

選擇主題（三）：打造你的冒險遊戲

by 田弘華 **Hung-Hua Tien**

重點複習1：真與假

我們把True和False稱為布林值。

電腦可以決定一個敘述是真或假。電腦會用這種真或假的值來決定要執行演算法的那個部分的程式碼。

- 我們用兩個等號或關係運算子來比較兩個項目。如果相同，則結果為真，否之則為假。
- 我們也可以結合邏輯運算子，**and/or/not**，得到布林值。

重點複習2：條件敘述式

我們會根據一件事的True或False來做決定和執行不同的動作。

同理，電腦也會根據「**if**條件敘述式」(有True/False值的布林表示式)來執行程式碼。

- 若條件是的值為True，則執行一組程式碼敘述式（**if** 程式區塊）；
- 若條件是的值為False，則執行另一組(**else**程程式區塊)。

～在條件敘述式輸入冒號後，後面的每一行都要縮排(indentation)來表示這是執行的程式碼區塊。

- Python對縮排很挑剔。
- 同屬一區塊的每行程式碼要有相同的縮排距離。
- 請統一空4格或用tab鍵，但不要混用。

～這種敘述稱為條件敘述式(**conditional statement**)或**if-else**敘述式。

- 巢狀條件式：在第一個條件後看另一個條件，再做進一步決定。
- **elif**敘述式：當一個問題需要在多個條件式使用不同的程式區塊，可以使用Python的**elif**結構來取代多重巢狀條件式。

重點複習3: 條件迴圈

條件迴圈(**conditional loop**)，只要條件維持True，電腦就可以執行迴圈（重複執行一組程式碼）。

這種迴圈是用在程式碼要執行的確切次數未知的時候。只要條件式維持True，它就會一直執行。也就是不知道要重複多少次的時候，就使用這種迴圈。

文字冒險遊戲，又稱互動故事 (interactive fiction)。

使用者在遊戲過程中做決定、收集物品，或回答問題，這是個練習條件式(if敘述式)和巢狀條件式的好機會。

- 這只是範例，你可以把它變成自己的冒險遊戲，擴充物品、選擇和角色。用這個概念打造更複雜的冒險遊戲是練習Python程式設計技巧的好方法。
- 用流程圖可以讓這個專題更容易規劃和設計程式，每一步都根據流程圖來加入程式碼。每一步完成後都要執程式碼來確認沒有問題。

故事線：使用者在山中健行，聽到一個聲音。後來迷路了，必須做決定把遊戲破關才能安全回家。

遊戲執行範例

歡迎來到聖塔克魯茲山上冒險遊戲

你現在在加州聖塔克魯茲。你在傍晚一個人爬山。

你可以帶一項物品上路：

地圖(m)、手電筒(f)、巧克力(c)、繩子(r)或棒子(s)：

你選哪個？：c

你聽到嗡嗡聲。

要尋找聲音來源嗎？輸入y或n：n

好主意。你沒有冒險。

你開始走回起點。

你發現自己迷路了！

身後的聲音愈來愈大。你開始驚慌！

要開始跑(r)或停下來打電話(c)？：c

電話不通。

要跑(r)或再打一次(c)？：c

電話不通。

要跑(r)或再打一次(c)？：r

你跑很快。聲音變很大。

一個女人騎電動機車從後面接近你。

她問：「我最喜歡的電腦程式語言是什麼？」：PYTHON

她說：「沒錯，Python是最喜歡的程式語言。如果你有巧克力，我可以幫你。」

幸運的是，你當初選對了！

你把巧克力給她。

她幫助你回家。

恭喜！你成功脫身。你破關了。

歡迎來到聖塔克魯茲山上冒險遊戲

=== === === === ===

你現在在加州聖塔克魯茲。

你在傍晚一個人爬山。

~Question 1：爬山帶什麼？ - 什麼選項？

你可以帶一項物品上路：

地圖(m)、手電筒(f)、巧克力(c)、繩子(r)或棒子(s)
你選哪一個? :**c**

~Question 2: 爬山遇險 - 聽到怪聲，要不要冒險？

你聽到嗡嗡聲。

要尋找聲音來源嗎？輸入y或n：

選擇一：n

好主意。你沒有冒險。

你開始走回起點。

你發現自己迷路了！

身後的聲音越來越大。你開始驚慌！

選擇二：y

你向聲音接近。

聲音突然停止。

你迷路了！...

