# Happy Python Day

https://bit.ly/scist python

高英耀 joseph@gm.tnfsh.tn.edu.tw



# 本課程由以下贊助商贊助辦理

DEVCORE





CATCHER smart process 可成科技





# 資安倫理宣傳

本課程目的在提升學員對資訊安全之認 識及資安實務能力,深刻體認到資安的 重要性!所有課程學習內容不得從事非 法攻擊或違法行為,所有非法行為將受 法律規範,提醒學員不要以身試險。

# 課程相關資訊

- Python語法: Python 3.X
- 撰寫環境:Linux + gedit + python3編譯
- 解題平台:
  - 程式解題平台:skyoj (http://skyoj.tnfsh.tn.edu.tw/sky)
  - CTF: SecurityFocusOnline CTF (http://140.110.112.212)

### 開啟Kali Linux

### 1.開啟 VirtualBox





### 2.點選Kali Linux, 按執行鈕

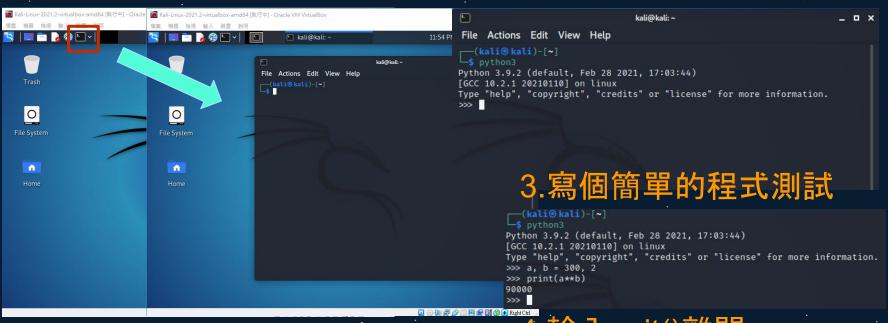




### 在 Kali 上撰寫 Python3 直譯程式

### 1.點選Terminal Emulator

### 2.輸入python3



4.輸入exit()離開

### 撰寫 Python3 程式檔案



-(kali⊕kali)-[~]

# Linux上撰寫Python 練習

#### 練習:

- 1. 開啟 Kali Linux, 並登入
- 2. 開啟 Shell 介面 (Terminal Emulator)
- 3. 使用 python3 直譯模式
- 4. 開啟 gedit 編修一個 python 程式碼,編修完成後執行該程式

### **Happy Python Day**

A.變數與資料型態 運算式與運算子

B.輸入與輸出

C.條件判斷與迴圈

<u>D.內建函數與物件方</u> <u>法</u>

**E.字**串

<u>F.列表</u>

G.函數、模組、套件

H.編解碼應用

I. PPC解題

















### A-1.變數命名規則

程式中的變數主要用於儲存使用者輸入資料,並隨程式執行,其儲存內容亦會隨之改變,故稱之為變數。

- 變數命名規則
  - 由英文、數字、底線組成
  - <u>不</u>得以<u>數字</u>開頭
  - 不能與Python內建的保留字相同

### A-2.變數指派

● Python變數<mark>不需宣告</mark>,依指派值自動設定資料型態。

● 指派語法:

變數名稱 = 指派值

• 範例:

```
a = 5 # a為整數, 且值為5
```

### A-2.變數指派

- 多個變數指派相同值 a = b = c = 10
- 同一列指派多個變數 name, number = 'Joseph', 35

### A-3.資料型態

- 數值型態(Numeric Data Types):int, float, bool
- 字串型態(String Data Types): str, chr.
- 容器型態(Container Types): list, dict, tuple, set

### A-4.資料型態轉換

• Python 自動轉換

```
score = 60score = score + 3.5
```

#自動轉換為浮點數, 結果為63.5

#### • 函數強制轉換:

- o int().
- o float()
- o str()

- # 將括弧內的資料轉換為整數
- # 將括弧內的資料轉換為浮點數
- # 將括弧內的資料轉換為字串

# A-5.運算式

- 運算式規則:變數 = 運算式
- 範例:sum = ((a+b)\*3+c\*4) / 10
- 說明:將等號右邊結果存於變數 sum

### A-6.運算子

- 運算子俗稱運算符號, 可區分為
  - 指派運算子:=
  - 算術運算子:+, -, \*, /, %, //, \*\*
  - 關係運算子:>, <, >=, <=, !=
  - 邏輯運算子: and, or, not:
  - 成員運算子:in
  - 位元運算子: &(且), |(或), ^(互斥或), ~(反相)
  - 位移運算子:<<(向左位移),>>(向右位移).

