Destroy Company

Jan 1, 2000

Mini Mine Store

411071114 林鈺翔

想法來源

報童問題

最大化店家預期利潤

利潤表示成：

$$Max\ pE[\min\{q, D\}] - cq \quad (q \geq 0)$$

D:不確定的需求量(Demand)為一連續值

F:D的累積分布函數

f:D的機率密度函數

p:單位銷售價格(price)

c:單位進貨成本(cost)

q:進貨數量

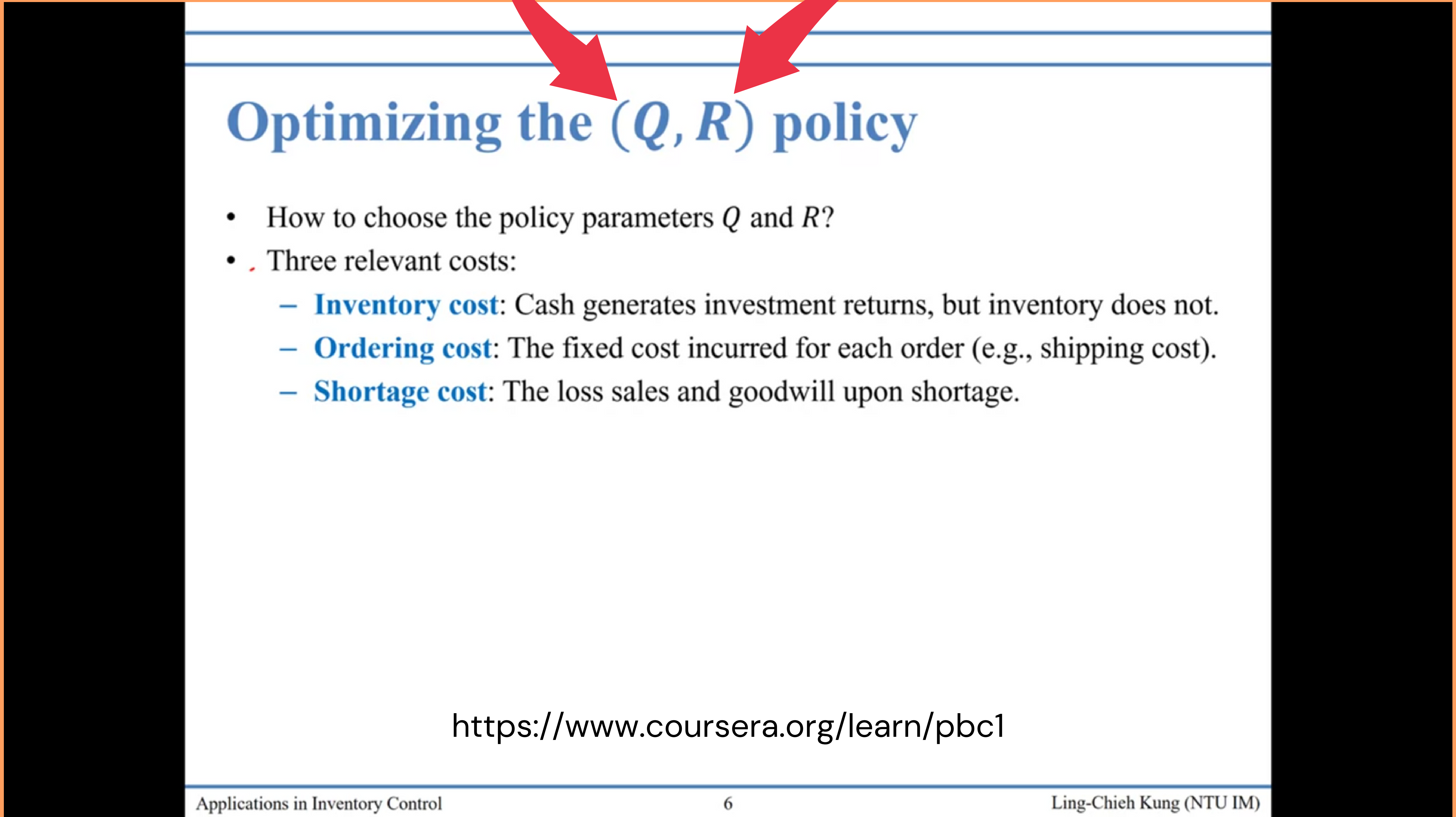
q*:最佳進貨量

C_u:售出一件產品的獲利$(p-c)$，又稱缺貨成本

C_o:未售出一件產品的損失，又稱儲藏成本

訂貨數量

標準

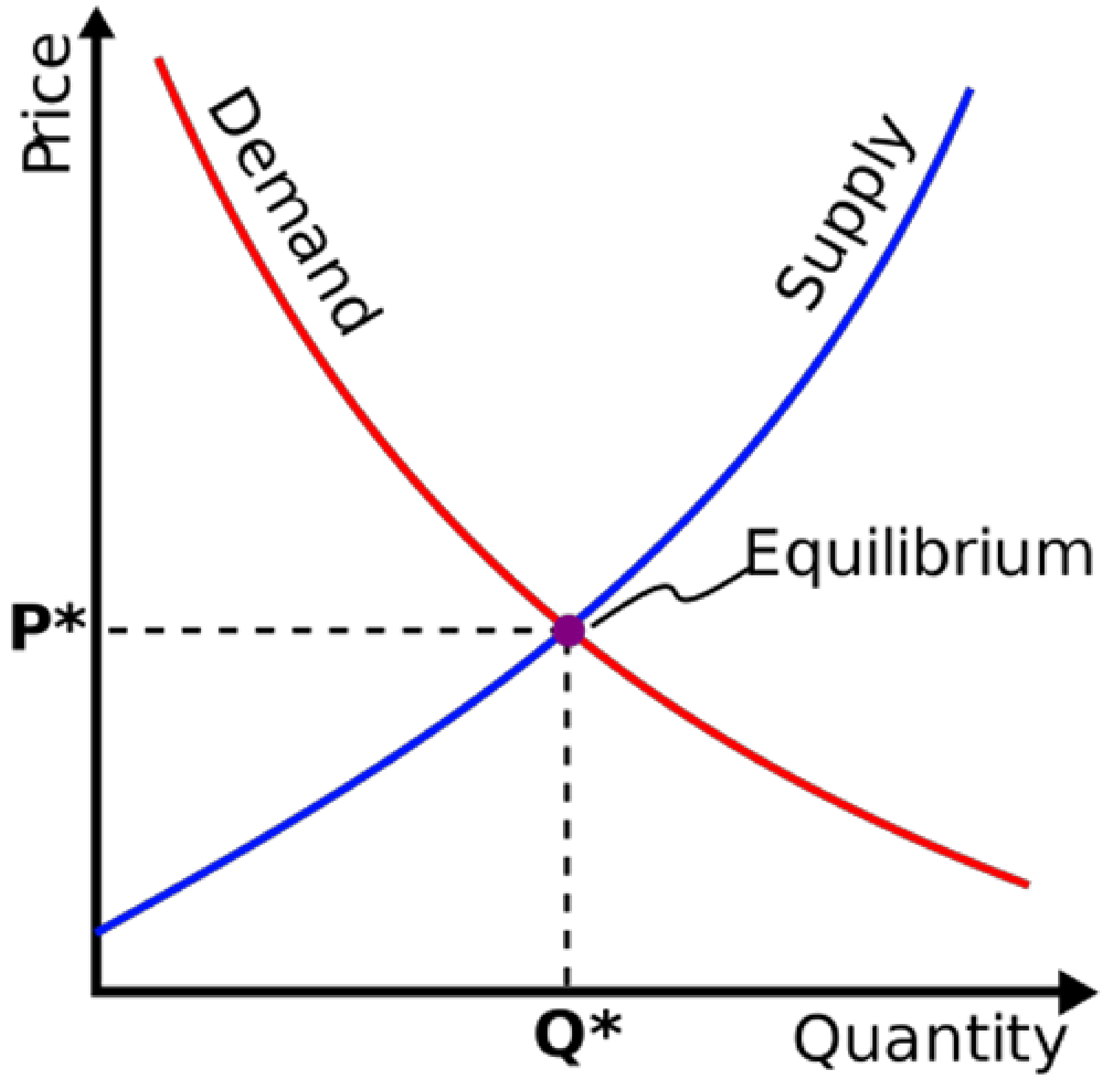


Optimizing the (Q, R) policy

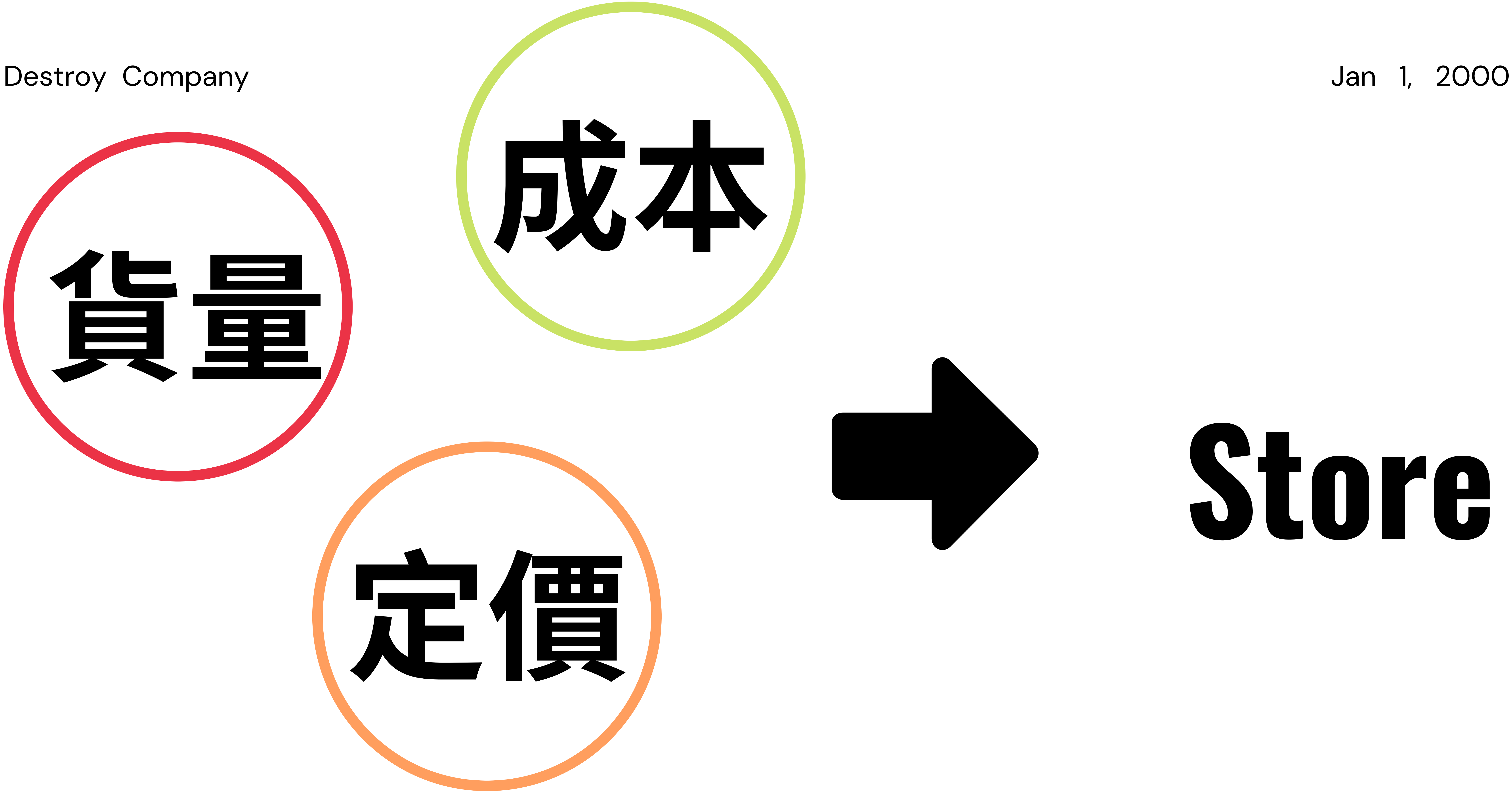
- How to choose the policy parameters Q and R?
- Three relevant costs:
  - **Inventory cost**: Cash generates investment returns, but inventory does not.