你嘗試打電話，但沒有訊號！

=== === === === ===

~**選擇n + 後續故事第1集**

要開始跑(r)或停下來打電話(c)? : **c**

電話不通。

要跑(r)還是再打一次(c)? : **c**

電話還是不通。

要跑(r)還是再打一次(c)? : **r**

你拼命地快跑，而聲音越來越大。

~**選擇n + 後續故事第2集**

一個女人騎電動機車從後面接近你。

她問：「我最喜歡的電腦程式語言是什麼？」：

答案I. Python

她說：「沒錯，*Python* 是我最喜歡的程式語言。如果你有巧克力，我可以幫你。」

(1)幸運的是，你當初選對了！

你把巧克力給她。

她幫助你回家。

恭喜！你成功脫身。你破關了！

(2)沒有巧克力，當初應該選巧克力才對。

她騎車走了，留下孤單的你。

你輸了。

答案II. 其他答案，答錯了

她不喜歡你的回答。

他騎車走了，留下迷路的你。

你輸了。

=== === === === ===

~**選擇y + 後續故事**

Q：迷路電話不通，選擇前進方向

你要往那個方向前進？ 北(NORTH)、南(SOUTH)、東(EAST)或西(WEST)：

（ 1 ）後續故事第1種：若選北NORTH

你抵達廢棄小屋。

若選地圖，你用地圖找到回家之路。

恭喜！你破關了。

～如果你有地圖，就能找到從這裡回家的路。

---你還在迷路！你輸了。---

（ 2 ）後續故事第2種，若選南SOUTH

你抵達有斷橋的河流。

若選繩子或棒子，你選了可以修好橋樑的物品。

你修好橋樑、過橋，並找到回家的路。

恭喜！你破關了。

～如果你有繩子或棒子，就能修好橋樑。

---你還是在迷路。你輸了。---

（ 3 ）後續故事第3種，若選西WEST

你走路時被傾倒的樹絆倒。

你腳受傷了。你坐下等待救援。

～這可能要花很久時間。你還在迷路。

---你輸了。---

（ 4 ）後續故事第4種，若選東EAST，即其他選項

你抵達公路邊。很暗。

若選手電筒，你用手電筒發出訊號。

一輛車停下來，載你回家。

恭喜你！你破關了，安全回到家。

