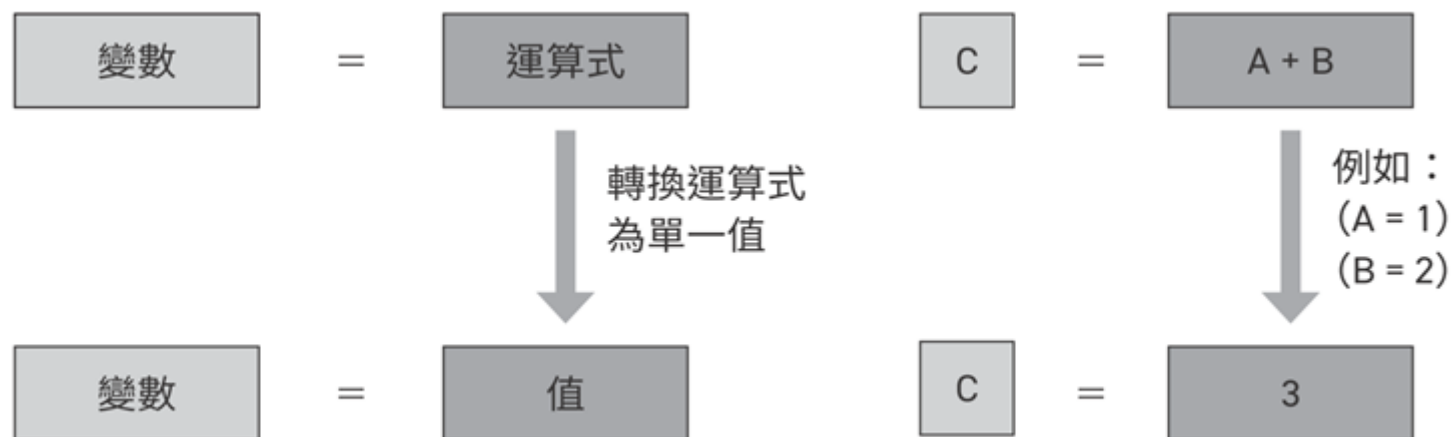
程式運算：算術運算、互動運算

3

變數資料型態

# 變數的命名與指派

- 變數是透過等號來設定內容值。
- 變數名稱是放在等號左邊。
- 變數值是放在等號右邊。



# 建立變數(初始化) 與修改變數

第一次使用變數時必須用等號指派變數一個明確的值，  
這個動作叫作初始化。

- $a = 1$

變數初始化之後才可以重新指派其他的物件變數。

- $b = a * 2$



# 變數命名規則

- 變數名稱可以是數字大小寫英文字與底線。
- 不建議使用中文或他國文字如日、韓文。
- 大寫和小寫字母是不一樣的。
- 變數名稱開頭不可為數字。
- 變數名稱不可為保留字：
  - 不同顏色來顯示保留字。
  - 保留給程式語言專用的。

## 保留字

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

# 變數命名範例

變數名稱	是否正確	理由
filename		
file_name		
file&name		
giveMe5		
giveMe-5		
_8i		
9_8i		
姓名		
True		
print		

# 資料型態

Python 的資料型態主要分為以下三種：

- 數值型態：整數(int), 浮點數(float)
- 字串型態：字串(string)
- 容器型態：陣列(List)、集合(Set)、字典(Dict)、元組(Tuple)

# 資料型態- 字串

單引號或雙引號的目的就是用來包括文字。

- 注意：字串不可同時使用單引號與雙引號。(範例：name = 'Anna')
- 把雙引號括在外面這樣單引號就會當作字串的一部分。

範例： `print("What's your name?")`

符號	說明	範例
<code>\n</code>	換行	<code>print('Hi\nAnna')</code>
<code>\t</code>	一個Tab	<code>print('Hi\tAnna')</code>
<code>\'</code>	單引號	<code>print('What\'s your name?')</code>
<code>\"</code>	雙引號	<code>print('What\'s your name?')</code>