  - **Ordering cost**: The fixed cost incurred for each order (e.g., shipping cost).
  - **Shortage cost**: The loss sales and goodwill upon shortage.

https://www.coursera.org/learn/pbc1

Applications in Inventory Control 6 Ling-Chieh Kung (NTU IM)



維基百科： https://upload.wikimedia.org/wikipedia/commons/8/8c/Supplydemandequilibrium.svg

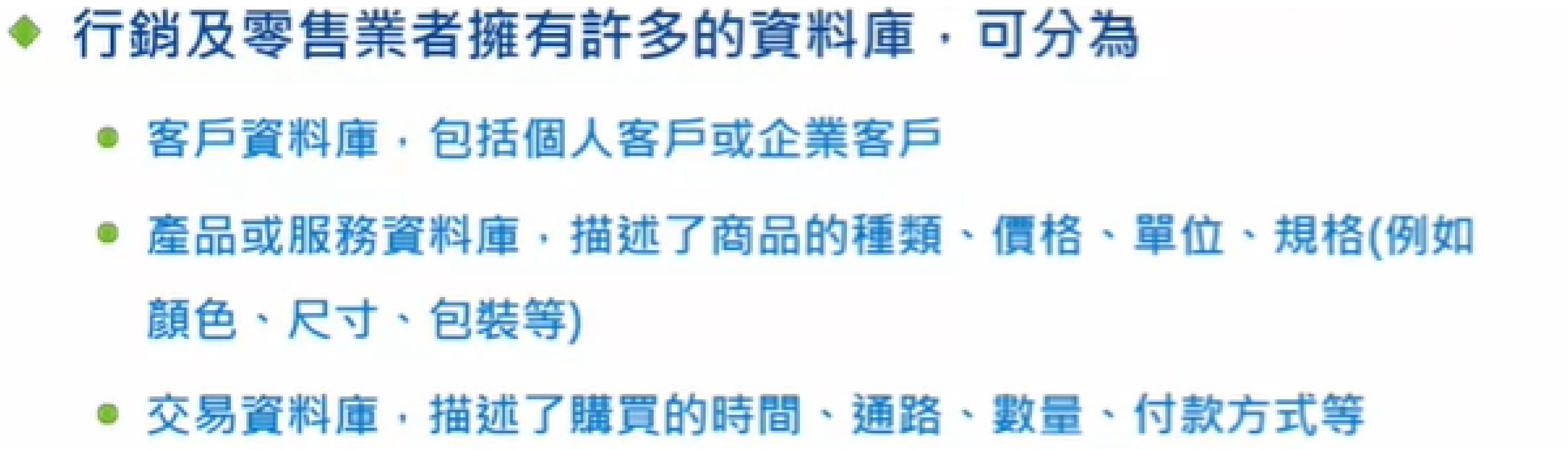


貨量

成本

定價

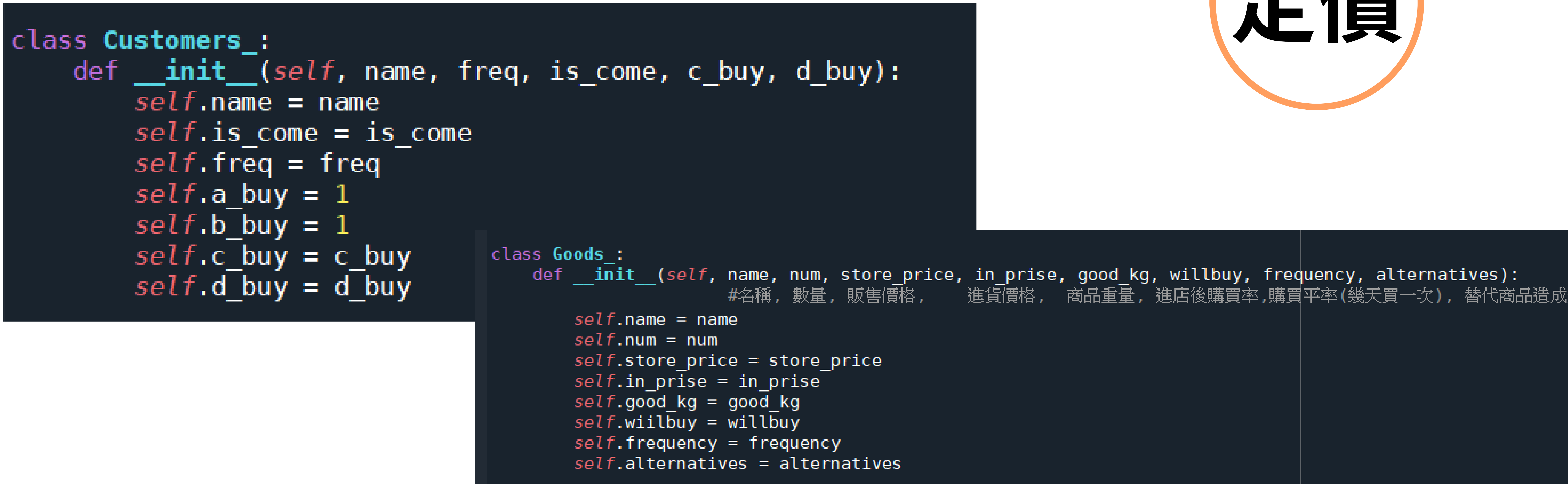
Store



- 行銷及零售業者擁有許多的資料庫，可分為
  - 客戶資料庫，包括個人客戶或企業客戶
  - 產品或服務資料庫，描述了商品的種類、價格、單位、規格(例如顏色、尺寸、包裝等)
  - 交易資料庫，描述了購買的時間、通路、數量、付款方式等