～如果你有手電筒，就能發出求救訊號。

---你還是在迷路。你輸了---

觀念：Input + Process + Output

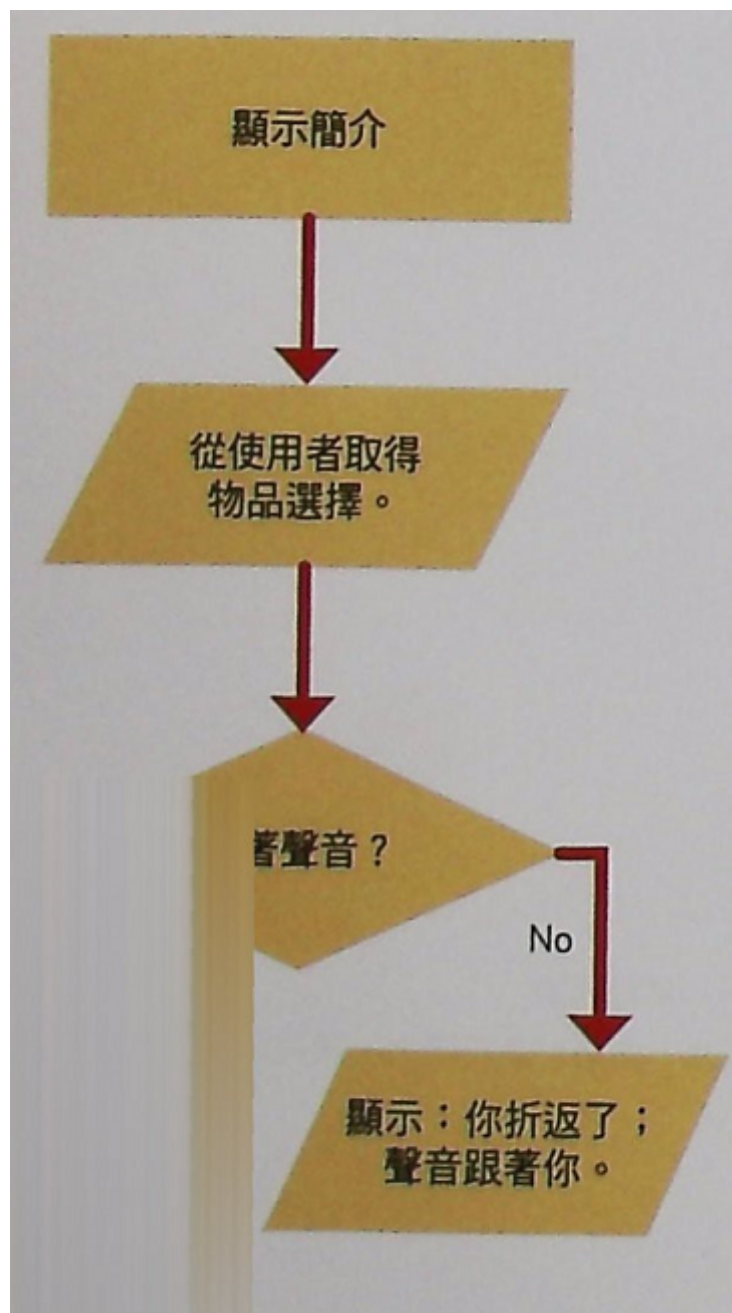
- print輸出：
在遊戲中簡介故事
- input輸入＋變數儲存＋print輸出：（單一選擇）
讓使用者用帶一個物品爬山。
- input輸入＋（變數儲存＋if條件選擇）＋print不同結果輸出
讓使用者決定冒不冒險，選擇不同、結果不同。
讓使用者依照情況，跑還是打電話、往東西南北的方向，選擇不同、結果不同。

第1步：加入簡介並要使用者做出第一個決定

為這個專題開一個新檔，可以命名為**AdventureGame.py**。

```
# 冒險遊戲
print("歡迎來到聖塔克魯茲山上冒險遊戲！")
print("*****")
print("你現在在加州聖塔克魯茲。")
print("你在傍晚一個人爬山。")
print("你可以帶一項物品上路：")
print("地圖(MAP)、手電筒(FLASH)、巧克力(CHOCOLATE)、繩子(ROPE)、或棒子(STICK): ")

item = input("你選擇那個？ ") # Question1:爬山帶什麼
```



```
# 開始冒險之旅
print("你聽到嗡嗡聲。") # Question2:冒險遇險 - 聽到怪聲
## 冒險旅程中選擇
```

```

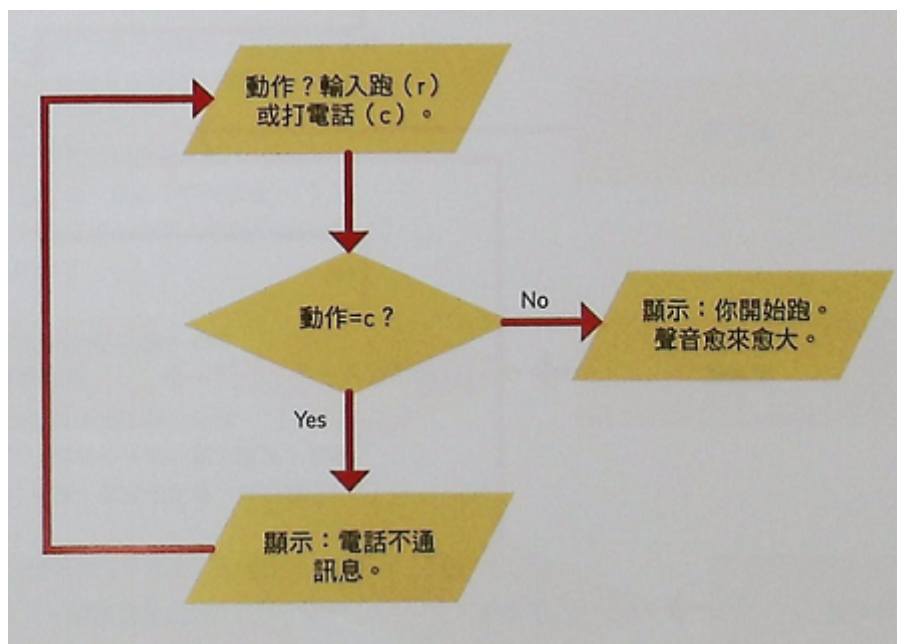
choice1 = input("要尋找聲音的來源嗎？ 請輸入YES 或 NO: ")
if choice1 == "YES":
    print("你向聲音接近。")
    print("聲音突然停止。")
    print("你迷路了！...")
    print("你嘗試打電話，但沒有訊號！")
else:
    print("好主意。你沒有冒險。")
    print("你開始走回起點。")
    print("你發你發現自己迷路了！")
    print("身後聲音越來越大，你開始驚慌！")

