

# Python

理論與實作

黃熠程

2022/01/15

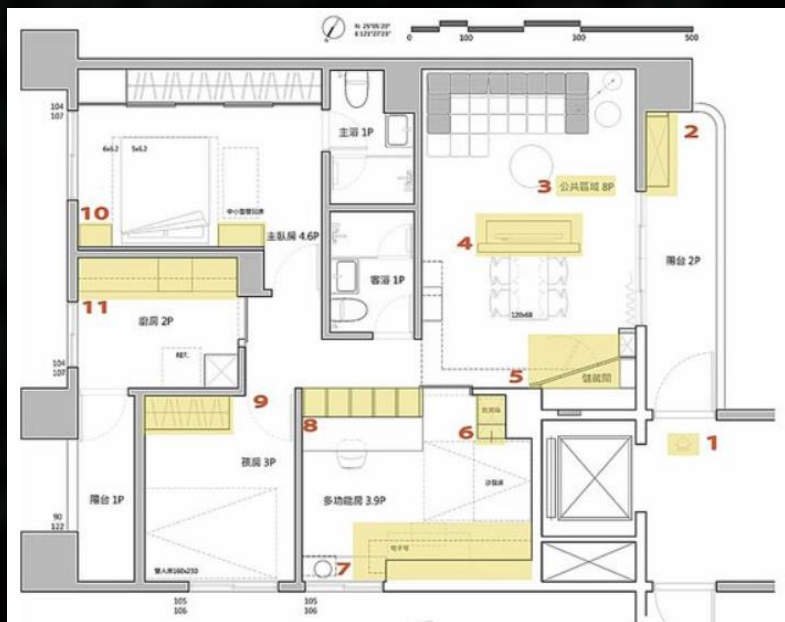
# Why Python

- Easy to Learn
- Open Source & Free
- Cross Platform
- Many Lib (Module)

第三方套件	功能
Pandas	數據處理與視覺化
Numpy	高效線性代數運算
Matplotlib	視覺化處理
SciPy	科學計算
BeautifulSoup	XML, HTML 解釋器
PyQt, tkinter	GUI APP
Requests	存取網頁資料
Django, Flask	架設網站
PyMySQL	資料庫
PyGame	多媒體與遊戲開發
Scikit-learn	機器學習
TensorFlow, PyTorch	深度學習

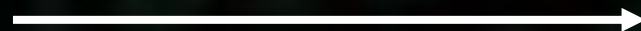
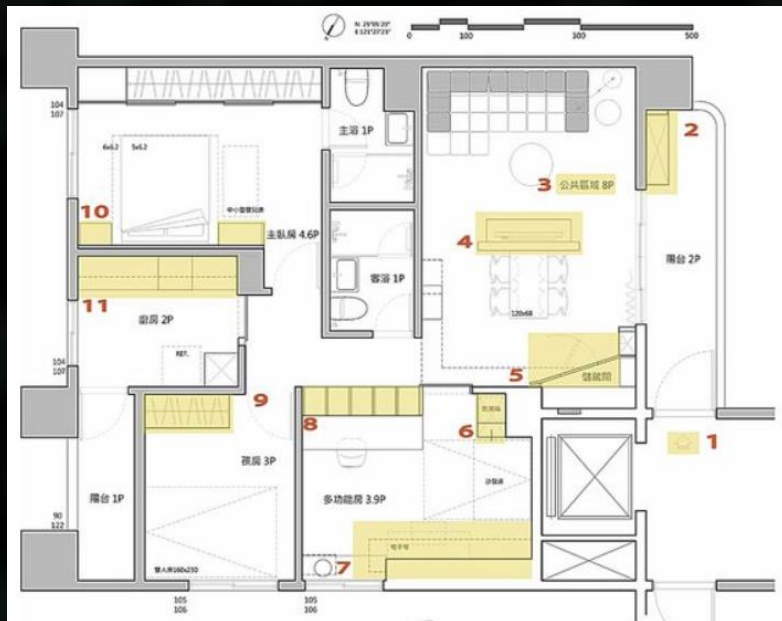
# 今日目標

