

# Python研習課程 (I)

---

程式設計競賽研習

2018/10/20 ~ 21

# 講師介紹

呂怡緯

Email: seaniwei@yahoo.com.tw

Blog : <http://seansharingblog.blogspot.com/>

Mobile: 0973-033109



## 經歷:

- ❑ 祐能科技有限公司 研發經理
- ❑ 勞動部勞動力發展署 微型創業鳳凰輔導服務計畫創業顧問
- ❑ 國立高雄科技大學  
創新育成中心 顧問

## 專長:

- ❑ 企業諮詢與診斷輔導
- ❑ 政府計畫與企業研發規劃
- ❑ 電子商務、網路行銷
- ❑ 資訊系統分析、設計與開發

# 2018 Python全國大專校院程式設計競賽

## ❑ 競賽方式：

- ❑ 實地上機競賽
- ❑ 時間 100分鐘
- ❑ 共 9 大題，第一至九類各考一題，除第四題 20 分外，其餘每題 10 分，總計 100 分
- ❑ 於競賽時間內依題目要求完成程式碼，存檔提交程式碼



<http://www.tqcplus.org.tw/python2018/>

# 2018 Python全國大專校院程式設計競賽

## ❑ 競賽環境：

- ❑ 作業系統：Windows 10 (64位元)
- ❑ 套件包：Anaconda 5.1
- ❑ Python版本：Python 3.6
- ❑ 程式編輯器：[Spyder](#)及Visual Studio Code

# 2018 Python全國大專校院程式設計競賽

## ❑ 排名方式：

- ❑ 成績相同時以剩餘時間多者為優勝；如剩餘時間相同時，以第一題開始依序比較成績。

## ❑ 答題注意事項：

- ❑ A. 輸入與輸出的格式必須完全相同，每一行字、空白都要一樣。(若題目不要求顯示輸入提示則不需要顯示)
- ❑ B. 特別注意輸出後有無空白與換行。
- ❑ C. 注意全型與半型字元。
- ❑ 請參照 TQC+ Python全國大專校院程式設計競賽 - [競賽說明](#)

# 2018 Python全國大專校院程式設計競賽

## □ TQC+ Python認證技能規範：

類別	技能內容
第一類	基本程式設計：變數、運算子、輸出函式與格式化輸出、基本函式
第二類	選擇敘述：if 敘述、條件比較與邏輯判斷
第三類	迴圈敘述：while 迴圈、for 迴圈、巢狀迴圈
第四類	進階控制流程：迴圈應用、break 與 continue 敘述控制迴圈
第五類	函式(Function)：自訂函式、回傳值、預設參數、全域 / 區域變數

# 2018 Python全國大專校院程式設計競賽

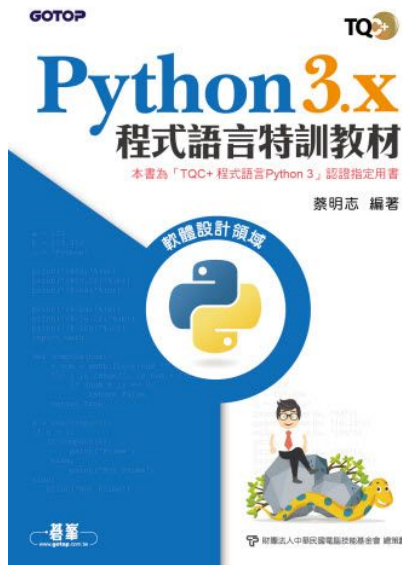
□ TQC+ Python認證技能規範：

類別	技能內容
第六類	串列(List): 一維、二維串列、List functions 和應用
第七類	數組(Tuple)、集合(Set)以及詞典(Dictionary): 定義、建立、方法和應用
第八類	字串(String): 建立字串、字串處理函式與應用
第九類	檔案與異常處理: 文字/二進位檔案建立、檔案寫入與讀取、異常處理

# 課程教材

## ❑ Python 3.x 程式語言特訓教材

- ❑ 作者：蔡明志，財團法人中華民國電腦技能基金會 總策劃
- ❑ 出版社：碁峰
- ❑ 出版日期：2018/4/18



## CODE JUDGER

Code Judger 是由 Kyosetai 共生智能股份有限公司所開發之自動化批改及教學管理系統，讓學生們在解題中學習，獲得成就，整合題庫與課程概念，為學習程式的學員、解題挑戰者以及程式教師提供最佳化的課程與題目管理。

台灣總代理：財團法人中華民國電腦技能基金會  
客服專線：(02)2577-8806#735  
Email：code\_judger@mail.csf.org.tw

## ❑ TQC+ Code Judger學習平台

- ❑ [www.codejudger.com](http://www.codejudger.com)



# 第一類：基本程式設計

- ❑ 變數
- ❑ 運算子
- ❑ 輸出函式與格式化輸出
- ❑ 基本函式

Python的設計哲學是「優雅」、「明確」、「簡單」

「用一種方法, 最好是只有一種方法來做一件事」

Python 創始人

吉多·范羅蘇姆 (Guido van Rossum)



# Python Variables

## ❑ Python 變數命名規則

- ❑ Identifiers 識別字命名: 英文字母 + 數字 + 底線(\_)
- ❑ 大小寫不同 (case-sensitive)
- ❑ 不可以數字為首
- ❑ 不可使用 Python Keywords (保留字)
- ❑ 可以使用中文 (但不建議!)
- ❑ 命名通則: 一般變數用小寫英文單字, 且單字間用底線符號相連

e.g., stu\_name, course\_info

# Python Variables

## ❏ Python Keywords 保留字/關鍵字

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

# Python Variables

## ❑ 合法的 Python 變數名

- ❑ `_` # 只有一個底線符號當變數名
- ❑ `apple8`
- ❑ `_apple`
- ❑ `iLikeApple`
- ❑ `i_Like_Apple_Very_Much`
- ❑ `我` # 中文變數名, 可以但不建議



# Python Variables

## ❑ 不合法的 Python 變數名

- ❑ 7eleven      # 不可以數字開頭
- ❑ 48            # 不可以數字開頭, 只有數字當然也不可以
- ❑ apple-x      # 符號只能用底線符號(\_)
- ❑ John&Mary   # 符號只能用底線符號(\_)
- ❑ apple x      # 不可以有空白
- ❑ yield        # 保留字



# Python Variables

## ❑ Variable assignment (變數指定)

- ❑ 不需宣告即可指定使用
- ❑ 不需指定資料型態
- ❑ (=) 指定運算子進行變數指定

```
a=1          # 不需指定資料型態  
a,b,c=0,1,2  # 多重指定 (multiple assignment)  
x=y=z=10     # 多重指定 (multiple assignment)
```

# Python Comments

## ❑ Comments (註解)

- ❑ ( # )單行註解
- ❑ ( """ )三個雙(單)引號是多行註解
- ❑ 三個引號同時是多行字串指定方式, 建議使用

( # )單行註解就好, 避免混淆

```
a=1          # 單行註解
```

```
"""
```

3個引號是多行註解,  
包在兩組3個引號間  
都會被視為註解,  
不管幾行。

```
"""
```

```
# 以下指定多行字串
```

```
helloMessage="""
```

```
*****
```

歡迎來到 Python 的世界  
相信您一定會喜歡上 Python

```
*****
```

```
"""
```



# Python Math Operators

## ❑ 算術運算子及優先序

Operator	Operation	Example	Evaluates to...
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4