```

第2步：加入迴圈

建立兩種選擇的可能性後，我們要加入更多程式碼來延伸故事。

- 這個步驟，我們要擴充else區塊（選擇一：n）。
- 我們讓使用者選擇跑或打電話求救。
 - 透過while迴圈，更生動地描述打電話還是跑的雙向選擇。



```

# 這行print("身後聲音越來越大，你開始驚慌！")後，加入以下程式碼：
action = input("要開始跑或停下來打電話？ 選擇RUN或CALL: ")

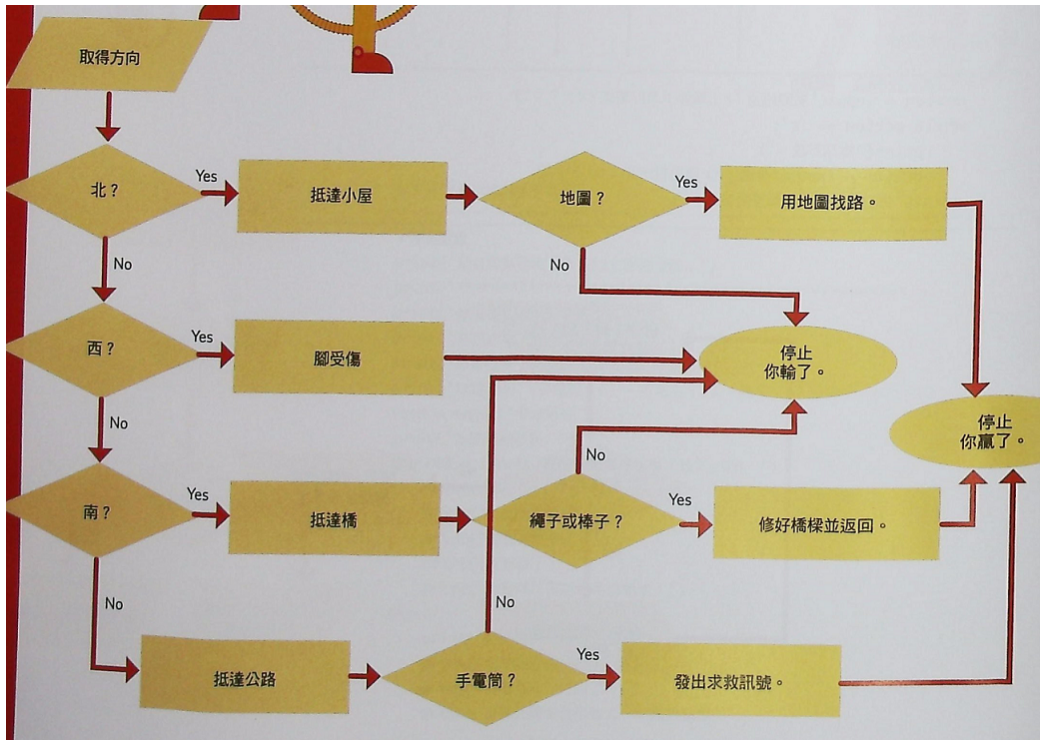
## 條件選擇與迴圈
while action == "CALL":
    print("電話不通。")
    action = input("要跑或再打一次電話？ 選擇RUN或CALL: ")
print("你拼命地快跑，而聲音越來越大。")