- 物件導向
- 函式
- 類別
- 使用套件
- 邏輯判斷



想像一下，你手上有一個設計藍圖，你可以根據這個藍圖，蓋出任意外觀的房子





類別  
class



物件  
object



人人果實

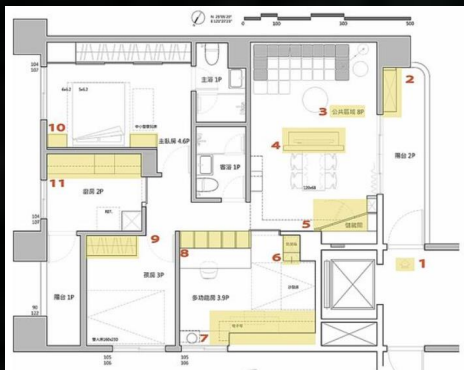


類別  
class



物件  
object

類別(class)



物件(object)



物件(object)



...

## Python 內建有三種基本類別

integer	整數1號	整數2號
float	浮點數1號	浮點數2號
string	字串1號	字串2號
...	...	...

...

...

...

...

# Python 資料型態

類別(class)





# Python 資料型態

類別(class)

integer

物件(object)

integerObj

描述：value

10

宣告

C/C++  
C#  
Java

```
Integer integerObj = 10
```

javaScript

```
var integerObj = 10
```

python

```
integerObj = 10
```

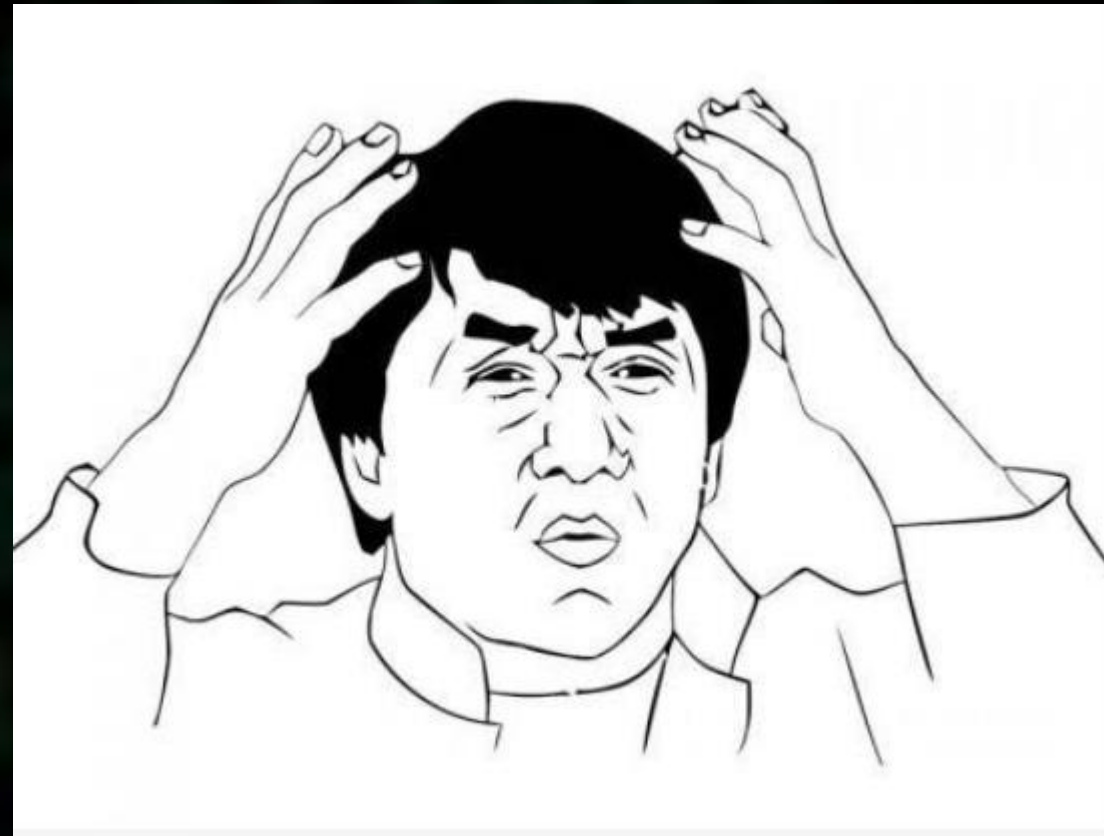
動態宣告

沒有動態宣告的狀況...

```
integer,parameter :: NX=200,NY=200,NZ=30
integer, parameter :: dx = 1000,dy = 1000, dz = 250
real, parameter :: rlatd = 22.25, rlond = 119.75
real, parameter :: lon_l = 118.29151, lat_l = 20.900108
real, parameter :: lon_g = 9.72325634E-03, lat_g = 8.99927784E-03

!~background
  real :: u0(NX,NY,NZ),v0(NX,NY,NZ),qv0(NX,NY,NZ),t0(NX,NY,NZ)

!~ascat
  integer,parameter :: sea_info=358
  real :: sea_site(3,sea_info)
  integer :: sea_site_id(sea_info)