# Python Math Operators

## ❑ 算術複合指定運算子

Python 沒有 `i++`, `--i`

前置遞增/遞減運算子

後置遞增/遞減運算子

運算	算術指定運算子	範例	相當於
加	<code>+=</code>	<code>a += 2</code>	<code>a = a + 2</code>
減	<code>-=</code>	<code>a -= 2</code>	<code>a = a - 2</code>
乘	<code>*=</code>	<code>a *= 2</code>	<code>a = a * 2</code>
除	<code>/=</code>	<code>a /= 2</code>	<code>a = a / 2</code>
商	<code>//=</code>	<code>a //= 2</code>	<code>a = a // 2</code>
餘數	<code>%=</code>	<code>a %= 2</code>	<code>a = a % 2</code>
次方	<code>**=</code>	<code>a **= 2</code>	<code>a = a ** 2</code>

# Python Data Types

## ❑ 基本資料型態

Numbers 數值	int 整數      ex. a=1, b=23454 沒有大小限制 float 浮點數    ex. c=123.12345 小數有效15位 complex 複數    ex. c = 1+2j
Strings 字串	"", 'a', 'This is a string', "This is a string"
Boolean 布林	True(真), False(偽)

# Python Data Types

## ❑ 抽象資料型態

- ❑ List (串列)
- ❑ Tuple (數組/元組)
- ❑ Set (集合)
- ❑ Dictionary (詞典)

# Python Data Types

## ❑ Numbers 數值

- ❑ int (整數) - 數值沒有小數點, 就是整數
- ❑ float (浮點數) - 只要有小數點, 就是浮點數
- ❑ 整數和浮點數的型態轉換 (type conversion)
  - ❑ 四則運算中有使用到浮點數, 結果就會變成浮點數
  - ❑ 除法會自動變成浮點數

# Python Data Types

## ❏ Numbers 數值

```
a=1.          # 指定變數a的值是1. (注意這裡僅有小數點)

print(a)      # 只要有小數點, 就是浮點數

print(1+1.0)  # 和浮點數作四則運算, 結果會變浮點數

print(2-1.0)  # 和浮點數作四則運算, 結果會變浮點數

print(2*1.0)  # 和浮點數作四則運算, 結果會變浮點數

print(2/1.0)  # 和浮點數作四則運算, 結果會變浮點數

print(4/2)    # 進行除法, 即使是整除也還是會變浮點數
```

# Python Data Types

## ❑ Strings 字串

- ❑ (') / (") 單/雙引號 (quote) 包括起來就是字串的設定 (assignment)
- ❑ (+) 加號是字串的連接 (concatenation), 注意加號的兩邊都必須是字串
- ❑ (\*) 乘號是字串的重複 (replication), \*5 就是重複5次

# Python Data Types

## ❏ Strings 字串

<code>a='a'</code>	# 1個字元也是字串
<code>b='This is string'</code>	# 使用單引號ok
<code>c="This is string, too"</code>	# 使用雙引號ok
<code>print(a+' and b')</code>	# (+) 加號將字串連接
<code>print(b*3)</code>	# (*) 乘號將字串重複
<code># print(a+3)</code>	# 字串加號的兩邊都必須是字串, 不 ok



# Python Output Formatting

## ❑ print() 輸出函式

- ❑ `print(*objects, sep=' ', end='\n')`
- ❑ `*objects`: 字串物件
- ❑ `sep=' '`: 多個輸出間分隔, 預設為單個空白符號
- ❑ `end='\n'`: 輸出結束, 預設為 `('\n')` 換行符號

```
a = 123
b = 123.456
c = 'Python'
print(a,b,c)
print(a,b,c,sep='*') # 輸出分隔為*號
print(a,end='')       # 輸出結束不換行
print(b,end='\t')     # 輸出結束以tab分隔
print(c)
```

# Python Output Formatting

## ❑ print() 格式化輸出

❑ %

- ❑ d 整數
- ❑ f 浮點數
- ❑ s 字串
- ❑ x 十六進位
- ❑ o 八進位
- ❑ - 靠左對齊 (預設均為靠右對齊)

```
a = 123
b = 123.456
c = 'Python'

print('%5d'%(a))      # 5位整數, 預設靠右對齊
print('%-5d'%(a))     # 5位整數, 靠左對齊
print('%10.2f'%(b))   # 10+2位浮點數, 預設靠右對齊
print('%-10.2f'%(b))  # 10+2位浮點數, 靠左對齊
print('%10s'%(c))      # 10位字串, 預設靠右對齊
print('%-10s'%(c))     # 10位字串, 靠左對齊
```

# Python Output Formatting

## ❑ print() 格式化輸出

### ❑ str.format()

- ❑ d 整數
- ❑ f 浮點數
- ❑ s 字串
- ❑ x 十六進位
- ❑ o 八進位
- ❑ b 二進位
- ❑ < 靠左對齊 (字串預設)
- ❑ > 靠右對齊 (數值預設)

```
a = 123
```

```
b = 123.456
```

```
c = 'Python'
```

```
print(format(a, '5d'))           # 5位整數, 數值預設靠右對齊
```

```
print(format(a, '<5d'))          # 5位整數, 靠左對齊
```

```
print(format(a, '5x'))          # 輸出十六進位
```

```
print(format(b, '<10.2f'))        # 10+2位浮點數, 靠左對齊
```

```
print(format(c, '10s'))         # 10位字串, 字串預設靠左對齊
```

```
print(format(c, '>10s'))         # 10位字串, 靠右對齊
```

# Python Output Formatting

## ❑ print() 多筆資料格式化輸出

```
a=123
```

```
b=123456
```

```
c=12
```

```
d,e,f=12,123,123456
```

```
print(format(a,'8d'),format(b,'8d'),format(c,'8d'))
```

# 使用 format()

```
print(format(d,'8d'),format(e,'8d'),format(f,'8d'))
```

```
print('{0:8d} {1:8d} {2:8d}'.format(a,b,c))
```

# format()另一種寫法

```
print('{0:8d} {1:8d} {2:8d}'.format(d,e,f))
```

# Python Output Formatting

## ❏ print() 多筆資料格式化輸出

```
a=123
```

```
b=123456
```

```
c=12
```

```
d,e,f=12,123,123456
```

```
print('{0:8d} {1:8d} {2:8d}'.format(a,b,c))
```

 # format() 另一種寫法

```
print('%8d %8d %8d'%(a,b,c))
```

 # 使用 %

```
print('%8d %8d %8d'%(d,e,f))
```

# Python Input

## ❑ input() 輸入函式

- ❑ 由鍵盤輸入取得資料, 皆為字串型態
- ❑ 轉換資料型態
  - ❑ int() 轉換為整數型態
  - ❑ float() 轉換為浮點數型態
  - ❑ eval() 轉換為數值(整數、浮點數)
  - ❑ str() 轉換為字串型態

```
a=int(input('Enter a number: '))  
a+=10  
print(a)  
# 輸入多筆需使用 eval()  
a,b=eval(input('Enter two numbers: '))  
print('a={:3d}, b={:3d}'.format(a,b))  
print('a=%3d, b=%3d'%(a,b))
```