```


第3步：加入方向選擇

建立兩種可能性後，我們要加入更多程式碼來延伸故事。

- 擴充if區塊（選擇二：y）。
- 我們讓使用者在聲音停止時選擇前進方向。
- 這是巢狀if敘述式的例子。
 - 用if-elif-else 做多向選擇，有東西南北四個選項。
 - 各方向選項中，加上前面攜帶物品的if-else條件判斷結果。



```
# 在這行後，print("你嘗試打電話，但沒有訊號！")，加入以下的程式碼：
direction = input("你要往那個方向前進？ 北(NORTH)、南(SOUTH)、東(EAST)或西(WEST)： ")

if direction == "NORTH":
    print("你抵達廢棄小屋。")
    if item == "MAP":
        print("你用地圖找到回家之路。")
        print("恭喜！你破關了。")
    else:
        print("如果你在地圖，就能找到從這裡回家的路。")
        print("---你還在迷路！你輸了。---")
elif direction == "SOUTH":
    print("你抵達有斷橋的河流。")
    if item == "ROPE" or item == "STICK":
        print("你選了可以修好橋樑的物品。")
        print("你修好橋樑、過橋，並找到回家的路。")
        print("恭喜！你破關了。")
    else:
        print("如果你有繩子或棒子，就能修好橋樑。")
        print("---你還是在迷路。你輸了。---")
elif direction == "WEST":
```

```

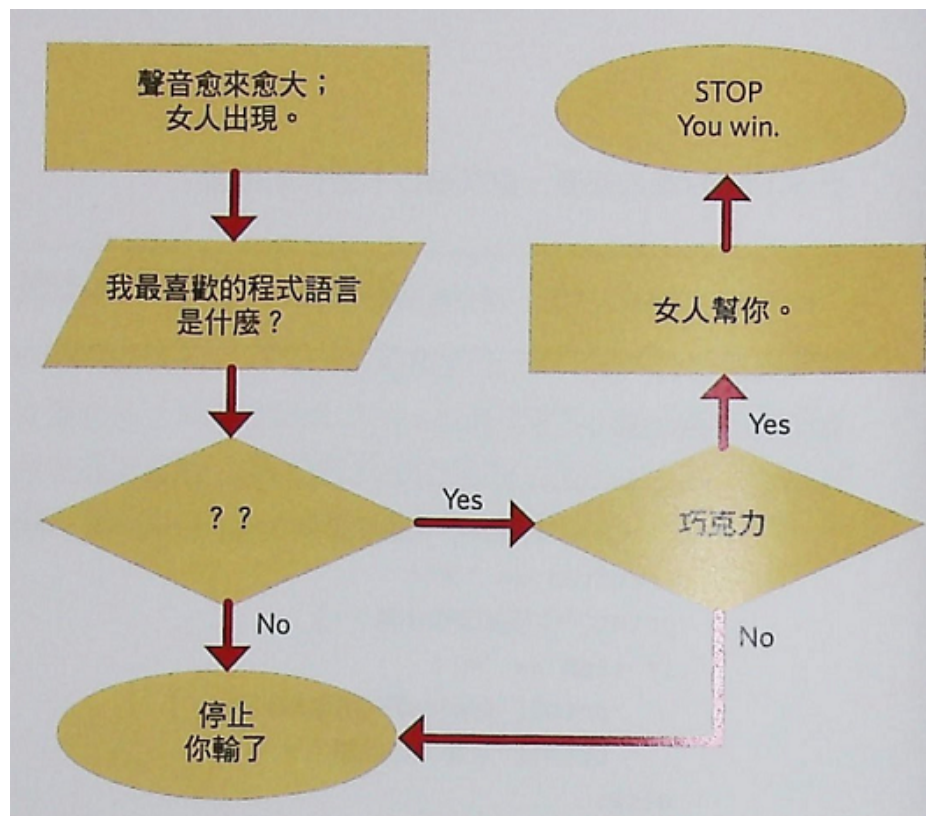
print("你走路時被傾倒的樹絆倒。")
print("你腳受傷了。你坐下等待救援。")
print("這可能要花很久時間。你還在迷路。")
print("---你輸了。---")
else:
    print("你抵達公路邊。很暗。")
    if item == "FLASH":
        print("你用手電筒發出訊號。")
        print("一輛車停下來，載你回家。")
        print("恭喜你！你破關了，安全回到家。")
    else:
        print("如果你有手電筒，就能發出求救訊號。")
        print("---你還是在迷路。你輸了---")