# 資料型態的範例

變數的資料型態將會影響資料內容、記憶體儲存空間、運算操作。

- 整數 (integer)：12
- 浮點數(小數，float)：12.0
- 字串 (string)："Hello, Anna" , "12" , “12.0”

Python 程式碼	說明	運算結果
<code>print(12 * 2)</code>	整數乘法，得到整數結果	整數：24
<code>print('12' * 2)</code>	字串乘法，得到字串結果	字串：'1212'
<code>print(12 + 2)</code>	整數加法，得到整數結果	整數：14
<code>print('12' + 2)</code>	字串不支援和整數相加，翻譯失敗	
<code>print('12' + '2')</code>	字串加法，得到字串結果	字串：'122'

# 資料型態的確認與轉換

- 使用 `type()` 函式來確認物件的型別。
- 自行轉換物件的型別：`float(12)` 將整數 12 轉換成浮點數 12.0

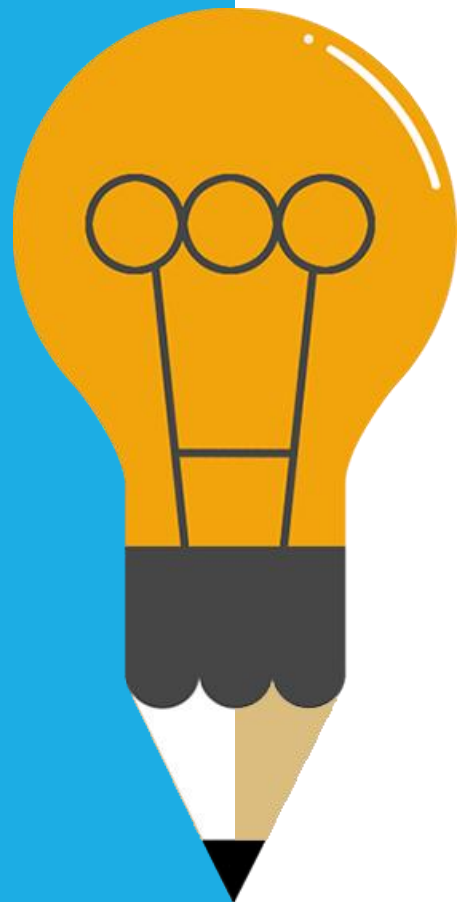
Python code	Python result
<code>type(12)</code>	<code>int</code>
<code>type(12*2)</code>	<code>int</code>
<code>type("12")</code>	<code>str</code>
<code>type("Hello, Anna")</code>	<code>str</code>
<code>print(type(12))</code> <code>print(type(float(12)))</code>	<code>&lt;class 'int'&gt;</code> <code>&lt;class 'float'&gt;</code>
<code>num_f = float(12)</code> <code>print(num_f*2)</code>	24.0

# 課堂學習：習題演練

C1\_P1：留言板

請在螢幕上輸出該訊息Hello World!