貨量

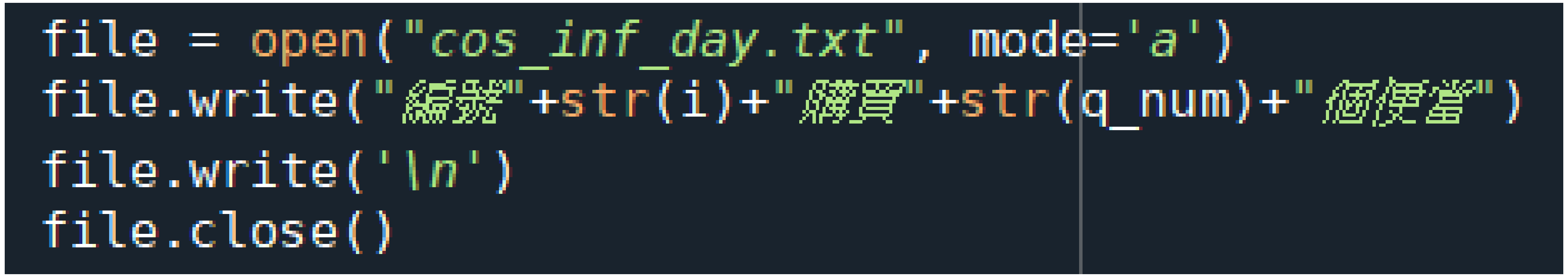
定價



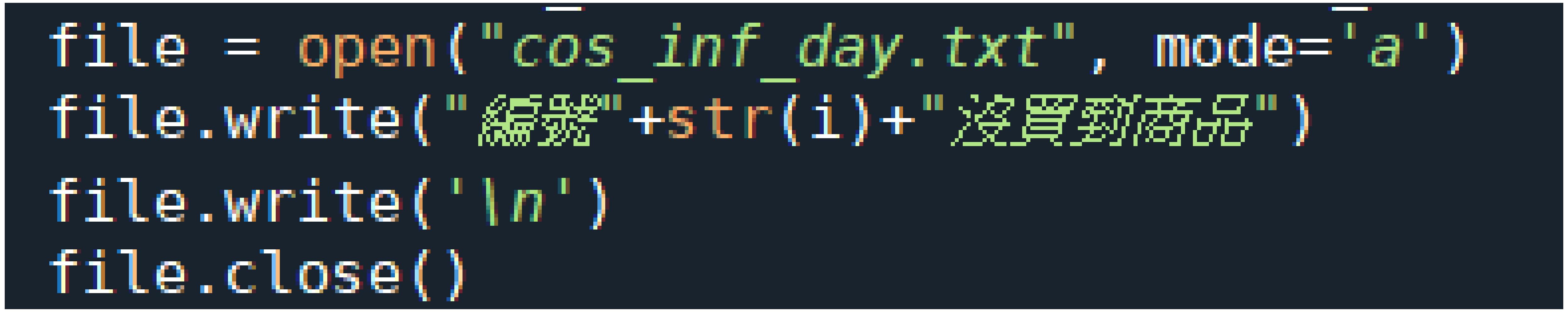
```
class Customers_:
    def __init__(self, name, freq, is_come, c_buy, d_buy):
        self.name = name
        self.is_come = is_come
        self.freq = freq
        self.a_buy = 1
        self.b_buy = 1
        self.c_buy = c_buy
        self.d_buy = d_buy
```

```
class Goods_:
    def __init__(self, name, num, store_price, in_prise, good_kg, willbuy, frequency, alternatives):
                       #名稱, 數量, 販售價格,    進貨價格,   商品重量, 進店後購買率,購買平率(幾天買一次), 替代商品造成
        self.name = name
        self.num = num
        self.store_price = store_price
        self.in_prise = in_prise
        self.good_kg = good_kg
        self.wiilbuy = willbuy
        self.frequency = frequency
        self.alternatives = alternatives
```

https://www.coursera.org/learn/bigdataanalysis

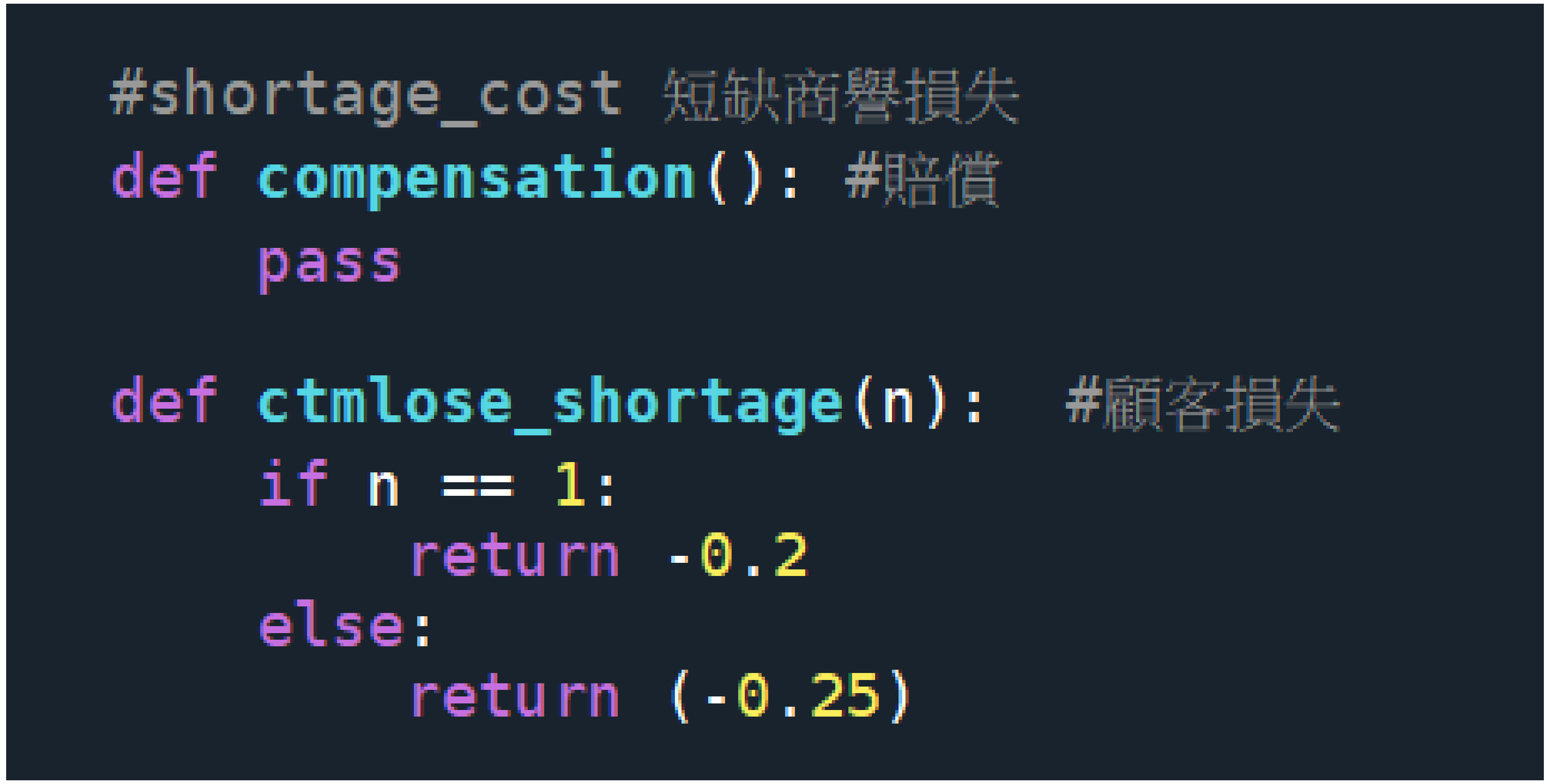


```python
file = open("cos_inf_day.txt", mode='a')
file.write("編號"+str(i)+"購買"+str(q_num)+"個便當")
file.write('\n')
file.close()
```