```

第4步：加入其他選擇

建立兩種可能性後，我們要加入更多程式碼來延伸故事。

- 這個步驟，我們要繼續擴充**else**區塊。
- 我們讓使用者回答問題或謎題，答案正確之後，才能進行下一個動作。



在while迴圈的print("你拼命地快跑，而聲音越來越大。")後，加入以下的程式碼：

兩人互動

```

print("一個路人騎著機車從後面接近你。")

answer = input("她說：『我最喜歡的電腦程式語言是什麼？』； ")
if answer == "PYTHON":
    print("她說：『沒錯，Python是最喜歡的程式語言。』")
    print("『如果有巧克力，我就幫你。』")
    if item == "CHOCOLATE":

```



```

print("很幸運地，你正好選擇了巧克力。")
print("你把巧克力給她。")
print("她幫助你回家。")
print("恭喜！你安全脫身，你破關了！")
else:
    print("沒有巧克力，當初應該選巧克力才對。")
    print("她騎車走了，留下孤單的你。")
    print("你輸了。")
else:
    print("她不喜歡你的回答。")
    print("他騎車走了，留下迷路的你。")
    print("你輸了。")

```

第5步：改善使用者輸入並加入錯誤檢查

已經有了堪用的遊戲了，我們來做些改進吧！

- (1)我們改善輸入**PYTHON**的步驟。
 - 讓使用者能輸入大寫、小寫和大小寫混合，即Python、python、PYTHON都好。
 - 用**or**運算子 讓選擇變成三選一。
 - 用**lower()**函式 將輸入統一轉換成python。（用upper()轉成PYTHON)
- (2)我們加入錯誤檢查。
 - 在冒險旅程中第一個選擇的地方，我們加入**while**迴圈。
 - 當使用者輸入錯誤時，可以重新輸入YES或NO的選項。
 - 當使用者輸入的不是YES或NO的選項之一時，就重試，所以用not條件判斷。

```

# 改善輸入PYTHON的步驟

## 方法（一）：
if answer == "python" or answer == "Python" or answer == "PYTHON"

## 方法（二）：
if answer.lower() == "python":

```

```

# 加入第一個選擇的錯誤檢查

choice1 = input("要尋找聲音的來源嗎？ 請輸入YES 或 NO： ") #冒險旅程中的第一個選擇
while not (choice1 == "YES" or choice1 == "NO"):
    choice1 = input("輸入無效，注意是輸入YES 或 NO二選一： ")

```

第6步：在故事加入暫停

為了讓遊戲更順暢，我們可以加入暫停來減緩輸出。這樣讓使用者有時間閱讀，也增加戲劇張力。

- 在遊戲開頭，告訴使用者他迷路了之後，可以先停頓幾秒再繼續。
- 用**time**模組中的**sleep()**函式來插入這種暫停。
- 遊戲中許多地方都可以這樣做。

```
import time

time.sleep(3)

choice1 = input("要尋找聲音的來源嗎？ 請輸入YES 或 NO: ") #冒險旅程中的第一個選擇
if choice1 == "YES":
    print("你向聲音接近。")
    print("聲音突然停止。")
    time.sleep(3) #暫停3秒讓讀者閱讀
    print("你迷路了！...")
    time.sleep(3) #暫停3秒增加戲劇性
    print("你嘗試打電話，但沒有訊號！")
else:
    print("好主意。你沒有冒險。")
    time.sleep(3) #暫停3秒
    print("你開始走回起點。")
    print("你發你發現自己迷路了！")
    time.sleep(3) #暫停3秒
    print("身後聲音越來越大，你開始驚慌！")
```