# Python Built-in Functions

## ❑ 內建函式

### ❑ Built-in Functions List

<code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

# Python Built-in Functions

## ❑ 內建函式

函式	功能	範例	範例結果
<code>abs(x)</code>	取得 x 的絕對值	<code>abs(-5)</code>	5
<code>chr(x)</code>	取得整數 x 的字元	<code>chr(65)</code>	A
<code>divmod(x, y)</code>	取得 x 除以 y 的商及餘數的元組	<code>divmod(44, 6)</code>	(7,2)
<code>float(x)</code>	將 x 轉換成浮點數	<code>float("56")</code>	56.0
<code>hex(x)</code>	將 x 轉換成十六進位數字	<code>hex(34)</code>	0x22
<code>int(x)</code>	將 x 轉換成整數	<code>int(34.21)</code>	34
<code>len(x)</code>	取得元素個數	<code>len([1,3,5,7])</code>	4
<code>max( 參數串列 )</code>	取得參數中的最大值	<code>max(1,3,5,7)</code>	7
<code>min( 參數串列 )</code>	取得參數中的最小值	<code>min(1,3,5,7)</code>	1



# Python Built-in Functions

## ❑ 內建函式

函式	功能	範例	範例結果
oct(x)	將 x 轉換成八進位數字	oct(34)	0o42
ord(x)	回傳字元 x 的 Unicode 編碼值	ord("我")	25105
pow(x, y)	取得 x 的 y 次方	pow(2,3)	8
round(x)	以四捨六入法取得 x 的近似值	round(45.8)	46
sorted( 串列 )	由小到大排序	sorted([3,1,7,5])	[1,3,5,7]
str(x)	將 x 轉換成字串	str(56)	56 ( 字串 )
sum( 串列 )	計算串列元素的總和	sum([1,3,5,7])	16
type( 物件 )	取得物件的資料型態	type(34.0)	float

# Python math module

## ❏ math 數學模組

❏ `import math`

函式	描述
<code>fab(x)</code>	以浮點數回傳 $x$ 的絕對值
<code>ceil(x)</code>	回傳大於 $x$ 的最小整數
<code>floor(x)</code>	回傳小於 $x$ 的最大整數
<code>exp(x)</code>	回傳 $e^x$
<code>log(x)</code>	回傳 $\ln(x)$
<code>log(x, base)</code>	回傳指定基底的對數值
<code>sqrt(x)</code>	回傳 $x^{1/2}$

# Python math module

## ❑ math 數學模組

❑ `import math`

函式	描述
<code>sin(x)</code>	回傳以弧度為單位的正弦函數值
<code>asin(x)</code>	回傳以弧度為單位的反正弦函數值
<code>cos(x)</code>	回傳以弧度為單位的餘弦函數值
<code>acos(x)</code>	回傳以弧度為單位的反餘弦函數值
<code>tan(x)</code>	回傳以弧度為單位的正切函數值
<code>degree(x)</code>	將 x 角度從弧度(radian)轉為度數(degree)
<code>radian(x)</code>	將 x 角度從度數轉為弧度

# 例題：浮點數格式化輸出

- ❑ 請撰寫一程式，輸入四個分別含有小數 1 到 4 位的浮點數，然後將這四個浮點數以欄寬為 7、每列印兩個的方式，先列印向右靠齊，再列印向左靠齊，左右皆以直線 | (Vertical bar) 作為邊界。

- ❑ 提示：輸出浮點數到小數點後第二位。

範例輸入

23.12

395.3

100.4617

564.329

範例輸出

| 23.12 395.30|

| 100.46 564.33|

|23.12 395.30 |

|100.46 564.33 |

```
num1 = eval(input())    # 輸入整數, 題目沒有要求輸入提示
```

```
num2 = eval(input())
```

```
num3 = eval(input())
```

```
num4 = eval(input())
```

```
print("|%7.2f %7.2f|" % (num1, num2))    # 預設靠右對齊
```

```
print("|%7.2f %7.2f|" % (num3, num4))
```

```
print("|%-7.2f %-7.2f|" % (num1, num2))  # (-) 靠左對齊
```

```
print("|%-7.2f %-7.2f|" % (num3, num4))
```

# 例題：浮點數格式化輸出

範例輸入

```
23.12
395.3
100.4617
564.329
```

範例輸出

```
| 23.12 395.30|
| 100.46 564.33|
|23.12 395.30 |
|100.46 564.33 |
```

```
num1 = eval(input())    # 輸入整數, 題目沒有要求輸入提示
```

```
num2 = eval(input())
```

```
num3 = eval(input())
```

```
num4 = eval(input())
```

```
a,b,c,d=eval(input('a,b,c,d= ')) # eval() 可取得多筆輸入
```

注意範例輸入是分開輸入

## 例題：字串格式化輸出

- 請撰寫一程式，輸入四個單字，然後將這四個單字以欄寬為10、欄與欄間隔一個空白字元、每列印兩個的方式，先列印向右靠齊，再列印向左靠齊，左右皆以直線 | (Vertical bar) 作為邊界。

範例輸入

```
I
enjoy
learning
Python
```

範例輸出

```
|          I          enjoy|
|  learning        Python|
|I              enjoy   |
|learning    Python    |
```

```
w1=input()          # 輸入字串
w2=input()
w3=input()
w4=input()

print('|%10s  %10s|'%(w1,w2))    # 預設靠右對齊
print('|%10s  %10s|'%(w3,w4))    # 預設靠右對齊
print('|%-10s  %-10s|'%(w1,w2))  # (-) 靠左對齊
print('|%-10s  %-10s|'%(w3,w4))  # (-) 靠左對齊
```

# 例題：字串格式化輸出

- 請撰寫一程式，輸入四個單字，然後將這四個單字以欄寬為10、欄與欄間隔一個空白字元、每列印兩個的方式，先列印向右靠齊，再列印向左靠齊，左右皆以直線 | (Vertical bar) 作為邊界。

範例輸入

```
I
enjoy
learning
Python
```

範例輸出

```
|          I          enjoy|
|  learning        Python|
|I              enjoy   |
|learning        Python  |
```

```
w1=input()           # 輸入字串
w2=input()
w3=input()
w4=input()
print('|{:>10s} {:>10s}|'.format(w1,w2)) # (>)靠右對齊
print('|{:>10s} {:>10s}|'.format(w3,w4))
print('|{:<10s} {:<10s}|'.format(w1,w2)) # (<)靠左對齊
print('|{:<10s} {:<10s}|'.format(w3,w4))
```

# 例題：圓形面積計算

❑ 請撰寫一程式，輸入一圓的半徑，並加以計算此圓之面積和周長，最後請印出此圓的半徑(Radius)、周長(Perimeter)和面積(Area)。

❑ 提示 1: 需 import math 模組，並使用 math.pi。

❑ 提示 2: 輸出浮點數到小數點後第二位。

若不使用pow(), 可以怎麼作？

範例輸入1

10

範例輸出1

```
Radius = 10.00  
Perimeter = 62.83  
Area = 314.16
```

```
import math  
PI=math.pi    # 回傳常數  $\pi = 3.141592...$   
radius=float(input())  
print("Radius = %.2f" % radius)  
print("Perimeter = %.2f" % (2*radius*PI))  
print("Area = %.2f" % (pow(radius,2)*PI))
```