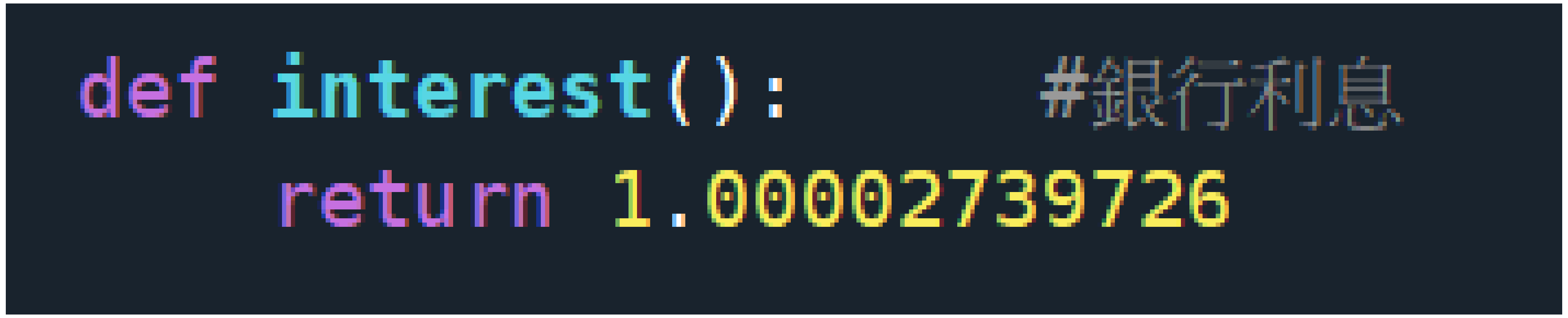


```python
file = open("cos_inf_day.txt", mode='a')
file.write("編號"+str(i)+"沒買到商品")
file.write('\n')
file.close()
```

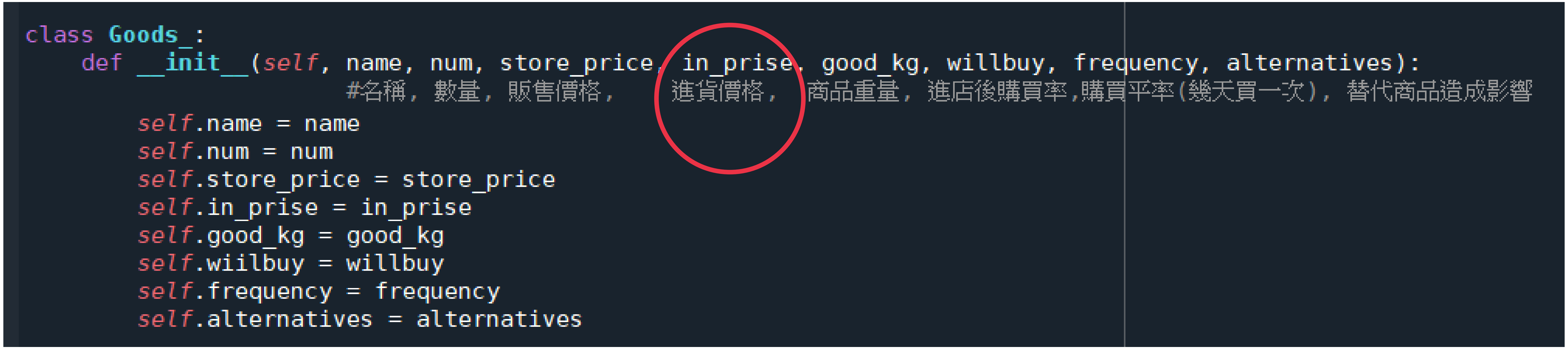
成本



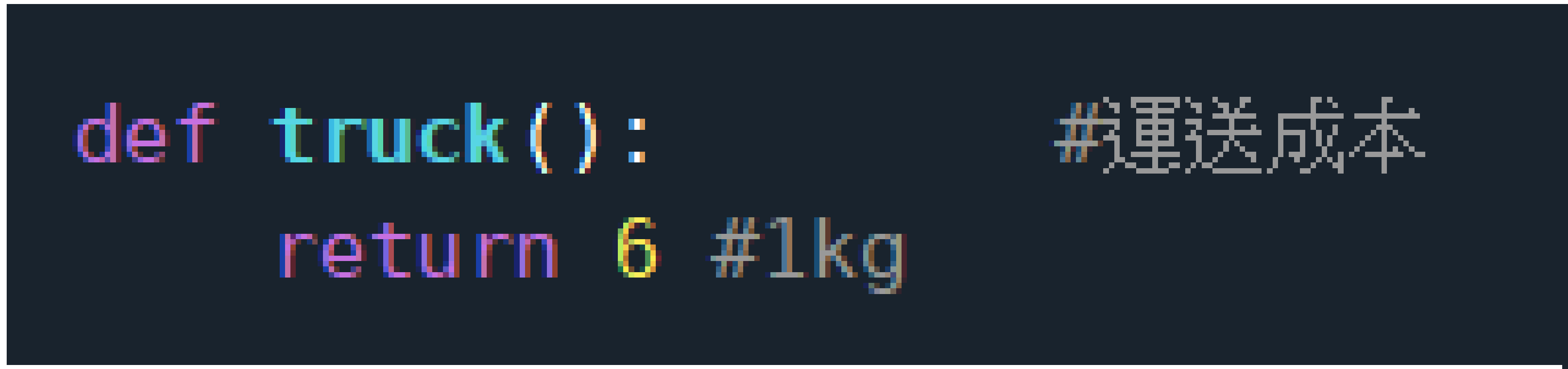
```python
#shortage_cost 短缺商譽損失
def compensation(): #賠償
    pass

def ctmlose_shortage(n):  #顧客損失
    if n == 1:
        return -0.2
    else:
        return (-0.25)
```



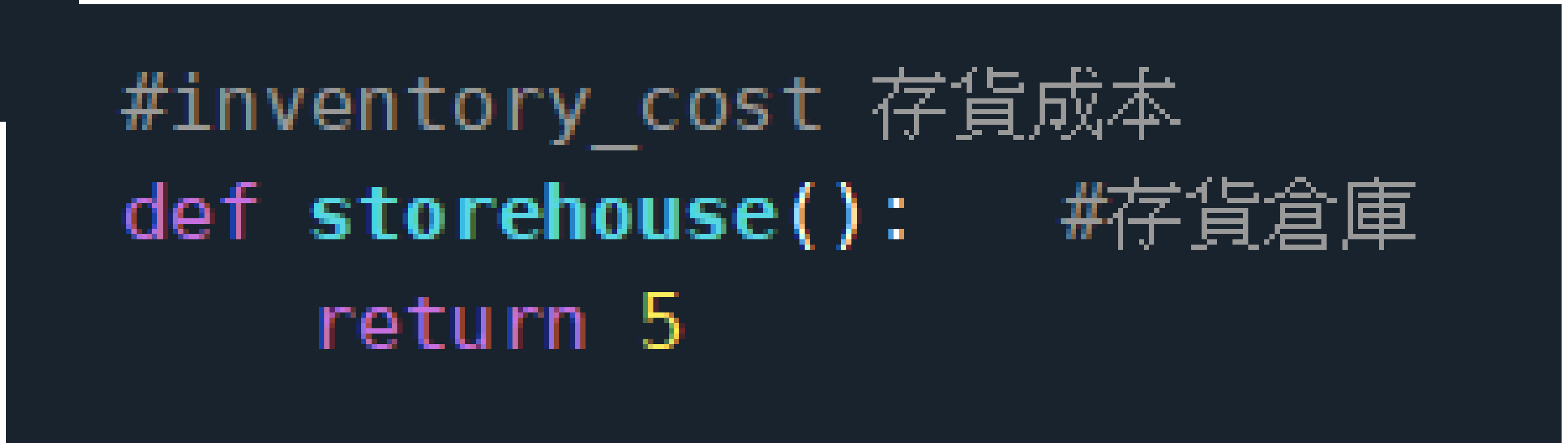
```python
def interest():      #銀行利息
    return 1.00002739726
```



```
class Goods_:
    def __init__(self, name, num, store_price, in_prise, good_kg, willbuy, frequency, alternatives):
                       #名稱，數量，販售價格，   進貨價格， 商品重量，進店後購買率,購買平率(幾天買一次)，替代商品造成影響
        self.name = name
        self.num = num
        self.store_price = store_price
        self.in_prise = in_prise
        self.good_kg = good_kg
        self.wiilbuy = willbuy
        self.frequency = frequency
        self.alternatives = alternatives
```