程式專題設計

～要訣心得

進行任何程式設計專題，最好的方式都是一步一步推進。

- 不要一次寫太多程式碼。
- 一個段落運作沒有問題之後，可以隨時回來修改，讓它運作更順利。

～方向一：你可以想一個更精彩的故事！

～方向二：你也可以擴充這個遊戲，讓它更好！！

- 在開頭替換或加入更多可以選則攜帶的物品。
- 替換或加入更多路人問的問題和產生的動作。
- 加入更多問題或是可以贈送、交換、交易的物品，讓謎題更豐富。
- 加入更多的錯誤檢查，檢查是否所有的輸入是有效的。
- 加入更多的使用者回應方式，除了YES和NO之外。
- 加入在不同關卡會變化的能量變數。
- 用**sleep()**函式加入更多的暫停，讓遊戲更順暢。
- 加入一些文字圖片，讓輸入更美觀。

～ 在故事中加入更多複雜性和決策，可以讓它更精彩。

～用各種想法擴充這個遊戲，用你的創意和學到的Python程式碼。

```
# 參考：完整的英文程式碼

# adventure game
print('welcome to the Santa Cruz Mountain Adventure Game!')
print('*****')
print('You are visiting Santa Cruz, California.')
print('You go on an evening hike alone in the mountains.')
print('You can pick one item to take with you - ')
print('map (m), flashlight(f), chocolate (c), rope(r), or stick (s): ')
item = input('What do you choose?: ')
print('You hear a humming sound.')
choice1 = input('Do you follow the sound? Enter y or n: ')
if choice1 == 'y':
    print('You keep moving closer to the sound.')
    print('The sound suddenly stops.')
    print('You are now LOST! ... ')
    print('You try to call on your phone, but there is no signal!')
    direction = input('Which direction do you go? north, south, east, or west: ')
    if direction == 'north':
        print('You reach an abandoned cabin.')
        if item == 'm':
            print('You use the map and find your way home.')
            print('CONGRATULATIONS! You won the game. ')
        else:
            print('If you had a map, you could find your way from here.')
            print('---You are still lost. You lost the game.---')
    elif direction == 'south':
        print('You reach a river with a broken bridge.')
        if item == 'r' or item == 's':
            print('You chose an item that can fix the bridge.')
            print('You fix the bridge, cross over, and find your way home')
            print('CONGRATULATIONS! You won the game.')
        else:
            print('If you had a rope or a stick, you could fix the bridge.')
            print('---You are still lost. You lost the game.---')
    elif direction == 'west':
        print('You are walking and trip over a fallen log.')
        print('You have hurt your foot. You sit down and wait for help.')
        print('This could be a long time. You are still lost.')
        print('---You lost the game.---')
    else:
        print('You reach the side of the highway. It is dark.')
```

```

        if item == 'f':
            print('You use the flashlight to signal.')
            print('A car stops and gives you a ride home.')
            print('CONGRATULATIONS! You got out safely. You won the
game.')
```

```

        else:
            print('If you had a flashlight, you could signal for
help.')
```

```

            print('---You are still lost. You lost the game.--')
```

```

    else:
        print('Good idea. You are not taking risks. ')
        print('You start walking back to the starting point.')
        print('You realize you are LOST! ')
        print('The sound is behind you and is getting louder. You
panic! ')
        action = input('Do you start running (r), stop to make a call
(c)?: ')
        while action == 'c':
            print('The call does not go through')
            action = input('Do you want to run (r), or try calling
again (c)?: ')
            print('You are running fast. The sound gets really loud')
```

```

        print('A woman on an electric scooter comes up behind you.')
```

```

        answer = input('She says, "Name my favorite computer
programming language.": ')
        if answer.lower() == 'python':
            print('She says, "Yes, Python is my favorite programming
language."')
```

```

            print('"If you have some chocolate, I can help you."')
```

```

            if item == 'c':
                print('Luckily you did choose correctly!')
```

```

                print('You give her the chocolate.')
```

```

                print('She helps you get home.')
```

```

                print('CONGRATULATIONS! You got out safely. You won the
game.')
```

```

            else:
                print('You should have chosen that chocolate!')
```

```

                print('She rides away, leaving you alone and lost.')
```

```

                print('You lost the game.')
```

```

        else:
            print('She did not like your answer.')
```

```

            print('She rides away, leaving you lost!')
```

```

            print('You lost the game.')
```