# 例題：正n邊形面積計算

❑ 請撰寫一程式，讓使用者輸入兩個正數n、s，代表正n邊形之邊長為s，計算並輸出此正n邊形之面積 (Area)。

- ❑ 提示1: 需import math模組，並使用math.pow及math.tan
- ❑ 提示2: 正n邊形面積的公式如下： $\text{Area} = (n * s^2) / (4 * \tan(\pi/n))$
- ❑ 提示3: 輸出浮點數到小數點後第四位

範例輸入

8  
6

範例輸出

Area = 173.8234

```
import math
n=eval(input())           # 輸入數值n邊
s=eval(input())           # 輸入數值s邊長
area=(n*s**2)/(4*math.tan(math.pi/n))
print('Area = %.4f'%area) # 輸出n邊形面積
```

# 練習：攝氏華氏溫度轉換

- ❑ 請撰寫一程式，請使用者輸入華氏溫度，然後輸出其對應的攝氏溫度。
  - ❑ 提示： $\text{攝氏溫度} = (\text{華氏溫度} - 32) * 5 / 9$
  - ❑ 輸出格式：Fahrenheit `***. **` ---> Celsius `***. **`

# 練習：攝氏華氏溫度轉換

- ❑ 請撰寫一程式，請使用者輸入華氏溫度，然後輸出其對應的攝氏溫度。
  - ❑ 提示： $\text{攝氏溫度} = (\text{華氏溫度} - 32) * 5 / 9$
  - ❑ 輸出格式：Fahrenheit **\*\*\*. \*\*** ---> Celsius **\*\*\*. \*\***

```
f=eval(input())          # 輸入華氏溫度
c=(f-32)*5/9              # 攝氏溫度轉換
print('Fahrenheit %3.2f ---> Celsius %3.2f' % (f,c)) # 輸出
```

# 練習：正五邊形面積計算

- ❑ 請撰寫一程式，讓使用者輸入一個正數s，代表正五邊形之邊長，計算並輸出此正五邊形之面積(Area)。
  - ❑ 提示1: 需import math模組，並使用math.pow及math.tan
  - ❑ 提示2: 正五邊形面積的公式:  $\text{Area} = (5 * s^2) / (4 * \tan(\pi/5))$
  - ❑ 提示3: 輸出浮點數到小數點後第四位。

# 練習：正五邊形面積計算

❑ 請撰寫一程式，讓使用者輸入一個正數 $s$ ，代表正五邊形之邊長，計算並輸出此正五邊形之面積(Area)。

- ❑ 提示1: 需import math模組，並使用math.pow及math.tan
- ❑ 提示2: 正五邊形面積的公式:  $\text{Area} = (5 * s^2) / (4 * \tan(\pi/5))$
- ❑ 提示3: 輸出浮點數到小數點後第四位。

範例輸入

5

範例輸出

Area = 43.0119

```
import math
s=eval(input()) # 輸入5邊形邊長
area=5*math.pow(s,2)/(4*math.tan(math.pi/5))
print('Area = {:.4f}'.format(area)) # 輸出5邊形面積
```

## 第二類：選擇敘述

- ❑ if 敘述
- ❑ 條件比較與邏輯判斷

"Algorithms + Data Structures = Programs"

Niklaus Wirth (1934.2.15-)

1984 Turing Award

# Python Relational Operators and if Statement

## ❑ 關係運算子

Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

## ❑ if statement

```
if condition expression:  
    statement(s)  
  
elif condition expression:  
    statement(s)  
  
else:  
    statement(s)
```



# Python Logical Operators

## ❑ 邏輯運算子

運算子	意義	範例	範例結果
not	傳回與原來比較結果相反的值，即比較結果是 True，就傳回 False；比較結果是 False，就傳回 True。	not(3>5) not(5>3)	True False
and	只有兩個運算元的比較結果都是 True 時，才傳回 True，其餘情況皆傳回 False。	(5>3) and (9>6) (5>3) and (9<6) (5<3) and (9>6) (5<3) and (9<6)	True False False False
or	只有兩個運算元的比較結果都是 False 時，才傳回 False，其餘情況皆傳回 True。	(5>3) or (9>6) (5>3) or (9<6) (5<3) or (9>6) (5<3) or (9<6)	True True True False

# Condition expression 條件判斷

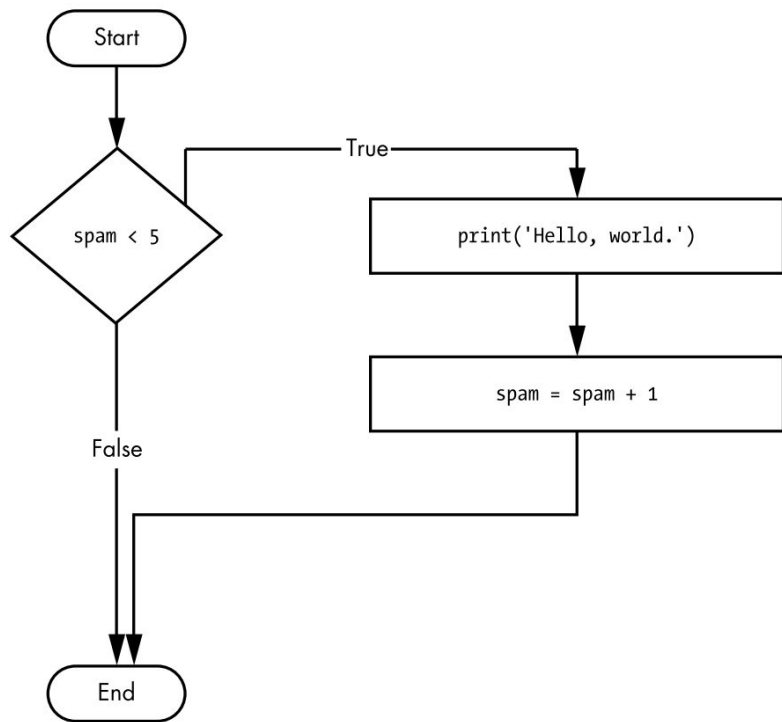
BMI	分類
< 18.5	過輕
18.5 - 23.9	正常
24.0 - 27.9	超重
≥ 28	肥胖

```
if bmi <= 23.9 and bmi >= 18.5:  
    print("正常")
```

```
if 18.5 <= bmi <= 23.9:  
    print("正常")
```