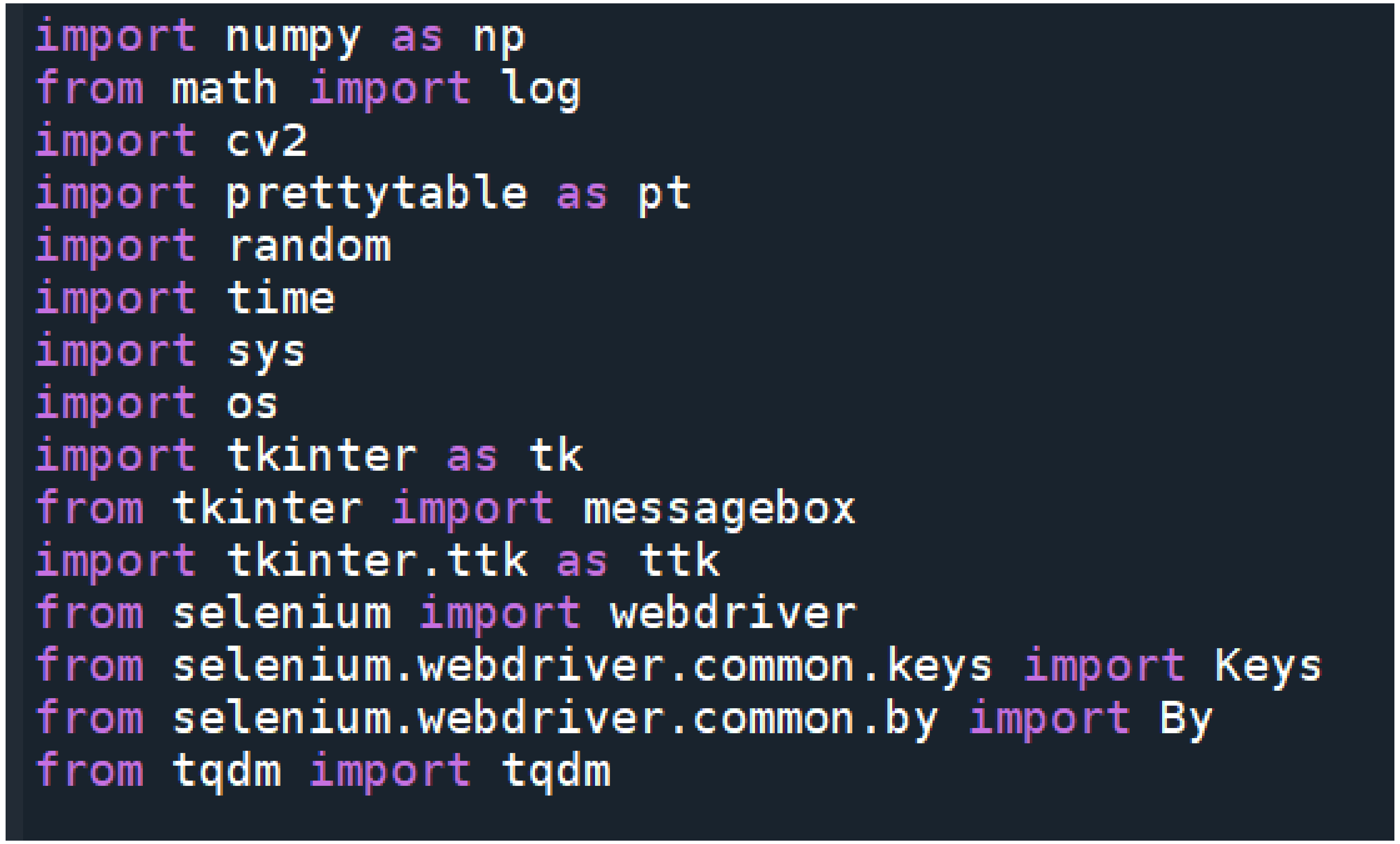


```
def truck():        #運送成本
    return 6 #1kg
```



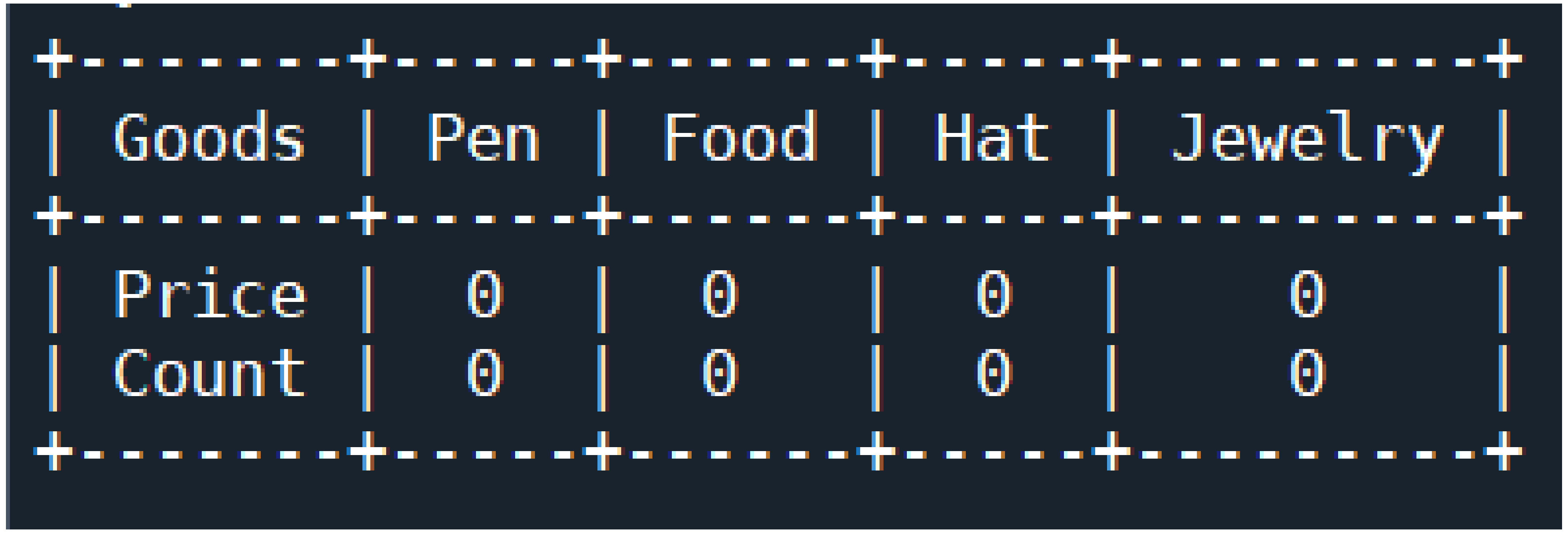
```
#inventory_cost 存貨成本
def storehouse():    #存貨倉庫
    return 5
```

執行程式：

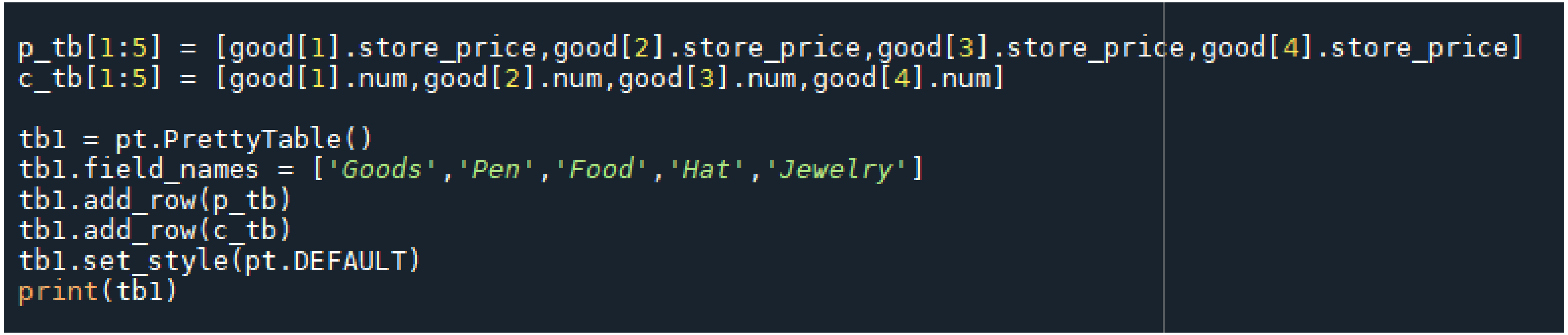


```
import numpy as np
from math import log
import cv2
import prettytable as pt
import random
import time
import sys
import os
import tkinter as tk
from tkinter import messagebox
import tkinter.ttk as ttk
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from tqdm import tqdm
```

prettytable

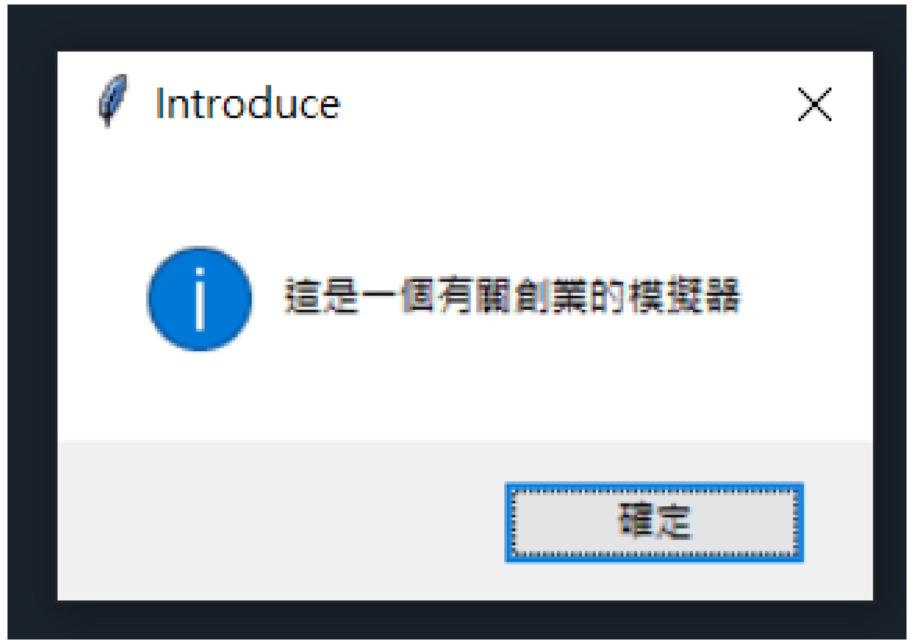


```
+-------+-----+------+-----+---------+
| Goods | Pen | Food | Hat | Jewelry |
+-------+-----+------+-----+---------+
| Price |  0  |  0   |  0  |    0    |
| Count |  0  |  0   |  0  |    0    |
+-------+-----+------+-----+---------+
```

