



# 1 python 是什麼

Python（英國發音： $/pa\ \theta n/$ ；美國發音： $/pa\ n/$ ），是一種廣泛使用的解釋型、高級和通用的編程語言。Python 支持多種編程范型，包括結構化、過程式、反射式、面向對象和函數式編程。它擁有動態類型系統和垃圾回收功能，能夠自動管理內存使用，並且其本身擁有一個巨大而廣泛的標準庫。它的語言結構以及面向對象的方法，旨在幫助程序員為小型的和大型的項目編寫邏輯清晰的代碼。— 維基百科

這有點無聊，再試一次吧。Python 是一種用來讓你看似與電影裡的主角一般，正在看著黑黑的畫面寫出能 DDoS FBI 官網的電神，但實際上你只是利用簡單的程式知識罷了。

## 1.1 編譯 vs 直譯

程式語言例子 執行過程 執行速度 錯誤檢查	編譯 (Compile)	直譯 (Interpret)
	C++	Python
	將原始碼轉換為二進位執行檔 (binary)，然後執行	將原始碼逐行翻譯成二進位後再執行
	快 編譯時檢查錯誤	慢 執行時檢查錯誤

## 1.2 電腦如何判斷陰陽頓挫

當你在讀文章時，你會怎麼判斷一個句子的結束？對你而言是句點。驚嘆號！問號？等標點符號判斷句子的結束。但對英文使用者而言是這種句點「.」而不是「。」在程式語言內，電腦要判斷每行程式（如同日常生活中的語句）也是需要標點符號的標記。如 C 語言系列以及 javascript 是用分號；以及括號對 `[]` 當作判斷依據，而 python 是使用換行以及縮排 (indent) 來判斷。換句話說，在 python 內，雖然語法較其他語言簡單，但是比起其他語言，縮排的重要性以及統一性高出許多。

# 2 python 基礎語法

## 3 Hello, World!

### 範例程式

```
1 print("Hello, World!")
```

### 先了解程式的意思吧

你可能之前就有聽說過，但在 python 內 print 不是指列印，而是指在螢幕上顯示出你所指定的某些文字

## 4 資料型態

### Python 資料型態表格

## 變數與常數

在電腦中要如何儲存資料到記憶體呢？就像搬家一樣，儲存資料需要有容器裝著才好分類管理，所以變數與常數就是資料的容器，隨時都可以拿出來取用。

- 變數：在執行過程中可以改變內容物的容器
- 常數：一旦存進容器後只能聽當標本，在執行中無法更改（例如 ）

## 「變數」是什麼？

變數是程式中很重要的一部份。簡單來說，就是「內容可以改變」的數，可以做為資料的容器。任何的內容型態，像是數字、文字、甚至是一串資料，都可以裝在變數裡面。

變數常出現在數學公式，例如位置、長度就是數字類的變數。

### 4.1 資料型態

資料有很多種類型，不同種可能是可以裝不同的東西，例如數字與字元；或是可以容納的大小不同，可以減少需要的資源。

## 為什麼要區分資料型態？

區分出不同的資料類別可以讓電腦在運算的時候，知道它裡面的資料是什麼。這可以確保在需要數字的時候不會跑出字元，不同的資料在使用同意種運算符號的時候，可能會有不同的結果

### 4.2 變數名稱

變數的取名是非常重要的，如果隨便取名的話，不只是其他人會看不懂，甚至自己過個幾天也會忘記這個變數代表什麼。

## 變數的命名

通常，如果

## 5 運算子

運算子可以對資料和變數進行操作，

## 算數運算子

算數運算子包含了常見的數學運算，例如：加法、減法、乘法、除法、次方、取餘數、取商數。

名稱	運算子	例子	結果
加法	+	x = 5 + 3	x = 8
減法	-	x = 5 - 3	x = 2
乘法	*	x = 5 * 3	x = 15
除法	/	x = 5 / 3	x = 1.66...
次方	**	x = 5 ** 3	x = 125
取餘數	%	x = 5 % 3	x = 2
取整除	//	x = 5 // 3	x = 1

## 賦予運算子

賦予運算子可以改變在它左邊的變數的內容，全部都有等號，是從其他運算子衍伸而來。

名稱	運算子	例子 (x=7)	結果
加法	+=	x += 2	x = 9
減法	-=	x -= 2	x = 5
乘法	*=	x *= 2	x = 14
除法	/=	x /= 2	x = 3.5
次方	**=	x **= 2	x = 49
取餘數	%=	x %= 2	x = 1
取整除	//=	x //= 2	x = 3

除了這幾個以外，還有一些賦予運算子會在之後提到。

## 6 邏輯判斷

### 邏輯判斷

#### 6.1 比較運算子

比較運算子可以比較兩個數字的大小，然後輸出是不是符合運算子的關係。比較運算子包含了數學裡面的等號、各種不等號。

名稱	運算子	例子 (x=7)	結果
等於	==	x 2	
不等於	!=	x 2	
大於	>	x 2	
小於	<	x 2	
大於或等於	>=	x 2	
小於或等於	<=	x 2	

#### 6.2 邏輯運算子

邏輯運算子可以把

## 7 迴圈

有時候，我們會需要重複執行一些一樣或非常相似的程式碼，可以使用迴圈來執行。使用迴圈不只可以寫得更快，也能讓程式更加簡潔易懂，便於維護。以下是兩種迴圈的介紹：

## 7.1 for 迴圈

for 迴圈通常是用在知道一個範圍的情況下，  
用法是：

for 變數 in 一個範圍/一個物件: 一段程式碼

## 7.2 while 迴圈

## 7.3 退出迴圈

# 8 函式

## 8.1

# 9 類別

## 9.1