# 單一個 if 敘述



```
spam = 0
```

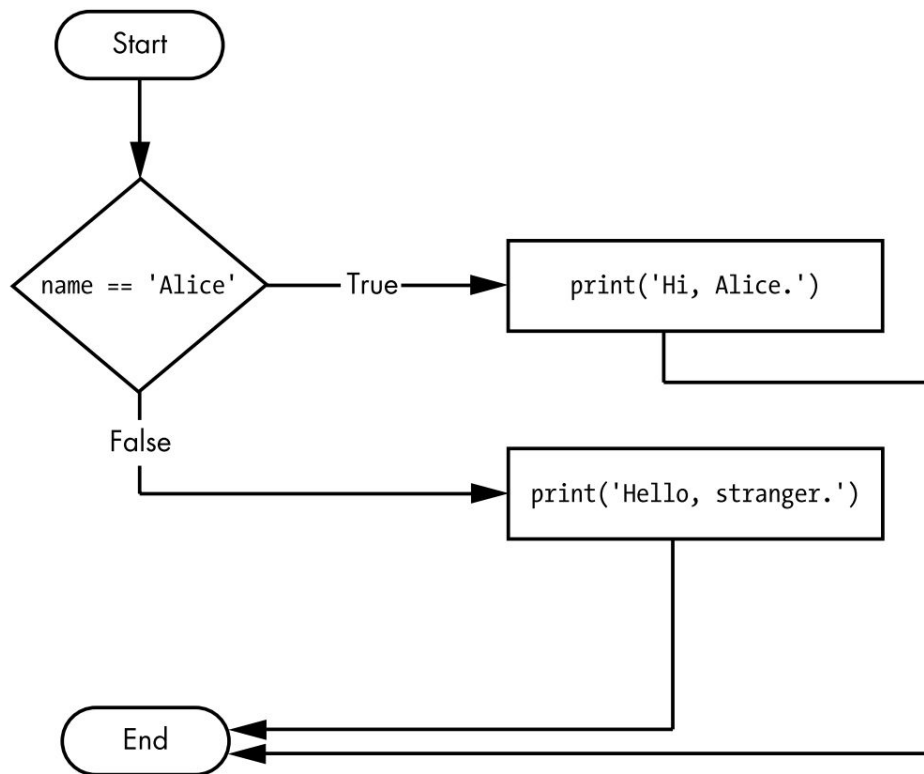
```
if spam < 5:
```

```
    print('Hello, world.')
```

```
    spam = spam + 1
```

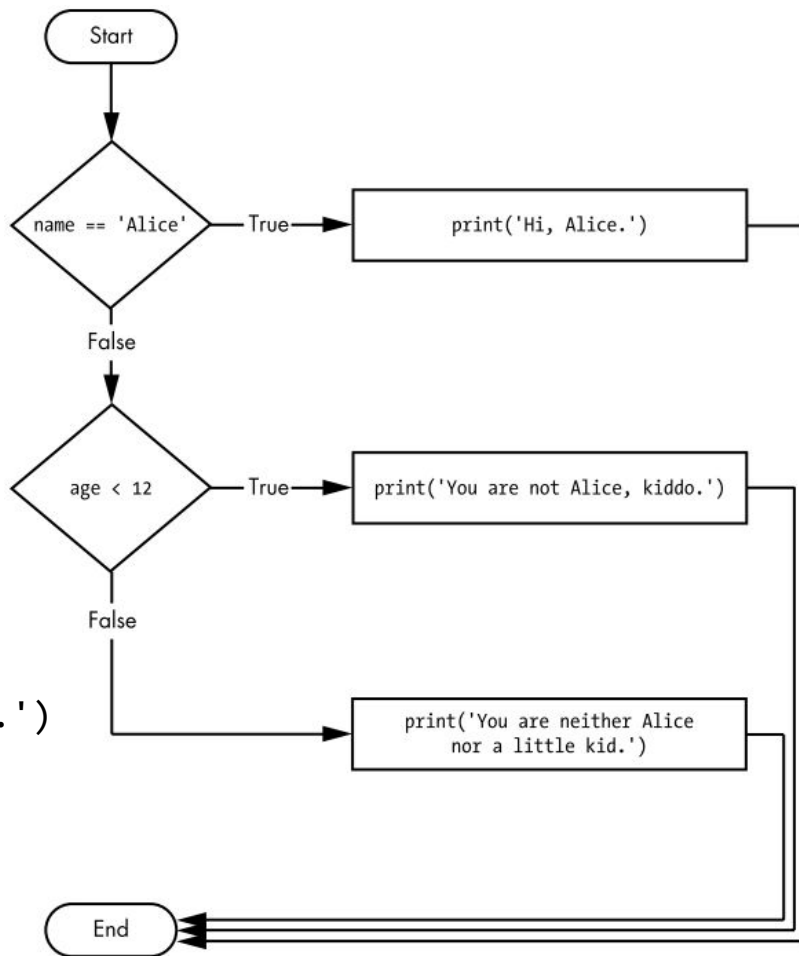
# if - else 敘述

```
name = 'Alice'  
if name == 'Alice':  
    print('Hi, Alice.')  
else:  
    print('Hello, stranger.')
```



# if - elif - else 敘述

```
name = 'Alice'
if name == 'Alice':
    print('Hi, Alice.')
elif age < 12:
    print('You are not Alice, kiddo.')
else:
    print('You are neither Alice nor a little kid.')
```



# if - elif 敘述

- 輸入：考試成績 ( 滿分100 )
- 輸出：若成績在
  - 90 分以上顯示「優等」，
  - 80 - 89 分顯示「甲等」，
  - 70 - 79 分顯示「乙等」，
  - 60 - 69 分顯示「丙等」，
  - 60 分以下顯示「丁等」。

```
score = 90
if 90 < score:
    print("優等")
elif 80 <= score < 90:
    print("甲等")
elif 70 <= score < 80:
    print("乙等")
elif 60 <= score < 70:
    print("丙等")
elif score < 60:
    print("丁等")
```

## 例題：倍數判斷

- 請使用選擇敘述撰寫一程式，讓使用者輸入一個正整數，然後判斷它是3或5的倍數，若此數值同時為3與5的倍數，顯示【x is a multiple of 3 and 5.】；如此數值皆不屬於3或5的倍數，顯示【x is not a multiple of 3 or 5.】，將使用者輸入的數值代入x。

範例輸入1

55

範例輸出1

55 is a multiple of 5.

```
n=int(input())
if (n%3==0) and (n%5==0):
    print("%d is a multiple of 3 and 5." % n)
elif n%3==0:
    print("%d is a multiple of 3." % n)
elif n%5==0:
    print("%d is a multiple of 5." % n)
else:
    print("%d is not a multiple of 3 or 5." % n)
```

## 例題: 204 算術運算

- ❑ 請使用選擇敘述撰寫一程式，讓使用者輸入兩個整數 a、b，然後再輸入一算術運算子(+、-、\*、/、//、%)，輸出這兩個數以及其經過運算後的結果。

範例輸入

30  
20  
\*

範例輸出1

600

```
a = eval(input())
b = eval(input())
opr = input() # 輸入的是字串
if opr=='+':
    print(a + b)
elif opr=='-':
    print(a - b)
elif opr=='*':
    print(a * b)
elif opr=='/':
    print(a / b)
elif opr=='//':
    print(a // b)
elif opr=='%':
    print(a % b)
```



## 例題：閏年判斷

- ❑ 請使用選擇敘述撰寫一程式，讓使用者輸入一個西元年份，然後判斷它是否為閏年 (leap year) 或平年。

- ❑ 提示：閏年判斷規則為，每四年一閏，每百年不閏，但每四百年也一閏。

範例輸入1

1992

範例輸出1

1992 is a leap year.

```
y=int(input())
if y%400==0 or (y%4==0 and y%100!=0):
    print(y,"is a leap year.")
else:
    print(y,"is not a leap year.")
```

# 例題：折扣方案

- ❑ 請使用選擇敘述撰寫一程式，要求使用者輸入購物金額，購物金額需大於8,000(含)以上，並顯示折扣優惠後的實付金額。

- ❑ 購物金額折扣方案如下表所示：

判斷順序可否反過來？

金額	折扣
8,000 (含) 以上	9.5折
18,000(含) 以上	9折
28,000(含) 以上	8折
38,000(含) 以上	7折

```
cost=eval(input())
if cost >= 38000:      # 先判斷>=38000
    pay=cost*0.7
elif cost >= 28000:
    pay=cost*0.8
elif cost >= 18000:
    pay=cost*0.9
elif cost >= 8000:
    pay=cost*0.95
print(pay)
```