# Python 互動運算



1

互動取值

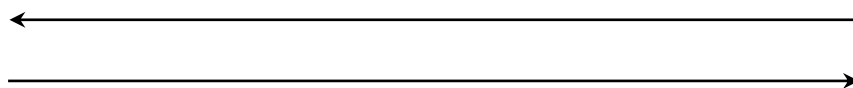
2

撰寫第一支程式

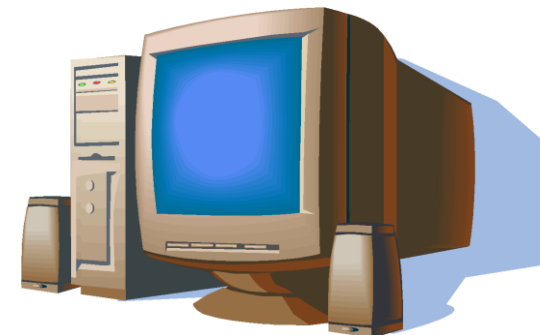


# 互動取值

- 獲取資料並運算



- 如何獲取資料？
- 如何儲存資料？
- 如何顯示運算結果？



# 寫程式注意事項：可讀性高的程式

- 名符其實的變數名稱：

使用有意義的變數名稱可以增加程式的可讀性。

無意義的變數名稱：

$a = 3.14$

$b = 2.2$

$c = a * b * b$

有意義的變數名稱：

$\pi = 3.14$

$\text{radius} = 2.2$

$\text{circle\_area} = \pi * \text{radius} * \text{radius}$

- 程式中的註解 #

將程式加上註解可以幫助其他人了解程式

# 輸出資料：print()

- print()：在螢幕中輸出變數內容與運算結果
- 語法：print(value, ..., value, sep=" ", end=" \n")

parameter	definition
value	設定欲輸出的多個值，並用逗號區隔
sep	兩個輸出值之間の間格字串，預設值為一個空白(" ")
end	結束字串，預設值為換行(\n)

# 輸出資料：print()

- print()：在螢幕中輸出變數內容與運算結果(預設會換行)
- 語法：print(value, ..., value, sep=" ", end="\n")

## code

```
x = 10
```

```
y = 20
```

```
print(x, y, x+y)
```

```
print("我", "是", "小丸子", sep="@", end="~~~")
```

```
name="Anna"
```

```
print("我", "是", name)
```

```
print("我", "是", name, sep="")
```

# 撰寫程式

- print()函式應用
- 程式範例：

	code	Result
1. 2.	<pre>print("Welcome to learn Python, ") print("Anna")</pre>	<b>程式執行結果：</b> 請輸入程式範例， 並將1,2行執行結果與3,4行執行結果的差異 以及造成原因寫到Memo
3. 4.	<pre>print("Welcome to learn Python, ", end="") print("Anna")</pre>	

# 動動腦

能否讓自己填入要顯示的名字？

例如：

Welcome to learn Python, \_\_\_\_\_.

由使用者輸入想印出的名字

輸入名字：Anna

輸出：

Welcome to learn Python , Anna.



# 讀入資料：input()

- input()：讓電腦取得從鍵盤輸入的資料
- 語法：input(“提示字串”)

提示字串其值預設是空字串。

提示字串能夠引導需要輸入那些資料。

輸入時應儘量給予清楚的提示字串。

以單引號或雙引號包起來的內容稱之為字串。

需使用變數來承接使用者輸入的資料，這個變數就裝有所輸入的資料，

否則就只有輸入但是程式卻無法使用使用者所輸入的內容。

提示字串(Prompt String)

```
x = input('請輸入一個整數值>')  
print(x)
```

```
請輸入一個整數值>520  
520
```

# 撰寫程式

`input("Prompt String")`

- 取得使用者在提示語之後，輸入螢幕的資料，而所取得之資料的型態為字串

```
x = input("請輸入一個整數值：")  
print(x)  
print(type(x))
```

```
name = input("請輸入姓名：")  
age = input("請輸入年齡：")  
print("type of name :", type(name))  
print("type of age :", type(age))  
print("大家好, 我的名字是 "+name+". 我的年齡是 "+age+" 歲。")
```



# 課堂學習：習題演練

C1\_P2：What's your name

請輸入姓名，

儲存在變數中，

並輸出**My name is (your name).**

注意：句子最後有一個半形的句點.

# 課堂學習：習題演練

C1\_P3：What's your name

請撰寫一個程式，  
接收使用者輸入兩個整數，  
並計算它們的總和，  
將輸出結果。

# 總結



Q

本章關鍵字  
本章重點概念簡述

A

請將你的章節總結寫在Memo上，  
Memo的title請設定為 **總結**

# 作業習題

## 〔 中央\_1132 〕 C1 Exercise

- C1\_E1：自我介紹產生器
- C1\_E2：字串合併
- C1\_E3：計算矩形面積

## 〔 中央\_1132 〕 C1 Advance

- C1\_E1：計算矩形面積和周長
- C1\_E2：簡易計算機
- C1\_E3：家庭介紹產生器