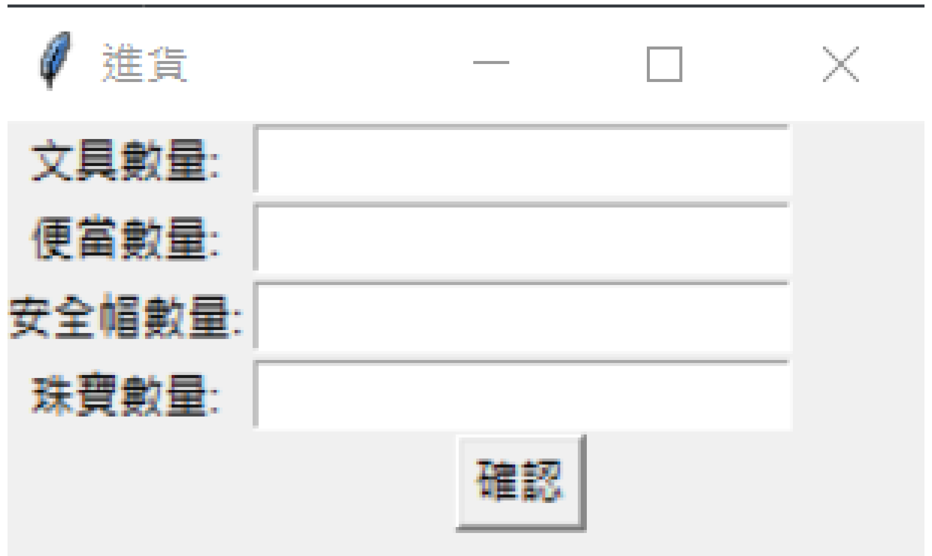


```
p_tb[1:5] = [good[1].store_price,good[2].store_price,good[3].store_price,good[4].store_price]
c_tb[1:5] = [good[1].num,good[2].num,good[3].num,good[4].num]

tb1 = pt.PrettyTable()
tb1.field_names = ['Goods','Pen','Food','Hat','Jewelry']
tb1.add_row(p_tb)
tb1.add_row(c_tb)
tb1.set_style(pt.DEFAULT)
print(tb1)
```

tkinter



這是一個有關創業的模擬器

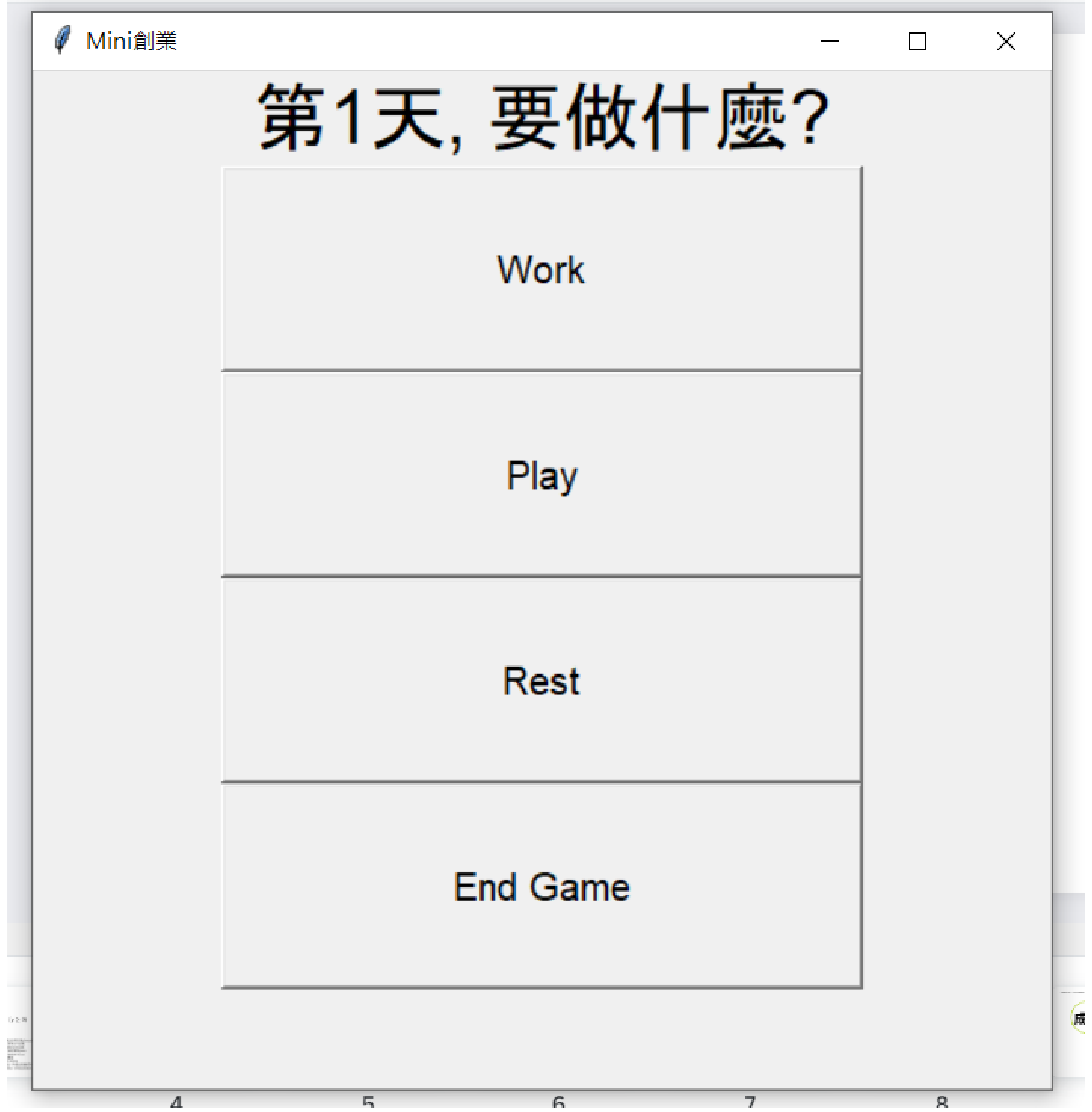


文具數量:

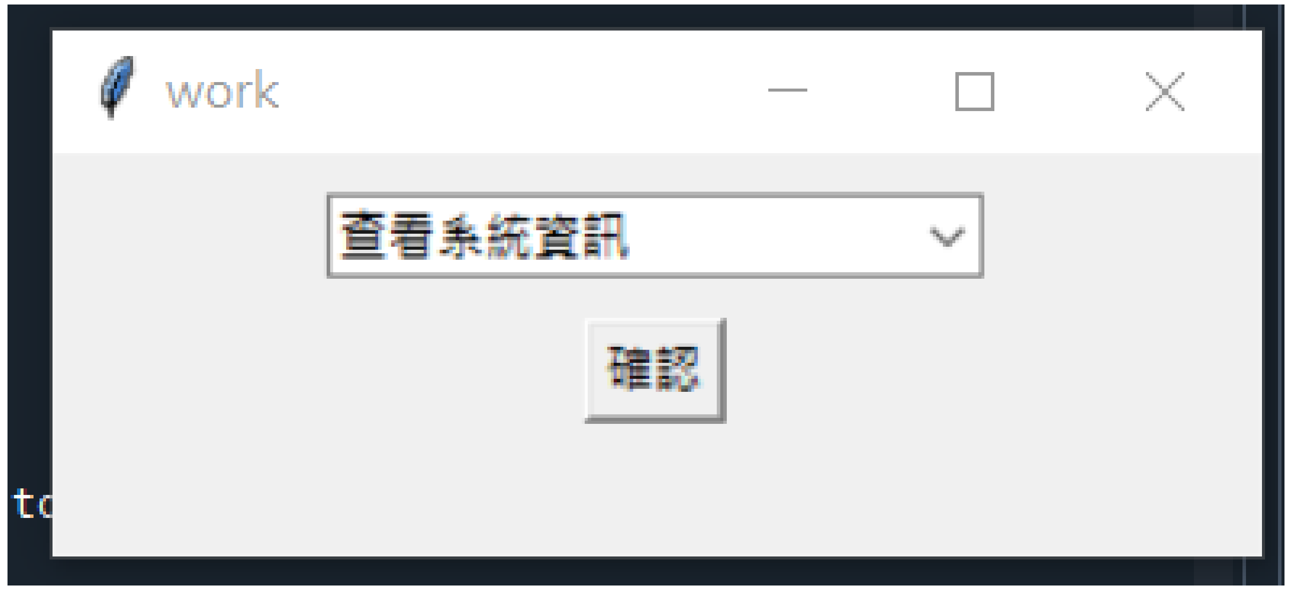
便當數量:

安全帽數量:

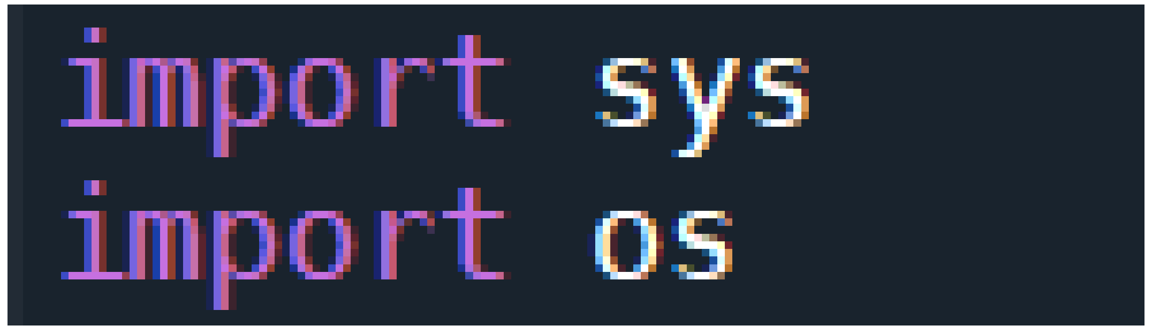
珠寶數量:



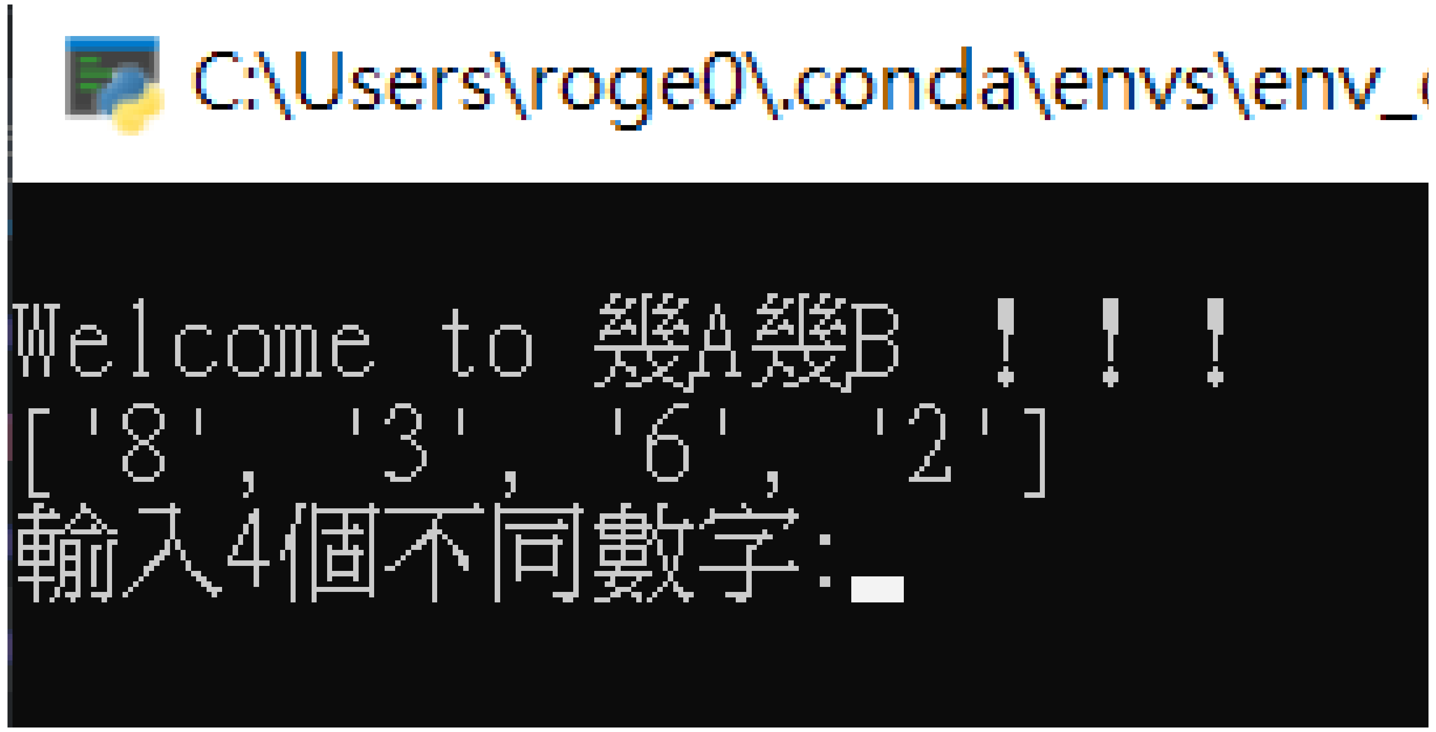
第1天, 要做什麼?



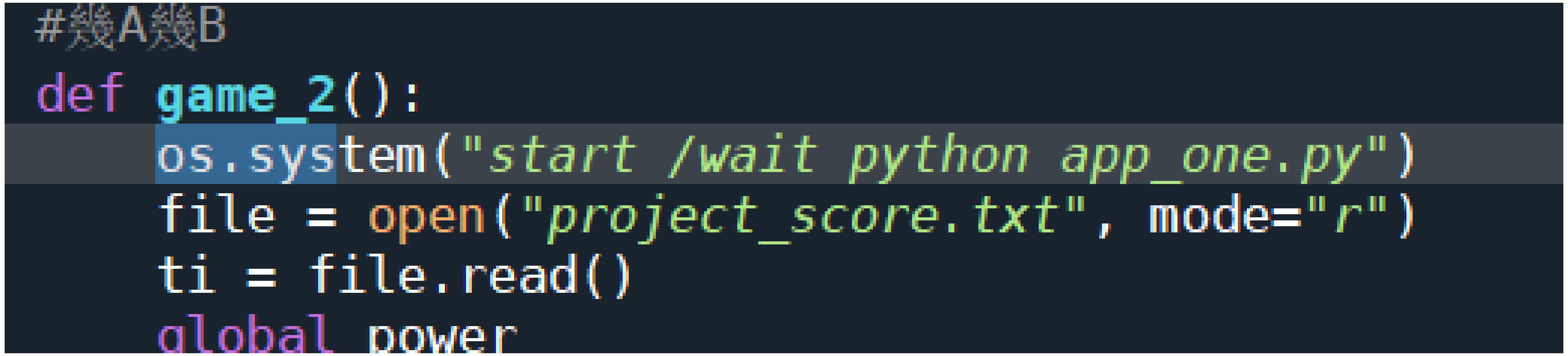
查看系統資訊



```
import sys
import os
```

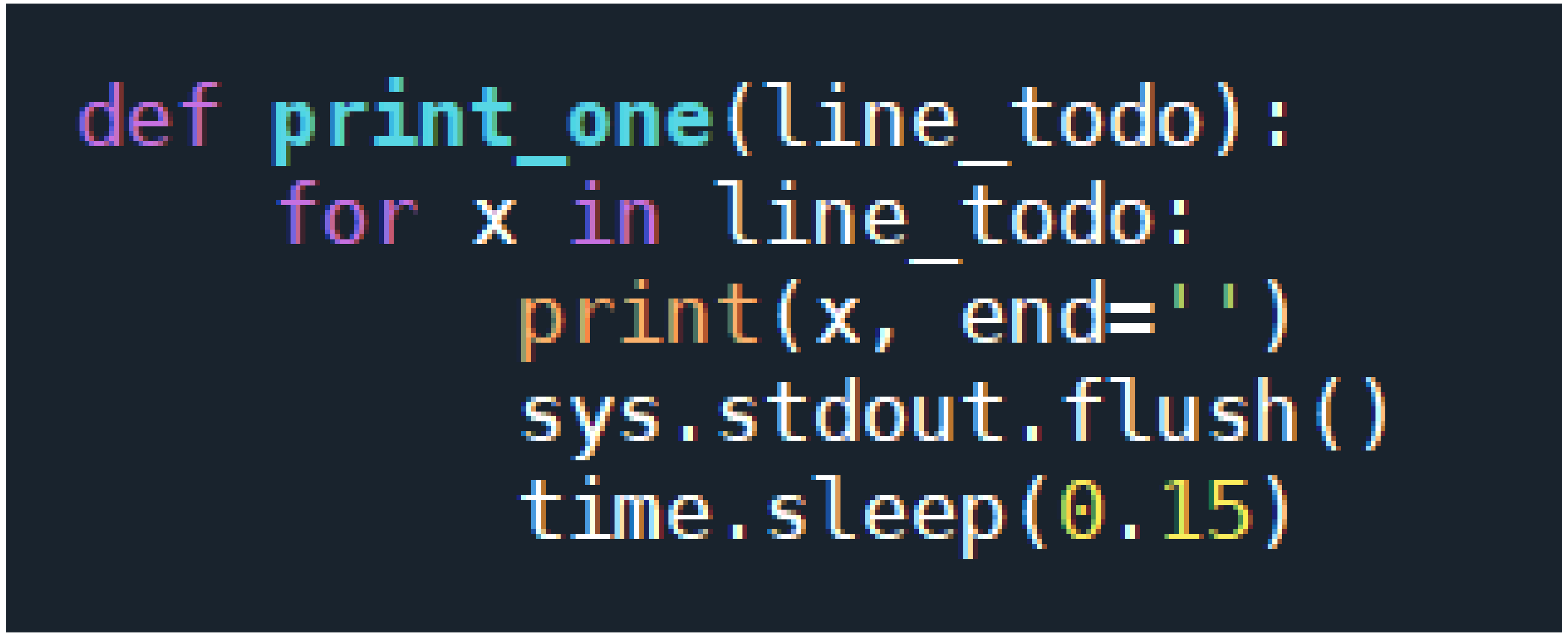


```
Welcome to 幾A幾B ！！！
['8', '3', '6', '2']
輸入4個不同數字:
```