## 例題：距離判斷

- ❑ 請使用選擇敘述撰寫一程式，讓使用者輸入一個點的平面座標x和y值，判斷此點是否與點(5, 6)的距離小於或等於15，如距離小於或等於15顯示【Inside】，反之顯示【Outside】。

❑ 提示：計算平面上兩點距離的公式： $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

範例輸入1

7  
20

範例輸出1

Inside

```
x=eval(input())
y=eval(input())
dist=((x-5)**2+(y-6)**2)**0.5    # 計算距離
if dist<=15:
    print("Inside")
else:
    print("Outside")
```

## 練習：奇偶數判斷

- ❑ 試撰寫一程式，利用亂數產生器產生介於 1~100 之間的亂數，然後檢視這個亂數是偶數(even number)或是奇數(odd number)。

範例輸入1

```
56
```

範例輸出1

```
56 is an even number.
```

## 練習：奇偶數判斷

- ❑ 試撰寫一程式，利用亂數產生器產生介於 1~100 之間的亂數，然後檢視這個亂數是偶數(even number)或是奇數(odd number)。

範例輸入1

56

範例輸出1

56 is an even number.

```
import random
n=random.randint(1,100)    # 產生1~100的整數
if n%2 == 0:
    print("%d is an even number." % n)
else:
    print("%d is an odd number." % n)
```

## 練習：距離判斷

- ❑ 試撰寫一程式，由使用者輸入點座標(x,y)，然後檢視該點是否位於中心點為(0,0)，半徑為 8 的圓內或圓外。
  - ❑ 提示：若點座標與中心點 (0,0)的距離小於或等於半徑 8，則位於圓內，否則位於圓外。

範例輸入1

```
7  
20
```

範例輸出1

```
Inside
```

## 練習：距離判斷

❑ 試撰寫一程式，由使用者輸入點座標(x,y)，然後檢視該點是否位於中心點為(0,0)，半徑為 8 的圓內或圓外。

❑ 提示：若點座標與中心點 (0,0)的距離小於或等於半徑 8，則位於圓內，否則位於圓外。

範例輸入1

7  
20

範例輸出1

Inside

```
x=eval(input())
y=eval(input())
dist=(x**2+y**2)**0.5  #計算距離
if dist <= 8:
    print("圓內")
else:
    print("圓外")
```

## 第三類：迴圈敘述

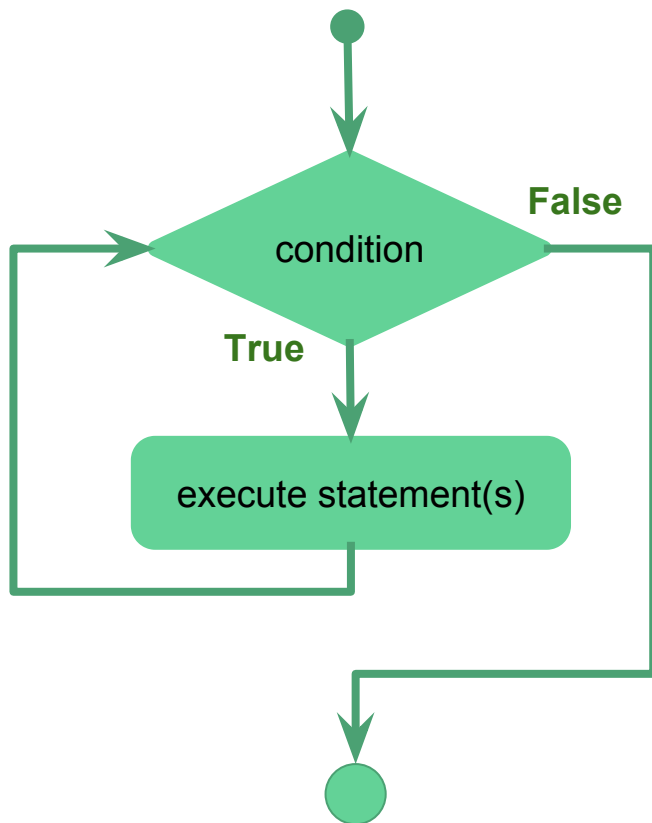
- ❑ while 迴圈
- ❑ for 迴圈
- ❑ 巢狀迴圈



# 迴圈敘述

Python Loops	Java Loops
for 迴圈 while 迴圈	for loop while loop do while for each <a href="#">java loops examples</a>

# 迴圈敘述



- ❑ for loop: 是否在某個範圍或某些項目中
- ❑ while loop: 是否符合某種條件判斷

# 迴圈敘述

## ❑ while statement

```
while condition expression:  
    statement(s)
```

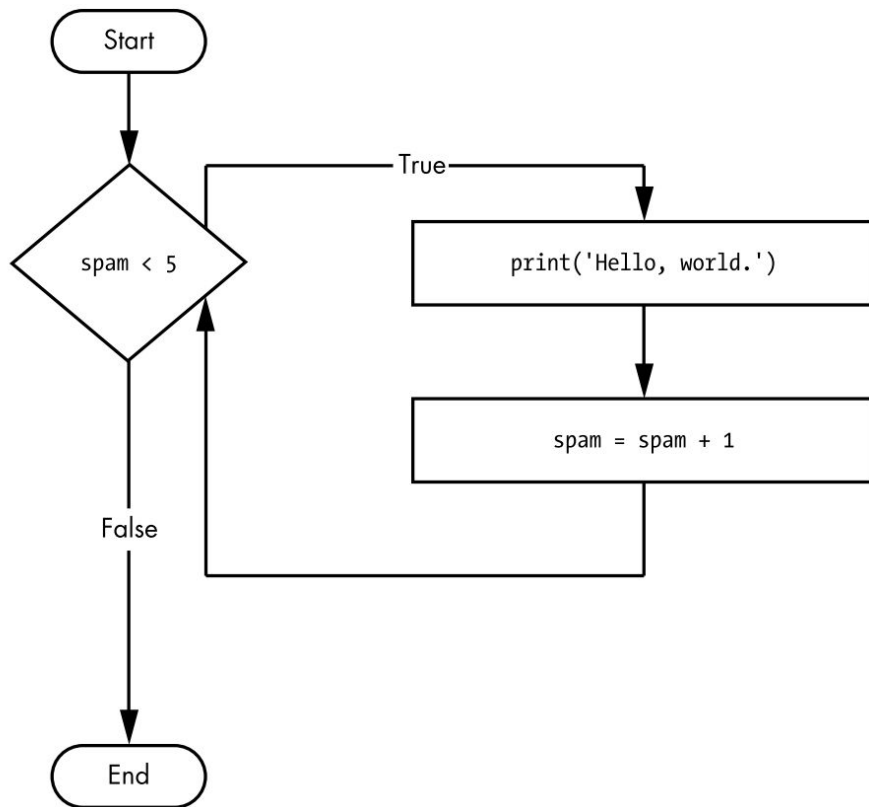
# Example

```
spam = 0
```

```
while spam < 5:
```

```
    print('Hello, world.')
```

```
    spam = spam + 1    # 設定可終止條件
```



# 迴圈敘述

```
spam = 0
while spam < 5:
    print('Hello, world.')
    spam = spam + 1
print(spam)
```

```
spam = 0
if spam < 5:
    print('Hello, world.')
    spam = spam + 1
print(spam)
```

while loop 和 if 差別在那裡？

# 迴圈敘述

❑ while statement

```
while condition expression:  
    statement(s)
```

# while statement example

```
i=1
```

```
while i <= 100:
```

```
    print("i=",i)
```

```
    i+=1  # 設定可終止的條件
```

```
while True:
```

```
    print("Hello World!")
```

```
    # 是否Ok? 會發生什麼事?
```

# 迴圈敘述

❑ `for...in... statement`

`for an item in a sequence:`

`statement(s)`

an item: 變數

a sequence: `range()`, 字串, `list`, `tuple`...