  character*13 :: infile_sea(sea_info)
  !
  INTEGER :: ID,YY,MM,DD,HH,NN,itime
  REAL :: pre,temp,td,RH,WIND,WINDIR
  real :: alt(sea_info),lon(sea_info),lat(sea_info)
  real :: spd(sea_info),dir(sea_info)
  !
  real,parameter :: badpt = -999.98999
  real :: ua(sea_info),va(sea_info)
  real,parameter :: u_hfactor=-7.7368259E-02
  real,parameter :: v_hfactor=0.612216
  real,parameter :: u_mfactor=-0.6058777
  real,parameter :: v_mfactor=-0.1411584
  real,parameter :: u_lfactor=6.8603404E-02
  real,parameter :: v_lfactor=0.1272687
```





C++之父: Bjarne Stroustrup



Java之父: James Gosling



PHP之父: Rasmus Lerdorf



Python之父: Guido van Rossum

1024.com

1024.com

# Python 資料型態

類別(class)	物件(object)	描述：value
integer	整數1號	10
float	浮點數1號	10.567
string	字串1號	"hello"
...	...	...
SomthingClass	Object1	Any Value

```
A = 10  
B = 10.567  
C = "hello"  
...
```



# Python 數學計算

數學符號	功能
+	加法
-	減法
*	乘法
/	除法
//	只取得整數的除法
%	取餘數

不同類別不能混在一起計算

Using Colab to check python variable type ...

[Sample0201](#)

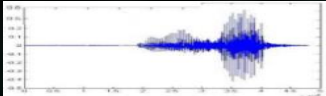
# 函数(Function)

