Python

for

```
初始值
while 條件式:
  敘述式
  條件改變
```

```
for 控制變數 in 條件範圍:
  敘述式
```

在 Python 語言有兩種重覆結構

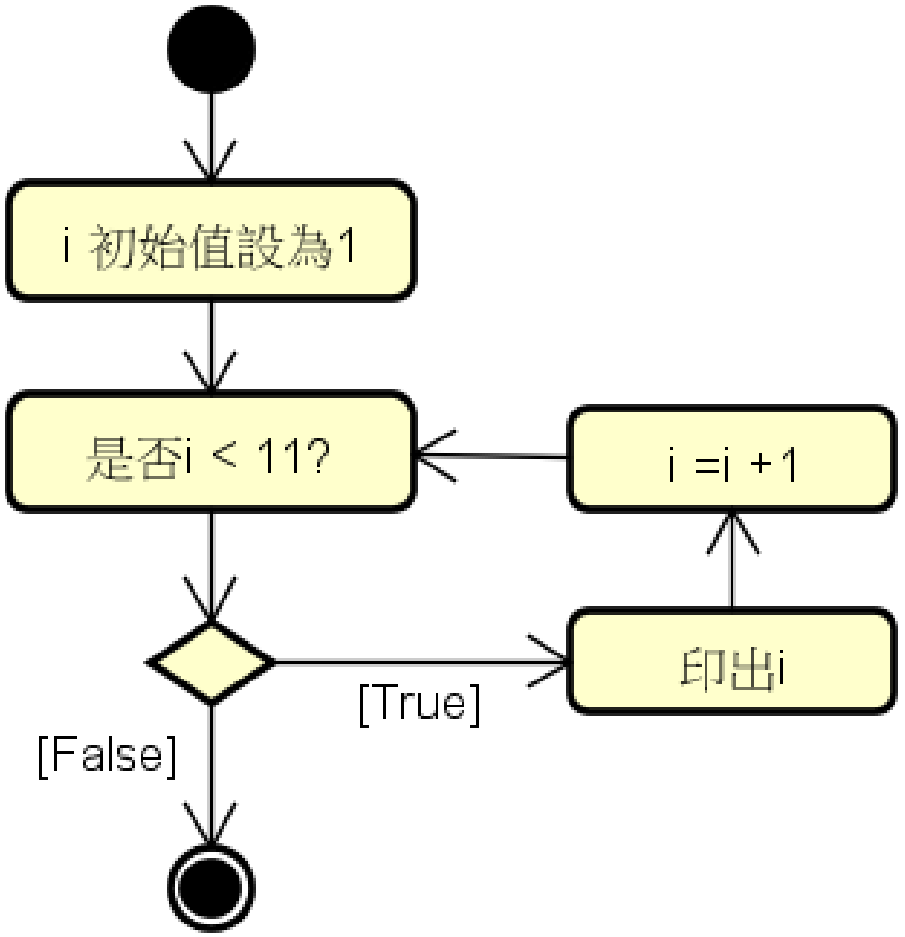
```
初始值
while 條件式:
  敘述式
  條件改變
```

```
for 控制變數 in 條件範圍:
  敘述式
```