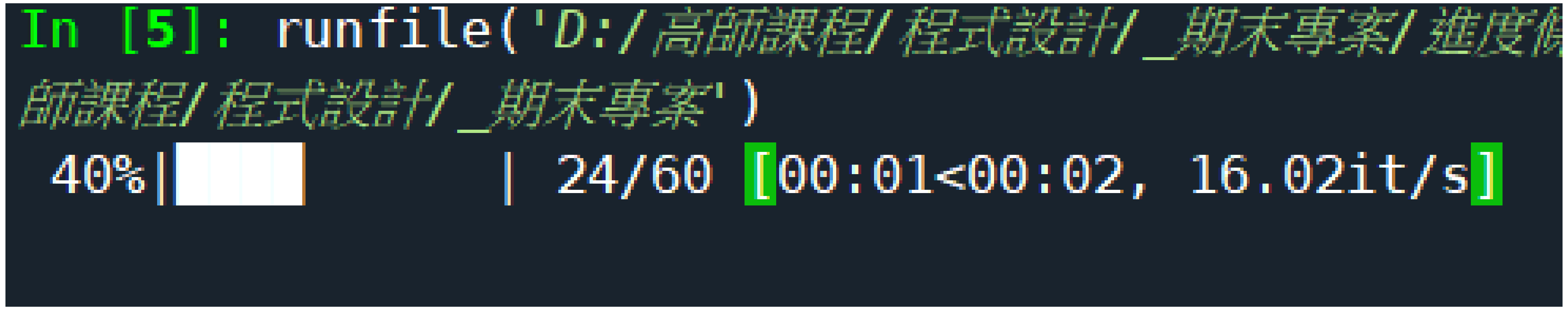
```
#幾A幾B
def game_2():
    os.system("start /wait python app_one.py")
    file = open("project_score.txt", mode="r")
    ti = file.read()
    global power
```

time



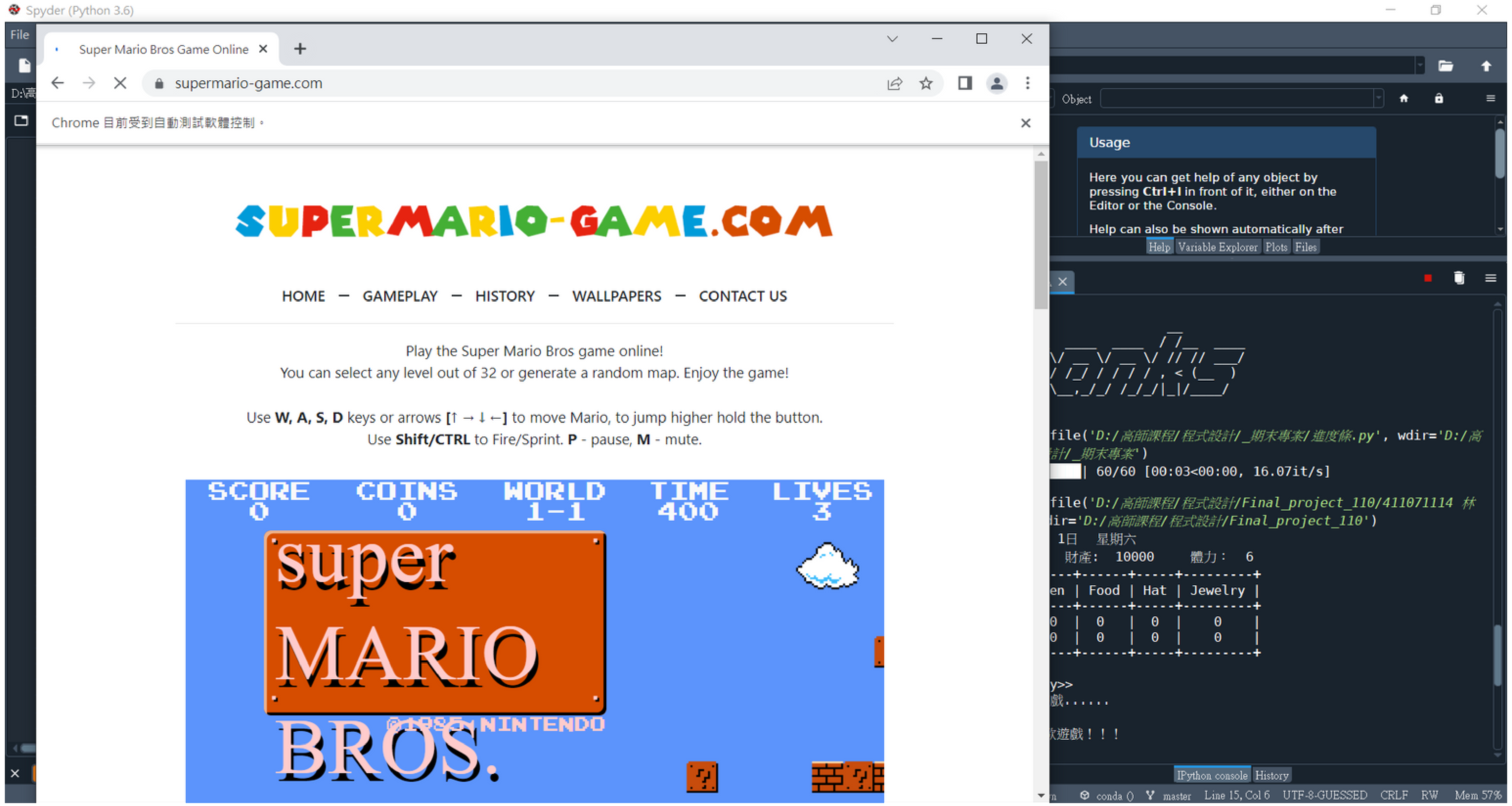
```python
def print_one(line_todo):
    for x in line_todo:
        print(x, end='')
        sys.stdout.flush()
        time.sleep(0.15)
```

tqdm



```
In [5]: runfile('D:/高師課程/程式設計/_期末專案/進度
師課程/程式設計/_期末專案')
 40%|████      | 24/60 [00:01<00:02, 16.02it/s]
```

selenium



supermario-game.com

Chrome 目前受到自動測試軟體控制。

SUPERMARIO-GAME.COM

HOME — GAMEPLAY — HISTORY — WALLPAPERS — CONTACT US

Play the Super Mario Bros game online!
You can select any level out of 32 or generate a random map. Enjoy the game!

Use **W, A, S, D** keys or arrows **[↑ → ↓ ←]** to move Mario, to jump higher hold the button.
Use **Shift/CTRL** to Fire/Sprint. **P** - pause, **M** - mute.

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after

```
                __
   ____ _____  / /_______
\/ __ \/ __ \/ //_/ ___/
/ /_/ / / / / ,< (__  )
\__,_/_/ /_/_/|_|/____/

file('D:/高師課程/程式設計/_期末專案/進度條.py', wdir='D:/高
計/_期末專案')
     | 60/60 [00:03<00:00, 16.07it/s]

file('D:/高師課程/程式設計/Final_project_110/411071114 林
dir='D:/高師課程/程式設計/Final_project_110')
 1日   星期六
  財產:  10000     體力:  6
---+------+-----+---------+
en | Food | Hat | Jewelry |
---+------+-----+---------+
0  |  0   |  0  |    0    |
0  |  0   |  0  |    0    |
---+------+-----+---------+

y>>
戲......

欢遊戲！！！
```