# 在人工智慧中我們要找一個function...


=方法

=模型


- Speech Recognition

$f(\text{  }) = \text{'How are you'}$

- Image Recognition

$f(\text{  }) = \text{'Cat'}$

- 五子棋

$f(\text{  }) = \text{'5-5' (next moving)}$

# Python 的函式

函式



函式





# Python 的自訂函式

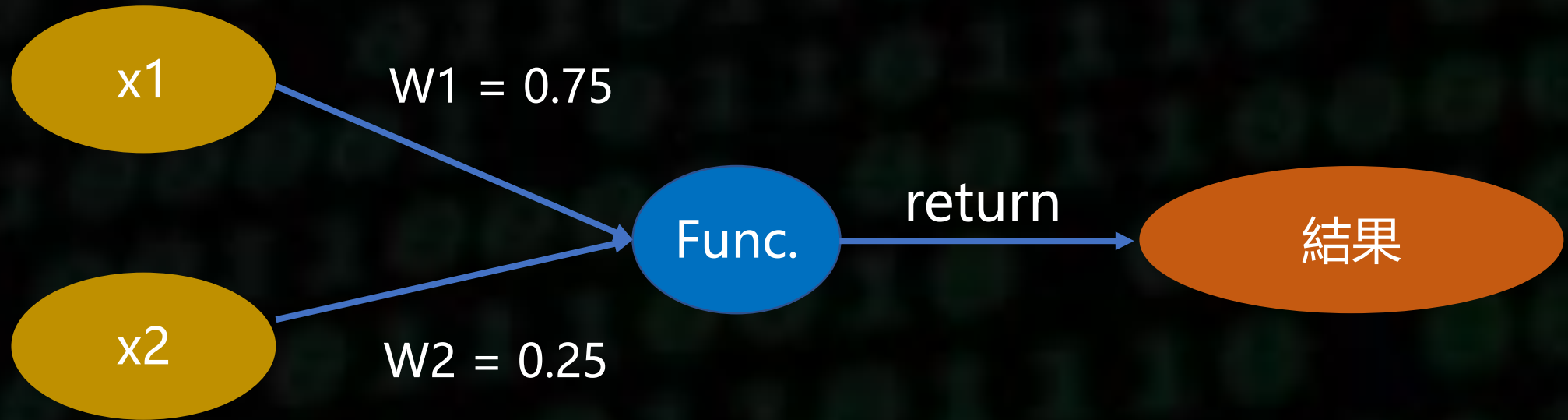
```
def 函式名稱(input something):  
    ....  
    return something
```

```
def 函式名稱():  
    ....
```

Using Colab to create your own function ...

[Sample0202](#)

# Practice



$$X_1 * W_1 + X_2 * W_2 = ?$$

# 自訂類別

類別(class)



物件(object)



描述：value

500 坪

客廳多大?  
幾個廁所?  
幾個門?

描述太少了.....



類別(class)  
類別(type)

integer

物件(object)

integerObj

描述：value

10

類別(Class)

integer

integer

integer

屬性(properties)

客廳大小

廁所數量

門數量

100

4

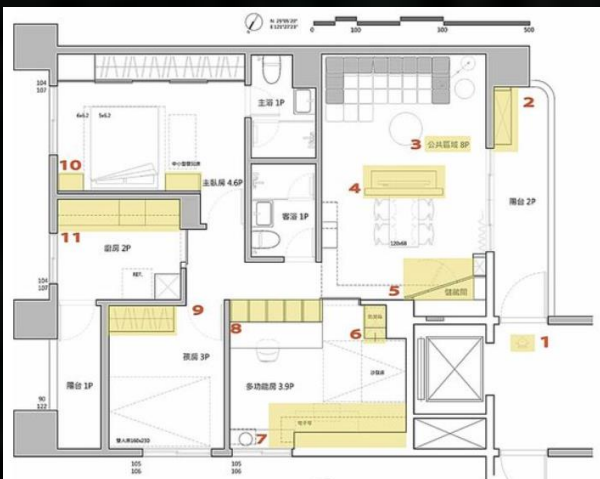
10



Using Colab to check python Class...

[Sample0203](#)

# Python Class



在這設計藍圖中，有“客廳”，“廚房”...等特徵，  
程式設計中，我們稱為 **屬性**(properties)或

我們還能定義一些**行為**(behaviors)，例如打開門，開啟水龍頭，  
這些行為在所有的房子(物件)中都能用到



套房A.開門()



套房B.開門()