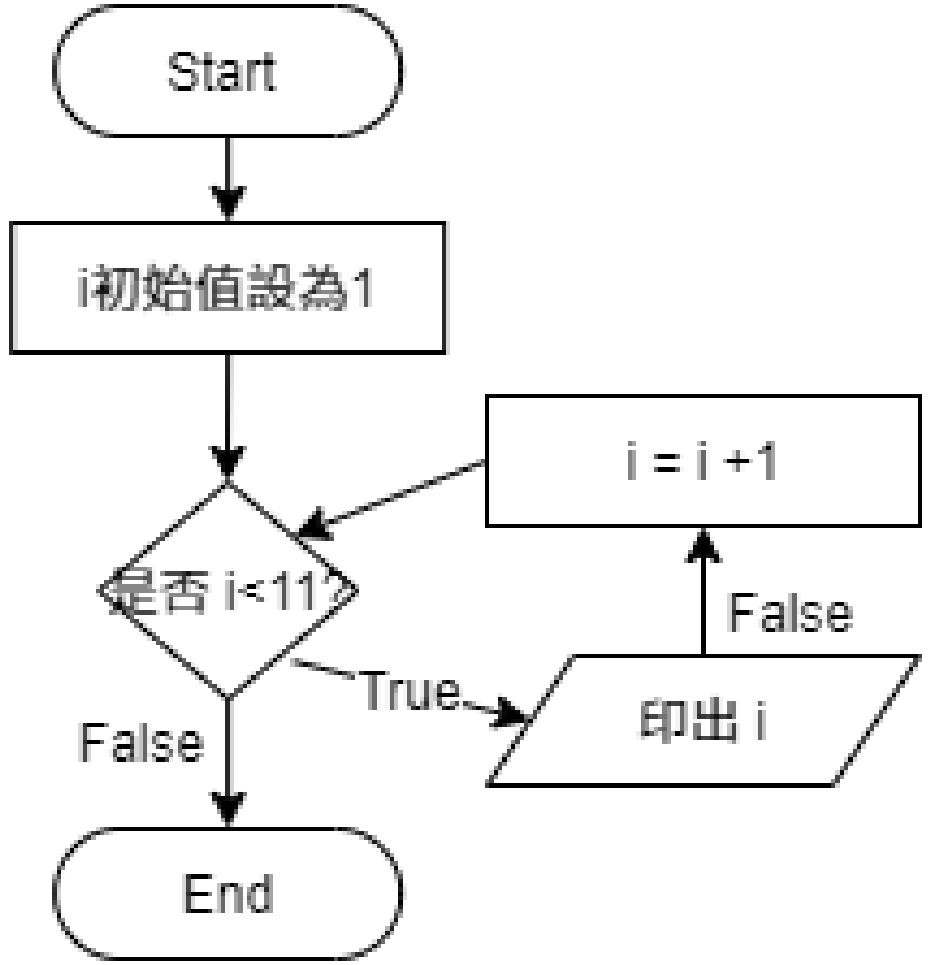
再詳細說明前，
先看看用這兩種方法如何解決以下問題

請從 1 印到 10

流程



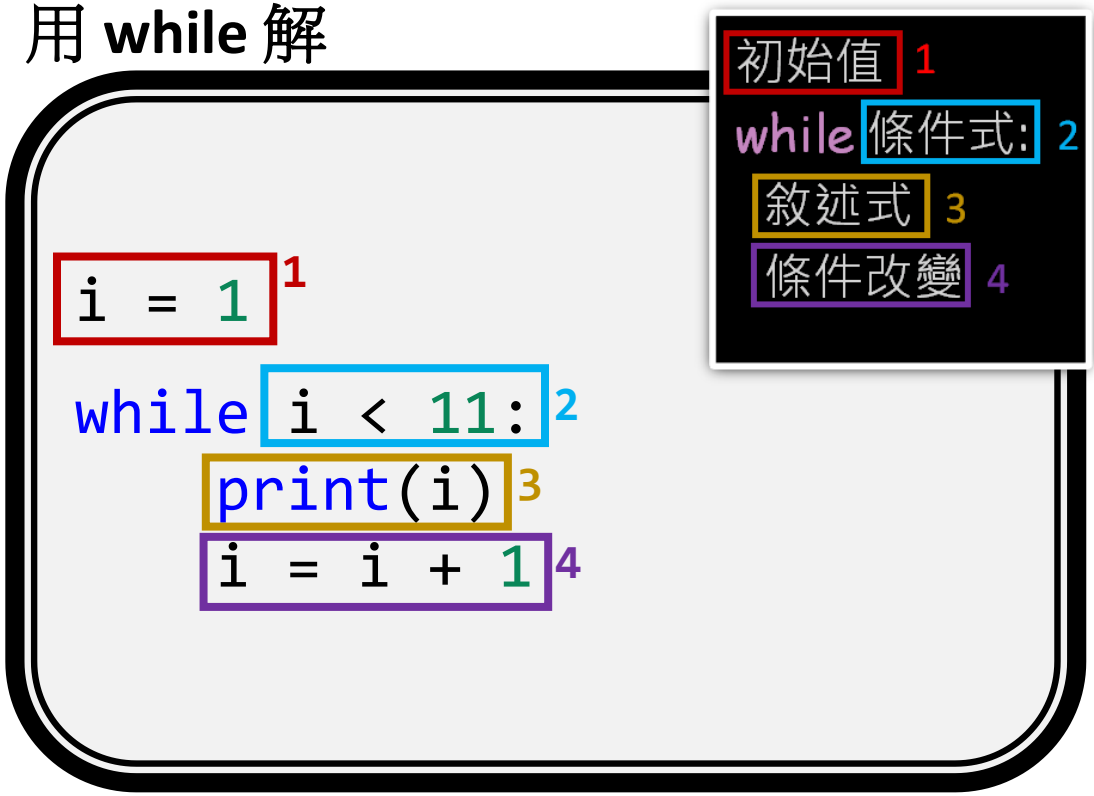
UML流程圖



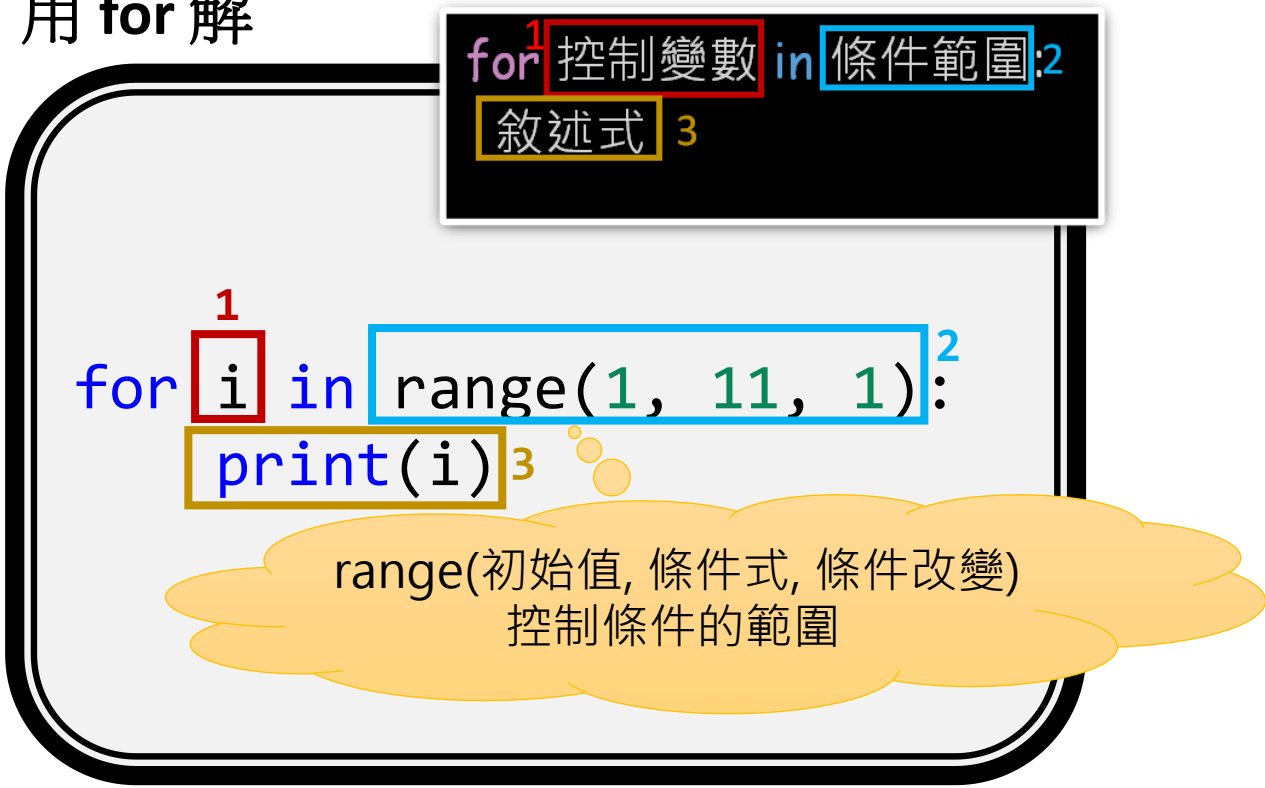
傳統流程圖

1 印到 10

用 while 解



用 for 解



```
初始值
while 條件式:
  敘述式
  條件改變
```

```
for 控制變數 in 條件範圍:
  敘述式
```

有觀察到 while 跟 for 的差異了嗎？

while vs. for

- while 分四種部分:
  - 初始值
  - **條件式**
  - 敘述式
  - 條件改變

只要能使用條件式的寫法，就可以使用 while

```
初始值
while 條件式:
  敘述式
  條件改變
```

- for 分三種部分:
  - 控制變數
  - **條件範圍**
  - 敘述式

能將迴圈範圍對象用條件範圍的寫法描述，就可以使用 for

```
for 控制變數 in 條件範圍:
  敘述式
```

for 的**簡潔度**高

條件改變與條件式全都放在條件範圍內

for 的**易讀性**高

while 通常要搭配內部敘述才可確認迴圈內容

while 的**應用範圍**比 for 大

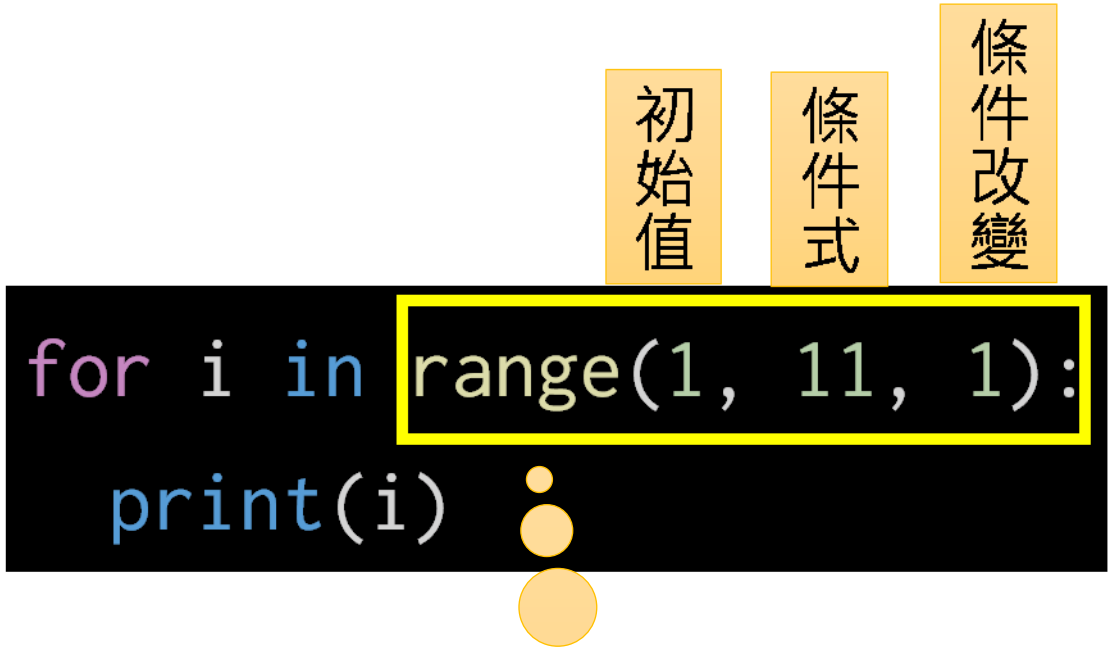
不知道結束時間，只知道結束條件時，for 就不適用

while 的**危險性**比 for 大

當 while 的條件式設定有誤時，可能會造成非預期的無窮迴圈

不同情境有不同適用情況，while 與 for 沒有絕對的誰好誰壞喔！

range() 控制條件範圍



range() 可視為一個等差級數列表表達式

- 首項 => 初始值
- 末像 => 條件式
- 公差 => 條件改變

一些比較少用到的舊知識再整理

拒絕換行

- 可以加上參數控制：end是控制換行的參數
- end參數用法如下：
  - print後面需要加上, end=""
- 以下程示範例參考

```
#拒絕換行程式碼範例
print("2-3=5", end="")
print("2-3=5")
```

