



Python

類別

賴璉錡

lclai.t11@o365.fcu.edu.tw



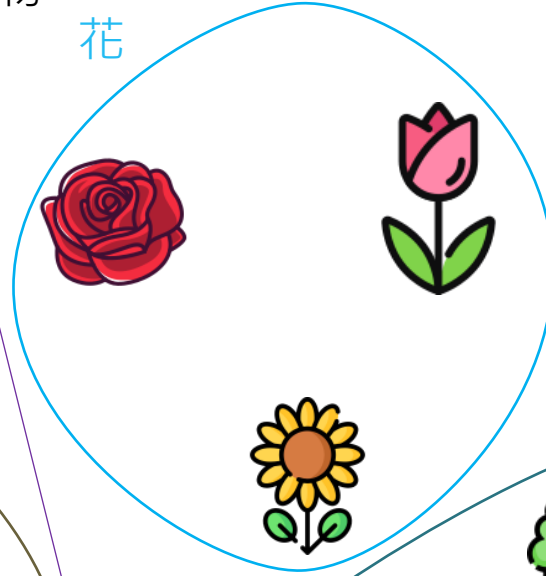
生物

動物 植物

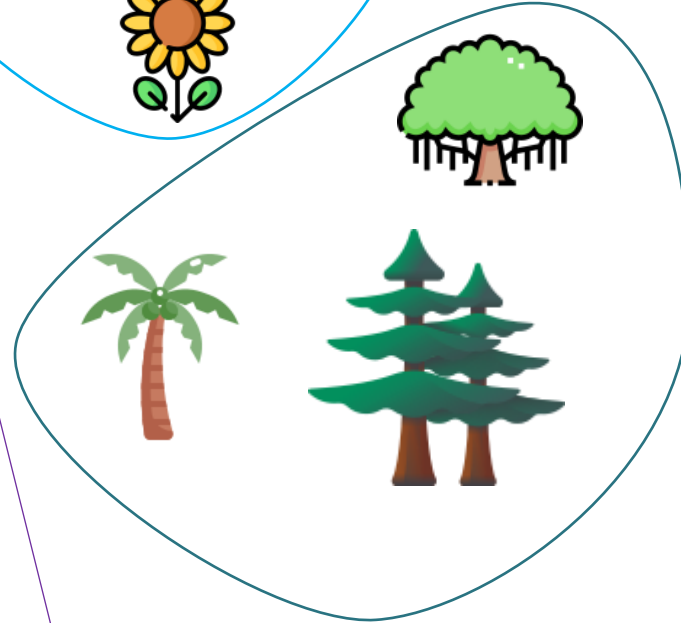
狗



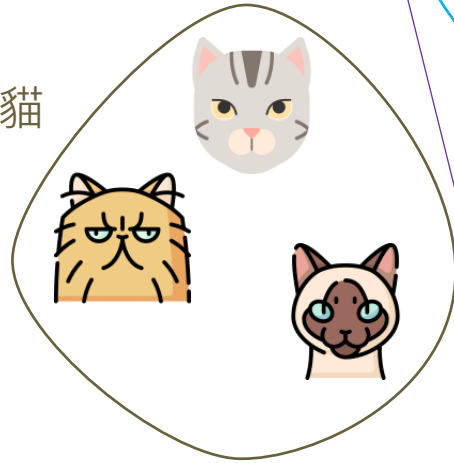
花

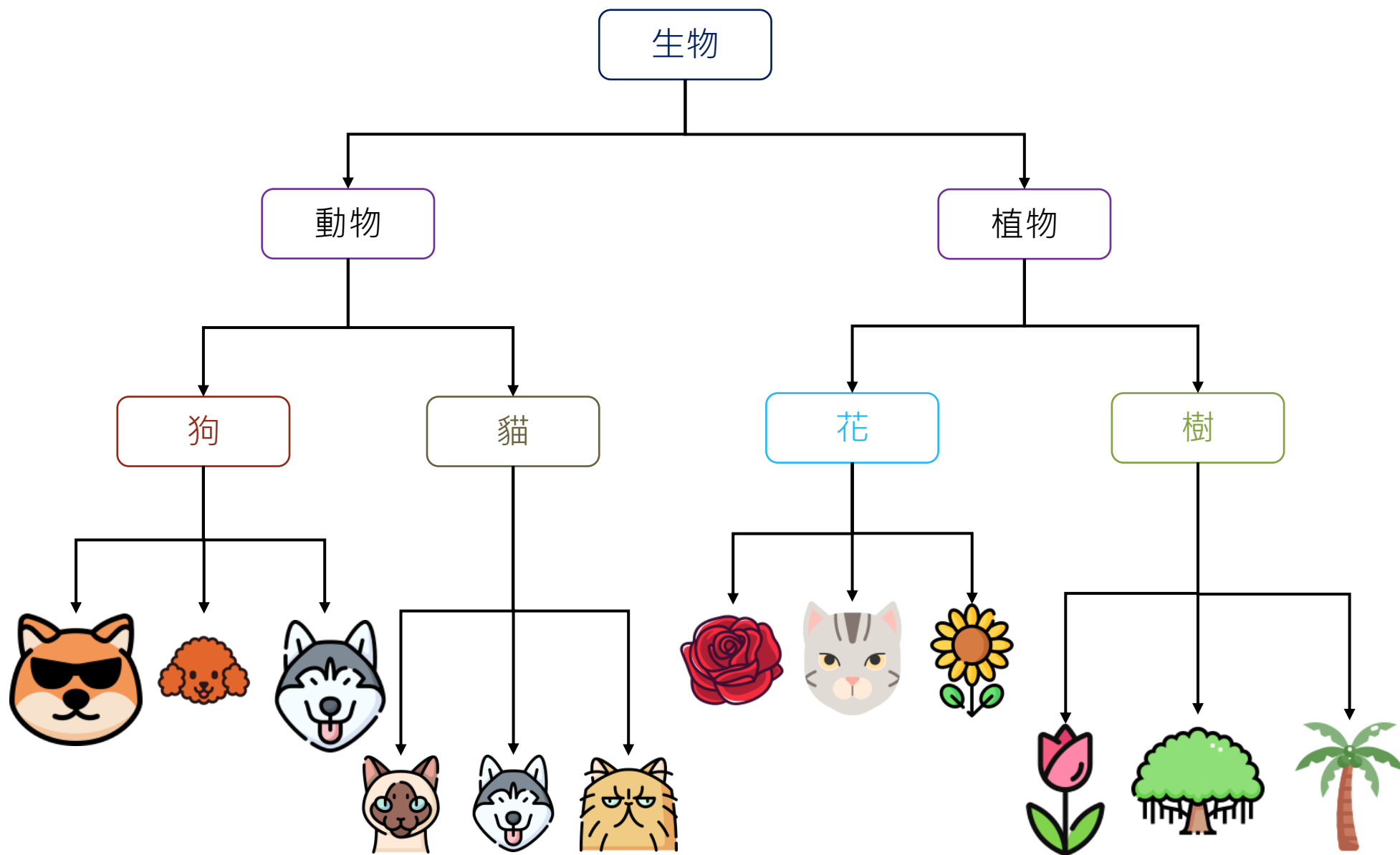


樹



貓





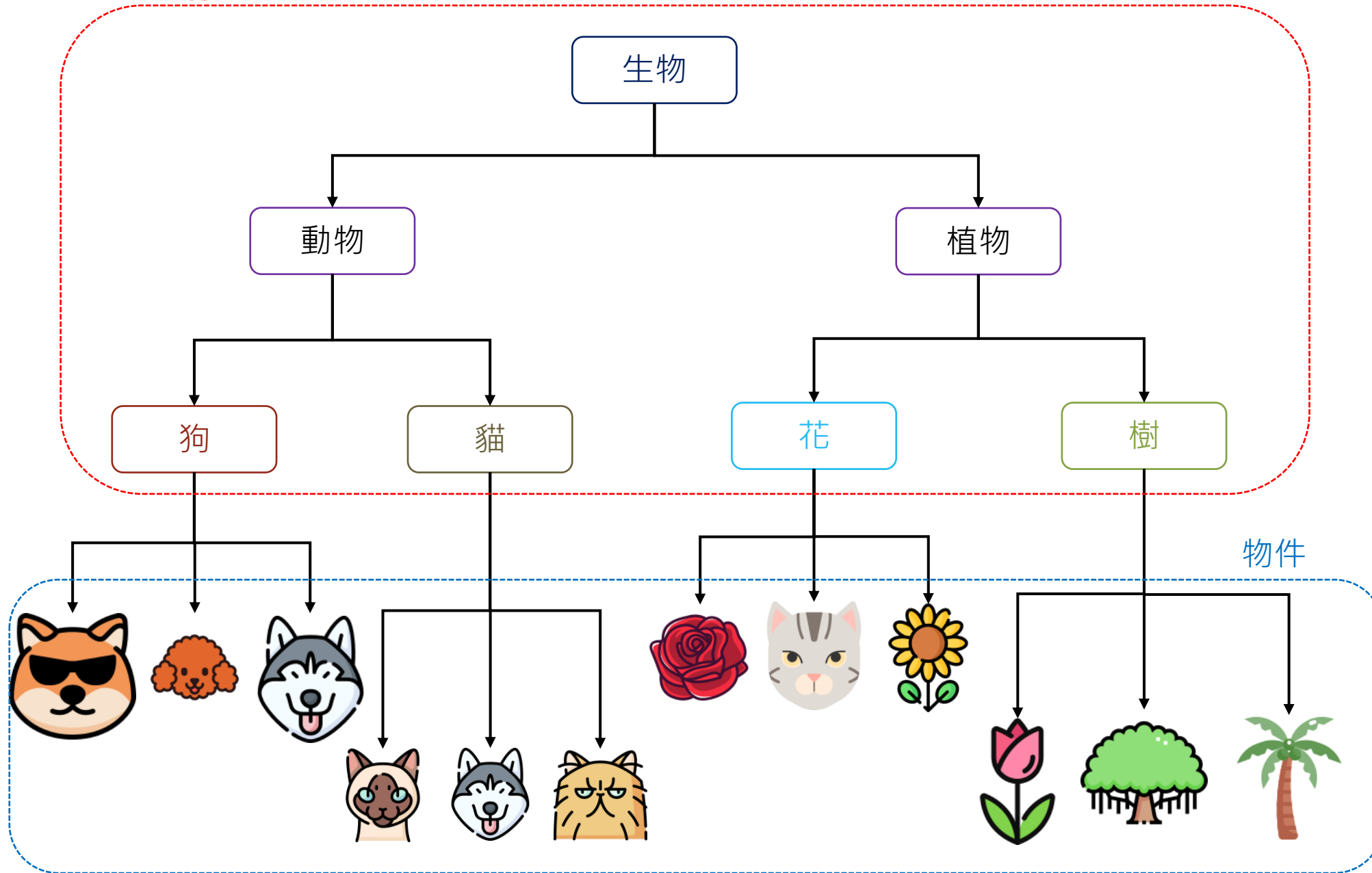
類別(Class)

- 它描述了一組具有相同屬性和行為的物件(Object)，可以說是物件的設計圖。
- 類別定義了物件的
屬性：可以是數字、字串、布林值或其他資料型態。
方法：可對物件進行操作的函數。

物件(Object)

- 透過類別(Class)實際建立的實體，依設計圖製造出來的實體物品。

類別



類別

類別名稱

建構式

方法: print毛髮
顏色

屬性: 顏色

屬性: 叫聲

屬性: 體型

方法: 回傳叫聲

```
class Dogs:
    def __init__(self, color, sound, size):
        self.color = color
        self.sound = sound
        self.size = size

    def get_color(self):
        print("My hair's color is", self.color, ".")

    def bark(self):
        return self.sound
```

類別的使用

透過Dogs類別新增一個husky物件

方法: print
毛髮顏色

方法: 取得叫聲

屬性: 體型

```
class Dogs:
    def __init__(self, color, sound, size):
        self.color = color
        self.sound = sound
        self.size = size

    def get_color(self):
        print("My hair's color is", self.color, ".")

    def bark(self):
        return self.sound

husky = Dogs("white and black", "Woof", "medium")

husky.get_color()
dog_bark = husky.bark()
print(dog_bark)
print("This dog's size is", husky.size)
```

```
My hair's color is white and black .
Woof
This dog's size is medium
```


Practice: Points類別

- 請宣告一個Points類別：
 - 屬性: `x1, y1`
 - 方法:
 - `distance(self, point_x, point_y):`
 - 印出該點與 `(point_x, point_y)` 之距離
 - $dis = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$

```
class Points:

p = Points(3, 4)
print(p.distance(0, 0))
```

Practice: Pets類別

- 請宣告一個Pets類別：

- 屬性: `name`, `color`, `age`

- 方法:

- `get_info()`: 同時印出`name`, `color`, `age`

- 利用`list`與`index`方式新增物件，並放入`pets_list`

- `pets_list = []`

- `p = {`

- `{"name": "Buddy", "color": "black", "age": 3},`

- `{"name": "Max", "color": "brown", "age": 10},`

- `{"name": "Molly", "color": "gray", "age": 10},`

- `}`

```

class Pets:
    def __init__(self, name, color, age):
        self.name = name
        self.color = color
        self.age = age

    def get_info(self):
        return f'{self.name} {self.color} {self.age}'

p = [{"name": "Buddy", "color": "black", "age": 3},
      {"name": "Max", "color": "brown", "age": 10},
      {"name": "Molly", "color": "gray", "age": 10}]

pets_list = []
for pet in p:
    pet_obj = Pets(pet["name"], pet["color"], pet["age"])
    pets_list.append(pet_obj)

print(pets_list)
for p_c in pets_list:
    p_c.get_info()

```

```

[<__main__.Pets object at 0x014DE430>, <__main__.Pets object at 0x014DE0A0>, <__main__.Pets object at 0x014DE4D8>]
Buddy black 3
Max brown 10
Molly gray 10

```

Practice: 銀行帳戶

- 請宣告一個BankAccount類別：
 - 屬性:balance (餘額)
 - 方法：
 - deposit(self):向帳戶存入指定金額的錢，返回更新後的帳戶餘額。
 - withdraw(self, amount):從帳戶中取出指定金額的錢，返回更新後的帳戶餘額。

Practice: 骰子

- 屬性：
 - 面數
 - 數值
 - 歷史紀錄
- 方法：
 - roll: `random.randint(1, self.sides)`
 - `get_last_roll`
 - `get_roll_history`
 - `get_average`

import用法

```
main.py x shape.py x
1 class Triangles:
2     def __init__(self, bottom, height):
3         self.bottom = bottom
4         self.height = height
5
6     def get_area(self):
7         return self.bottom * self.height / 2
```

從 shape 引入 Triangles

```
Project pythonProject3 C:\Users\se
main.py shape.py
1 from shape import Triangles
2
3 tri1 = Triangles(2, 5)
4
5 print(tri1.get_area())
6
```

```
Project pythonProject3 C:\Users\se
main.py shape.py
1 from shape import Triangles
2
3 tri1 = Triangles(2, 5)
4
5 print(tri1.get_area())
6
```

引入整個檔案，必須
指定類別名稱

Practice : 引用檔案

- 請建立shapes.py，並建立以下class
 - Circle:
 - 屬性: r(半徑)
 - 方法: get_area(self): 回傳計算圓面積($\pi=3.14$)
 - Triangle():
 - 屬性: bottom(底)、height(高)
 - 方法: get_area(self): 回傳計算三角形面積
 - Rectangle():
 - 屬性: width(底)、height(高)
 - 方法: get_area(self): 回傳計算四邊形面積
- 建立一個cal_area.py檔案
- 分別計算以下面積
 - 圓: r=15
 - 三角形: bottom=20, height=10
 - 四邊形: weight=10、height=25

```
circle = shape.Circle(15)
print(circle.get_area())
tri = shape.Triangles(20, 10)
print(tri.get_area())
rec = shape.Rectangle(10, 25)
print(rec.get_area())
```

```
706.5
100.0
250
```

Practice: 機器人設計

- 建立一個可以自訂機器人裝備以獲得不同數值的程式

- 程式功能：

- 機器人名稱
- 裝備武器
- 裝備防具
- 顯示狀態

```
weapons = [  
    Weapon("雷射槍", 30),  
    Weapon("電漿劍", 50),  
    Weapon("左輪手槍", 40)  
]
```

```
armors = [  
    Armor("鋼鐵護甲", 20),  
    Armor("能量護盾", 40),  
    Armor("鈦合金裝甲", 30)  
]
```


HW: 探險背包

- 請設計一個背包程式，可以選擇將要帶走在探險過程中發現的物品。

- 類別：背包、物品
- 背包中可以放入相同的物品
- 重量不能超過上限

- 類別方法：

- `add_item()`
 - 在背包中放入物品
- `remove_item()`
 - 移除背包中的物品
- `show_status()`
 - 顯示背包中的物品

```
backpack = Backpack()
items = [
    Item("繩子", 2.0),
    Item("水壺", 1.0),
    Item("手電筒", 0.5),
    Item("地圖", 0.1),
    Item("急救包", 1.0),
    Item("指南針", 0.2),
    Item("帳篷", 3.0),
    Item("睡袋", 1.5)
]
```

`while True:`

```
print("請選擇操作：")
print("1. 放入物品")
print("2. 移除物品")
print("3. 查看背包")
print("4. 結束程式")
choice = input("選擇 (1-4): ")
```