=方法

=模型

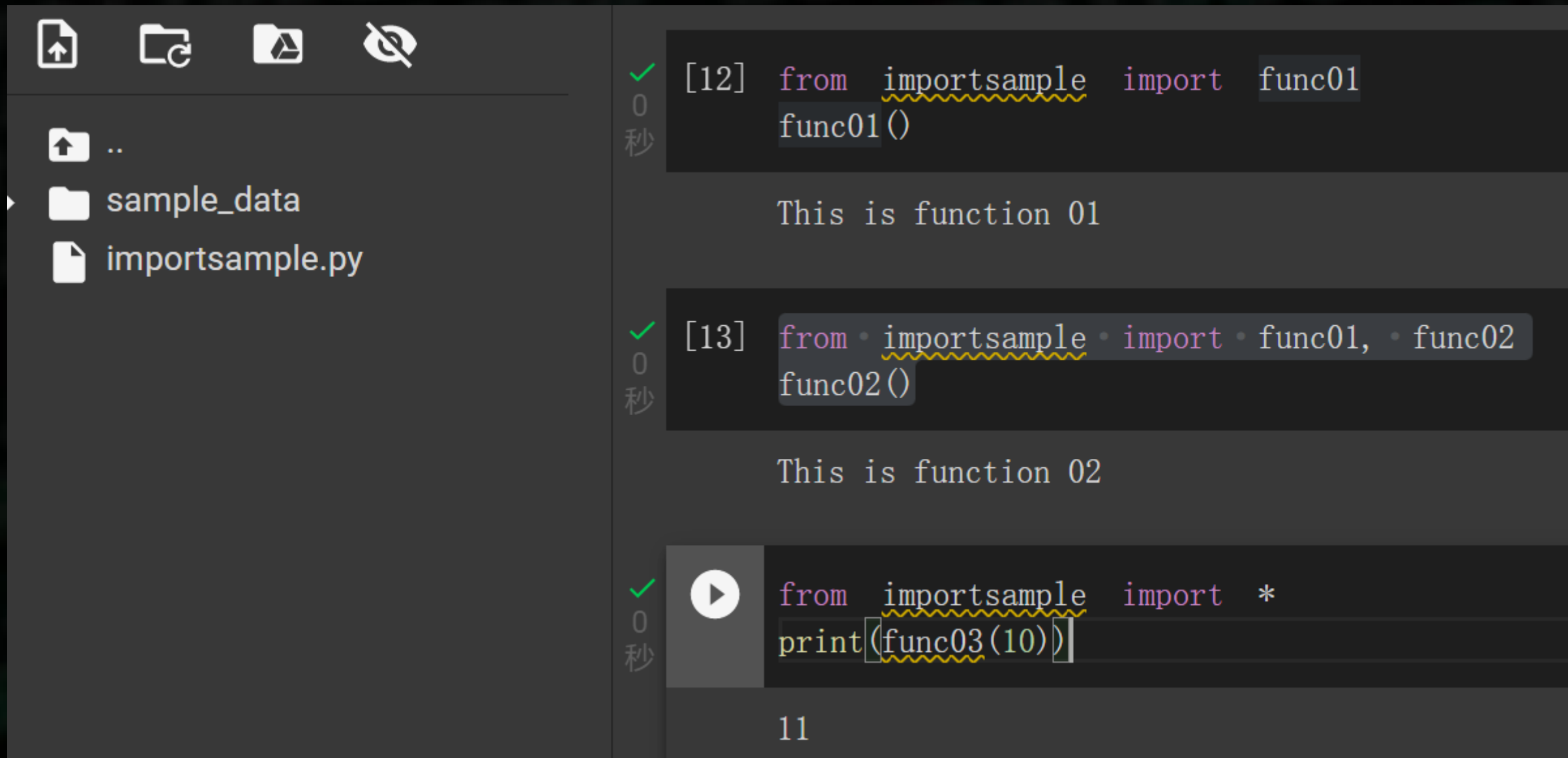
Using Colab to check python Class...

[Sample0204](#)

**使用套件**



# 引入自定義套件



The image shows a Jupyter Notebook interface. On the left is a file explorer with icons for file operations and a list of files: '..', 'sample\_data', and 'importsample.py'. The main area displays three code cells, each with a green checkmark and '0 秒' (0 seconds) execution time.