`range()` 為 Python 內建函式, 建立整數數列

語法: `range(stop)`, `range(start, stop[, step])`

`range(10)`

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

`range(1, 11)`

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

`range(0, 30, 5)`

0, 5, 10, 15, 20, 25

`range(0, -10, -1)`

0, -1, -2, -3, -4, -5, -6, -7, -8, -9

# 迴圈敘述

❏ `for...in... statement`

`for an item in a sequence:`  
    `statement(s)`

```
for i in range(10, 15):  
    print('i=', i)
```

```
for i in range(0, 10, 2):  
    print('i=', i)
```

```
for i in range(5, -1, -1):  
    print('i=', i)
```

# 迴圈敘述

## for 迴圈

```
total=0
for i in range(5):
    total=total+i
    print('i=', i,', total=', total)
print('sum=', total)
```

## while 迴圈

```
i=0    # i for while loop condition
total=0
while i < 5:
    total=total+i
    print('i=', i,', total=', total)
    i=i+1
print('sum=', total)
```



# Nested Loop

## ❑ 巢狀迴圈

- ❑ 又稱多重迴圈，迴圈中還有迴圈。

```
# Nested loop example
```

```
for i in range(1,10):
```

```
    for j in range(1,10):
```

```
        print("%2d*%2d=%2d" % (i,j,i*j), end=" ")
```

```
    print()
```

## 例題: 302 迴圈偶數連加

- 請使用迴圈敘述撰寫一程式，讓使用者輸入兩個正整數 $a$ 、 $b$  ( $a < b$ )，利用迴圈計算從  $a$  開始的偶數連加到  $b$  的總和。例如：輸入 $a=1$ 、 $b=100$ ，則輸出結果為2550( $2 + 4 + \dots + 100 = 2550$ )。

範例輸入

```
14
1144
```

範例輸出

```
327714
```

```
a = int(input())
b = int(input())
ans = 0
for i in range(a, b+1):
    if i % 2 == 0: # 偶數才加總
        ans += i
print(ans)
```

## 例題：304 迴圈倍數總和

- ❑ 請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數a，利用迴圈計算從1到a之間，所有5之倍數數字總和。

範例輸入

21

範例輸出

50

```
# 迴圈倍數總和
num = int(input())
ans = 0
for i in range(1, num+1):
    if i % 5 == 0 : # 5的倍數才加總
        ans += i
print(ans)
```

## 例題：迴圈階乘計算

- ❑ 請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數 $n$ ，利用迴圈計算並輸出 $n!$ 的值。

範例輸入

15

範例輸出

1307674368000

```
# 迴圈階乘計算
```

```
num=int(input())    # 輸入正整數
```

```
factorial=1
```

```
for i in range(1,num+1):
```

```
    factorial *= i    # 1*2*3...
```

```
print(factorial)
```

## 例題：迴圈公式計算

- ❑ 請使用迴圈敘述撰寫一程式，讓使用者輸入正整數 $n$  ( $1 < n$ )，計算以下公式的總和並顯示結果：

$$\frac{1}{1 + \sqrt{2}} + \frac{1}{\sqrt{2} + \sqrt{3}} + \frac{1}{\sqrt{3} + \sqrt{4}} + \dots + \frac{1}{\sqrt{n-1} + \sqrt{n}}$$

- ❑ 提示：輸出結果至小數點後四位。

```
# 迴圈公式計算
n=eval(input()) # 輸入正整數
sum=0           # 計算總和變數
for i in range(2,n+1):
    sum+=1/((i-1)**0.5+i**0.5)
print("%.4f" % sum)
```

# 例題：存款總額計算

- ❑ 請使用迴圈敘述撰寫一程式，提示使用者輸入金額 (如10,000)、年收益率 (如5.75)，以及經過的月份數 (如5)，接著顯示每個月的存款總額。
  - ❑ 提示：四捨五入，輸出浮點數到小數點後第二位。
  - ❑ 舉例：假設存款\$10,000，年收益為5.75%。
  - ❑ 過了一個月，存款是： $10000 + 10000 * 5.75 / 1200 = 10047.92$
  - ❑ 過了兩個月，存款是： $10047.91 + 10047.91 * 5.75 / 1200 = 10096.06$

範例輸入

```
50000  
1.3  
5
```

範例輸出

Month	Amount
1	50054.17
2	50108.39
3	50162.68
4	50217.02
5	50271.42

```
total=eval(input())    # 輸入金額  
rate=eval(input())     # 輸入年收益  
months=eval(input())   # 輸入月份數  
print('%s \t %s' % ('Month', 'Amount'))  
for i in range(1, months+1):  
    total+=total*rate/1200  
    print('%3d \t %.2f' % (i, total))
```

# 例題：九九乘法表

❑ 請分別以 for 迴圈和 while 迴圈撰寫九九乘法表

```
# for loop 九九乘法表
for i in range(1,10):
    for j in range(1,10):
        print("{0} * {1} = {2:>2d} ".format(i,j,i*j),end="")
    print()
```

# 例題：九九乘法表

❑ 請分別以 for 迴圈和 while 迴圈撰寫九九乘法表

```
# while loop 九九乘法表
i=1
while i <= 9:
    j=1
    while j <= 9:
        print("{0} * {1} = {2:>2d} ".format(i,j,i*j),end="")
        j+=1
    i+=1
    print()
```



# 例題：九九乘法表

❑ 請分別以 for 迴圈和 while 迴圈撰寫九九乘法表

```
# for loop 九九乘法表
```

```
for i in range(1,10):
```

```
    for j in range(1,10):
```

```
        print(...)
```

```
    print()
```

```
# while loop 九九乘法表
```

```
i=1
```

```
while i <= 9:
```

```
    j=1
```

```
    while j <= 9:
```

```
        print(...)
```

```
        j+=1
```

```
    i+=1
```

```
    print()
```

# 練習：乘法表

- ❑ 請使用迴圈敘述撰寫一程式，要求使用者輸入一個正整數 $n$ ，顯示 $n*n$ 乘法表。
  - ❑ 提示：每個數字輸出欄寬為 2，且需靠左對齊不跳行，使橫向排列的列數即是正整數  $n$  的數，每項乘積需進行格式化排列整齊。

範例輸入1

3

範例輸出1

1 * 1 = 1	2 * 1 = 2	3 * 1 = 3
1 * 2 = 2	2 * 2 = 4	3 * 2 = 6
1 * 3 = 3	2 * 3 = 6	3 * 3 = 9

# 練習：乘法表

- ❑ 請使用迴圈敘述撰寫一程式，要求使用者輸入一個正整數 $n$ ，顯示 $n*n$ 乘法表。
  - ❑ 提示：每個數字輸出欄寬為 2，且需靠左對齊不跳行，使橫向排列的列數即是正整數  $n$  的數，每項乘積需進行格式化排列整齊。