執行結果

```
2-3=52-3=5
```

轉自輸出語句的章節

拒絕換行但是兩組資料之間想要有個空

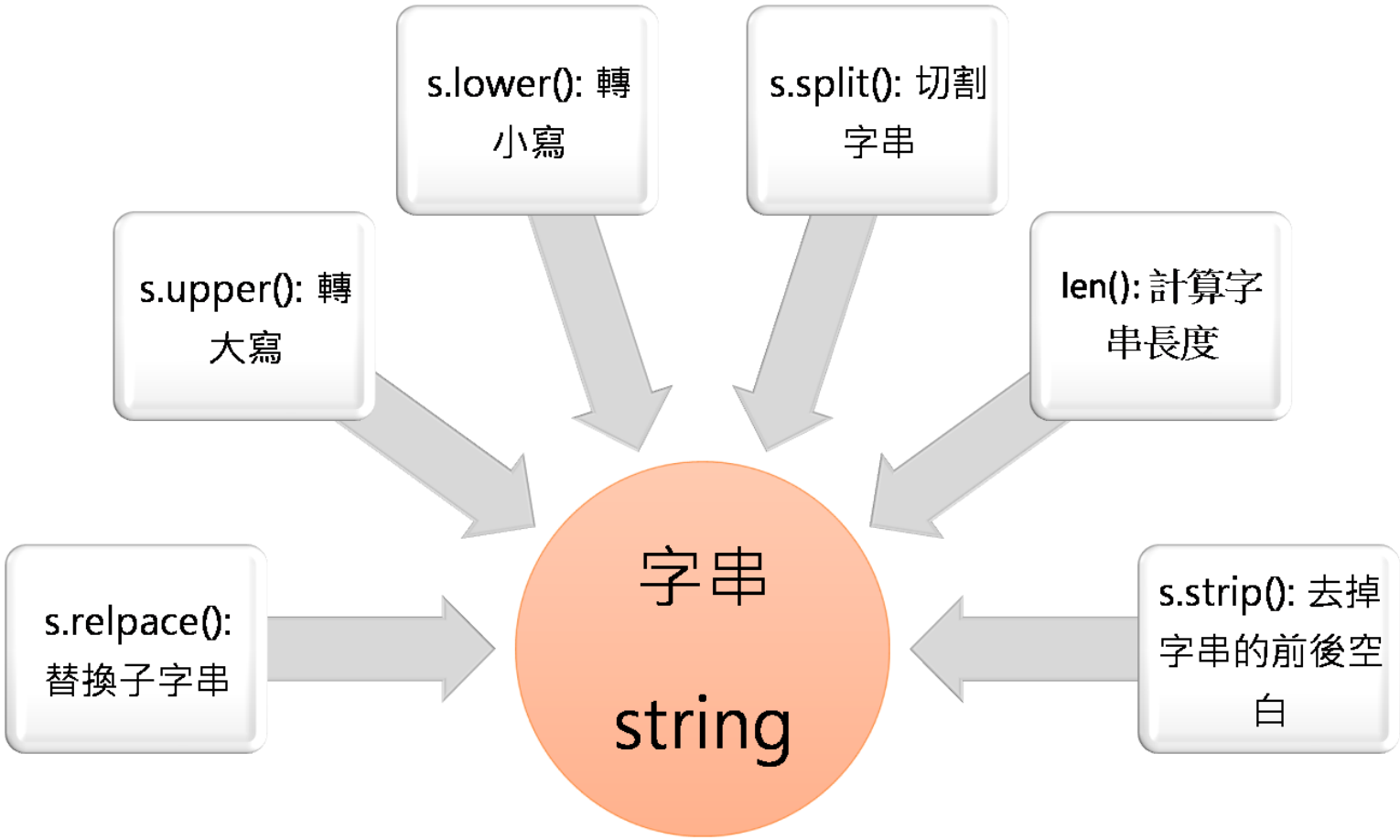
- 可以加上參數控制：end是控制換行的參數
- end參數用法如下：
  - print後面需要加上, end=" "
  - 雙引號之間加個空白
- 以下程示範例參考

中間是有一個空格的

```
#拒絕換行程式碼範例
print("2-3=5", end=" ")
print("2-3=5")
```

轉自輸出語句的章節

學習切割字串：split()



split(): 切割字串

寫一個程式，輸入一個句子如輸入範例，然後分別印出每一個子字串與索引值1的子字串。

輸入範例:

I love you

輸出範例:

['I', 'love', 'you']
love

程示範例如下圖

```
1  text = input()          #input得到一個字串
2  text1 = text.split(" ")
3  text2 = text.split(" ")[1]   #分割子字串
4  print(text1)            #印出索引1的子字串
5  print(text2)
```

```
[Python_6_9]$ python3 -d main.py | tee  main.py.err
I love you
['I', 'love', 'you']
love
[Python_6_9]$
```