**Cell 12:** Imports `func01` from `importsample` and calls it.

```
[12] from importsample import func01
      func01()
```

This is function 01

**Cell 13:** Imports `func01` and `func02` from `importsample` and calls `func02`.

```
[13] from importsample import func01, func02
      func02()
```

This is function 02

**Cell 14:** Imports all functions from `importsample` and calls `func03` with the argument 10.

```
from importsample import *
print(func03(10))
```

11

# 引入第三方套件

[Sample0205](#)

✓  
0  
秒

```
[25] from datetime import date  
d1 = date(2022, 1, 1)  
print(d1)
```

2022-01-01

✓  
0  
秒

```
▶ import datetime  
d1 = datetime.date(2022, 1, 1)  
print(d1)
```

☐→ 2022-01-01

✓  
0  
秒

```
▶ import numpy as np  
a = np.array([1, 2, 3])  
print(type(a))  
print(a)
```

<class 'numpy.ndarray'>  
[1 2 3]

延伸閱讀: [When Use 'import' and When 'from import'](#)

# Numpy

# 向量

var1	var2	var3	var4	var5	var6
------	------	------	------	------	------

編號0      編號1      編號2      編號3      編號4      編號5

# 矩陣

var1	var2	var3	var4	var5	var6
var7	var8	var9	var10	var11	var12
var13	var14	var15	var16	var17	var18

Row(列)

Column (行)

# Numpy 基本介紹

- 基本資料型態 array，可以用來存放與處理多維資料

`<class 'numpy.ndarray'>`

- `numpy.ndarray` 建立時可指定元素型態，`dtype`可顯示跟指定元素型態

```
a = np.array([1, 5, 9], dtype = 'int64')
print(a.dtype)

int64
```

- `shape()`方法可以得知每一維度的個數，進一步推算 row 與 column

```
a = np.array([1, 5, 9], dtype = 'int64')
print(a.shape)
b = np.array([[1, 5, 9], [2, 6, 10]], dtype = 'int64')
print(b.shape)

(3, )
(2, 3)
```

1	5	9
2	6	10

矩陣方向都是  
先列後行

Using Colab to check python Class...

[Sample0206](#)



# 矩陣切片 [AIworkshop0205](#)

矩陣方向都是  
先列後行

- `print(c[:, 0])` # 指定第0行,所有列

1	2	3
4	5	6
7	8	9

```
print(c[:, 0]) # 指定第0行,所有列  
print(c[:, 1]) # 指定第1行,所有列  
print(c[:, 2]) # 指定第2行,所有列
```

```
[1 4 7]  
[2 5 8]  
[3 6 9]
```

- `print(c[:2, :2])` # 指定第0~1列, 第0~1行

1	2	3
4	5	6
7	8	9

```
print(c[:2, :2]) # 指定第0~1列, 第0~1行
```

# 流程控制

# Python 的流程控制

If (邏輯判斷a):  
描述 ...

elif(邏輯判斷b):  
描述 ...

else:  
描述 ...

## 邏輯判斷運算

==

判斷是否"等於" (equivalent)

!=

判斷是否"不等於" (not equivalent)

>

判斷是否"大於" (greater than)

>=

判斷是否"大於或等於" (greater than or equivalent)

<

判斷是否"小於" (less than)

<=

判斷是否"小於或等於" (less than or equivalent )

## 判斷集合的運算

and 兩判斷式皆為真.AND.才為真

or 兩判斷式至少一個為真.OR.才為真

Using Colab to check python if statement...

[Sample0207](#)

# Python 的流程控制

```
for iterating_var in sequence:  
    statements(s)
```

Using Colab to check python if statement...  
[Sample0207](#)

# Summary

- Python中所有變數皆是物件，其型別由等號右邊決定