範例輸入1

3

範例輸出1

1 * 1 = 1	2 * 1 = 2	3 * 1 = 3
1 * 2 = 2	2 * 2 = 4	3 * 2 = 6
1 * 3 = 3	2 * 3 = 6	3 * 3 = 9

# 乘法表

```
n=int(input()) # 輸入正整數
```

```
for i in range(1,n+1):
```

```
    for j in range(1,n+1):
```

```
        print("%-2d* %-2d= %-2d  " % (j,i,i*j), end="")
```

```
    print()
```

## 練習：迴圈數值相乘

- ❑ 請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數，然後以「倒」三角形的方式依序輸出此數的相乘結果。

- ❑ 提示：輸出欄寬為 4，且需靠右對齊。

```
輸入 --> 5
輸出 -->  5  10  15  20  25
          4   8  12  16
          3   6   9
          2   4
          1
```

## 練習：迴圈數值相乘

- ❑ 請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數，然後以「倒」三角形的方式依序輸出此數的相乘 ( $i*j$ ) 結果。

- ❑ 提示：輸出欄寬為 4，且需靠右對齊。

```
# 迴圈數值相乘-倒三角形
```

```
n=int(input()) # 輸入正整數
```

```
for i in range(n,0,-1):
```

```
    for j in range(1,i+1):
```

```
        print("%4d" % (i*j), end="")
```

```
    print()
```

## 第四類：進階控制流程

- ❑ 迴圈應用
- ❑ break 與 continue 敘述控制迴圈

# 不定數迴圈

## ❑ 不定數迴圈

- ❑ 無法事先知道迴圈執行的次數。

## ❑ break statement

- ❑ 立刻終止該迴圈的執行, 跳出迴圈

## ❑ continue statement

- ❑ 略過後續的敘述, 直接回到該迴圈起始的條件運算式

```
for i in range(5):  
    if i < 3:  
        break  
    else:  
        print(i)
```

```
for i in range(5):  
    if i < 3:  
        continue  
    else:  
        print(i)
```

# 亂數產生器

- ❑ 寫程式時常常需要用到亂數，隨機產生的數值。
- ❑ 可使用 `random.randint(a, b)`
  - ❑ `randint()` 位於 `random` module 中，故需 `import random`。
  - ❑ 回傳介於 `a, b` ( $a \leq N \leq b$ ) 之間的亂數整數。

```
# 產生(1 ~ 10)亂數
import random
for i in range(1,10):
    print("random number = ",random.randint(1,10))
```



# 猜數字例子

```
import random
secretNumber=random.randint(1,10)
while True:
    print('From 1 to 10. Take a guess...')
    try:
        guessNumber=int(input())
    except (TypeError, ValueError):
        print('Number please!!!')
        continue # 繼續下次輸入
    if guessNumber < secretNumber:
        print('Your guess is too low!')
    elif guessNumber > secretNumber:
        print('Your guess is too high!')
    else:
        print('Good guess.')
        break # 猜中了結束遊戲(迴圈)
```

## 例題：不定數迴圈-求最小值

- 請撰寫一程式，讓使用者輸入數字，輸入的動作直到輸入值為9999才結束，然後找出輸入的最小值，並輸出最小值。

範例輸入

```
29
100
948
377
-28
0
-388
9999
```

範例輸出

```
-388
```

```
# 不定數迴圈-求最小值

n=eval(input())    # 先取得第1個值
min=n              # 該值當最小值
while n != 9999:
    n=eval(input())
    if n < min:     # 若新值比min更小
        min=n      # 更新最小值min
print(min)
```

## 例題：數字反轉判斷

- 請撰寫一程式，讓使用者輸入一個正整數，將此正整數以反轉的順序輸出，並判斷如輸入0，則輸出為0。

範例輸入1

31283

範例輸出1

38213

```
# 數字反轉
num=int(input())           # 輸入正整數
if num == 0:
    print(num)
else:
    while num != 0:
        print(num%10,end="") # 取除以10的餘數
        num //= 10           # 取除以10的商數
```

## 例題：不定數迴圈-分數等級

- ❑ 請撰寫一程式，以不定數迴圈的方式輸入一個正整數(代表分數)，之後根據以下分數與GPA的對照表，印出其所對應的GPA。假設此不定數迴圈輸入-9999則會結束此迴圈。標準如下表所示：

分 數	GPA
-----	-----

90 ~ 100	A
----------	---

80 ~ 89	B
---------	---

70 ~ 79	C
---------	---

60 ~ 69	D
---------	---

0 ~ 59	E
--------	---

```
grade=""
score=eval(input())
while score != -9999:
    if score >= 90:
        grade="A"
    elif 80 <= score <= 89: # 分數區間判斷
        grade="B"
    elif 70 <= score <= 79:
        grade="C"
    elif 60 <= score <= 69:
        grade="D"
    else:
        grade="E"
    print(grade)
    score=eval(input())
```

# 例題：質數判斷

- ❑ 請撰寫一程式，輸入一整數，然後判斷它是否為質數(prime number)。
  - ❑ 提示：質數只能被1和自身整除，試除法：若一自然數  $n$ ，無法被2與 $\sqrt{n}$ 之間的任一整數整除，則  $n$  為質數。

```
n=int(input())           # 輸入整數n
isPrime=True
for i in range(2,int(n**0.5)+1):
    if n % i == 0:        # n被整除
        isPrime=False    # 不是質數
        break            # 終止迴圈
if isPrime:
    print(n,"is a prime number")
else:
    print(n,"is not a prime number")
```

# 練習：繪製等腰三角形

- ❑ 請撰寫一程式，利用雙層for迴圈，依照使用者輸入的  $n$ ，畫出對應的等腰三角形。

範例輸入

7

範例輸出

```
  *  
 ***  
*****  
*****  
*****  
*****  
*****  
*****
```

# 練習：繪製等腰三角形

❑ 請撰寫一程式，利用雙層for迴圈，依照使用者輸入的n，畫出對應的等腰三角形。

範例輸入

7

範例輸出

```
  *
 ***
*****
*****
*****
*****
*****
*****
```

# 繪製等腰三角形

```
n=int(input())
```

# 輸入整數n

```
for i in range(n):
```

# 有n列

```
    for j in range(n-i,1,-1):
```

# 輸出(n-i)個空白

```
        print(" ",end="")
```

```
    for k in range(i*2+1):
```

# 輸出(2i+1)個星號

```
        print("*",end="")
```

```
    print()
```

# 練習：計算最大公因數

- ❑ 請撰寫一程式，輸入三個整數，然後求出最大公因數。
  - ❑ 提示：窮舉法：逐一系列出可整除各整數的所有因數，找出最大的公因數。



# 練習：計算最大公因數

- ❑ 請撰寫一程式，輸入三個整數，然後求出最大公因數。
  - ❑ 提示：窮舉法：逐一系列出可整除各整數的所有因數，找出最大的公因數。

```
# 計算最大公因數
a=int(input())
b=int(input())
c=int(input())
gcd=1          # 預設公因數為1
k=2            # 最小可能因數
while k<=a and k<=b and k<=c:
    if a%k==0 and b%k==0 and c%k==0:
        gcd=k
    k+=1        # 下一個可能因數
print("gcd(%d, %d, %d) = %d" % (a,b,c,gcd))
```