# A-6-1.算術運算子

算術運算子	意義	範例	運算結果
+	加	5+3	8
-	減	5-3	2
*	乘	5*3	15
1	除	5/3	1.666666666666667
%	餘數運算	5%3	2
//	商數運算	5//3	1
**	指數運算	5**3	125

# A-6-2.關係運算子

算術運算子	意義	範例	運算結果
>	大於	5>2	True
<	小於	5<2	False
>=	大於等於	5>=2	True
<=	小於等於	5<=2	False
==	等於(比較)	5==2	False
!=	不等於	5!=2	True
>	大於	5>2	True

# A-6-3. 邏輯運算子

算術運算子	意義	範例	運算結果
		(5>3)and(3>2)	True
and	TO.	(5>3)and(3<2)	False
	和	(5<3)and(3>2)	False
		(5<3)and(3<2)	False
or		(5>3)or(3>2)	True
		(5>3)or(3<2)	True
	或	(5<3)or(3>2) True	True
		(5<3)or(3<2)	False
not	反相	not(5>3)	False
		not(5<3)	True

# A-6-4.複合指定運算子

複合指定運算子	意義	原運算式	縮寫後運算式
+=	加法	A=A+3	A += 3
-=	減法	A=A-3	A -= 3
*=	乘法	A=A*3	A *= 3
/=	除法	A=A/3	A /= 3
%=	求餘數	A=A%3	A %= 3
//=	求商數	A=A//3	A //= 3
**=	指數	A=A**3	A **= 3

# A-6-5. 成員運算子

in 用來判斷list, string等資料型態是否包含指定元素、dict是否包含指定的key, 亦可與for連用。

範例	輸出	範例	輸出
<pre>a = [1, 2, 3] print(1 in a) print("1" in a)</pre>	True False	<pre>d = {1:2, 3:4} print(1 in d) print(2 in d)</pre>	True False
b = "123" print("23" in b)	True	<pre>a = [1, 2, 3] print(1 not in a) print("1" not in a)</pre>	False True

### 運算子練習

SecurityFocusOnline CTF

[Python101] 11的51次方



















### B-1.輸出函數

print 語法

print([項目1, 項目2, ..., sep = 分隔字元, end = 結束字元])

- 若輸出變數內容, 直接放變數名稱
- 若要輸出字串, 前後加上'或"
- sep (分隔字元) 預設為空白字元
- end (結束字元) 預設為換行字元\n

# B-2.輸出函數範例

範例	輸出	範例	輸出
a, b = 5, 10 print(a, b)	5 10	<pre>a, b = 5, 10 print(a, end='') print(b)</pre>	510
a, b = 5, 10 print(a, b, sep=',')	5,10		

### B-3.輸出格式化字串

Python 有 3 種不同的方式來達成字串格式化 (String format).

若輸出較複雜的字串,可採用以下方式:

- %-formatting
- str.format (Python 2.6+)
- f-string (Python 3.6+)

# B-3-1.%-formatting

- 最早的Python 格式化字串延用C語言的字串格式化·
- 透過%運算符號,將在元組(tuple)中的各個元素依照指定的格式化 方式輸出。
  - %s(字串)、%d(十進位整數)、%f(浮點數)

```
>>> "I am %s %s. %s" % ("Monkey", "D", "Luffy")
'I am Monkey D. Luffy'
```

### B-3-2.str.format()

- Python2.6 新增格式化字串函數str.format(')
- 解決 %-formatting 不易閱讀的困擾。
  - 透過 { } 和 format 來代替 % 符號

```
>>> s = 'I am {first_name} {middle_name}. {last_name}'
>>> s.format(first_name='Monkey', middle_name='D', last_name='Luffy')
'I am Monkey D. Luffy'
```

# B-3-3.f-string

- Python3.6 新增f-string格式化字串功能, 只需於字串前加上前綴字f
- 解決 %-formatting 不易閱讀的困擾。
- 解決str.format() 接變數後程式碼超長的問題

```
first_name = "Monkey"
middle_name = "D"
last_name = "Luffy"
print(f"I am {first_name} {middle_name}. {last_name}")

# Output:
I am Monkey D. Luffy
```

### B-4.輸入函數

• input: 以字串型態讀入整行資料, 直至換行為止。語法如下:

變數 = input([提示字串])

範例	輸出
a = input('輸入國文成績:') b = input('輸入數學成績:') c = input() print('三科成績為%5s %5s %5s' %(a,b,c))	輸入國文成績:75 輸入數學成績:55 80 三科成績為75 55 80

# B-5.輸入強制轉資料型態

• input()搭配int()將輸入的字串型態轉成整數型態

變數 = int( input() )

範例	輸出
a = int(input('輸入國文成績:')) b = int(input('輸入數學成績:')) c = int(input()) print(f'三科總分為{a+b+c}')	輸入國文成績:75 輸入數學成績:55 80 三科總分為210

### B-6.程式解題平台常見輸入格

#### 輸入說明

• 共一行,為正整數n,代表前一個數, $1 < n < 10^8$ 

#### 輸出說明

- 請輸出小明數列的下一個數。
- 輸出完請換行。

#### 輸入範例

#### 輸出範例

# case 1: 一行只有一個數字 **二**ase 2: 一行有多個數字

#### 輸入說明

• 共一行,有二個整數a,b,分別代表兩個國家在戰場上的士兵數, $1 < a,b < 2^{63}$ 

#### 輸出說明

- 輸出最後倖存的士兵數若無倖存士兵,則輸出"all dead"。
- 輸出完請換行。

#### 輸入範例1

#### 輸出範例1

### B-6-1.輸入格式:每行只有一個數字

- 因input()讀入的資料型態為字串,無法算術運算,需搭配int()
- 語法: a = int(input())
- 範例:輸入有兩行,每行一個數字,請輸出這兩個數字和。
- 輸入範例:60 a b

40

a, b = int(input()), int(input())
print(a+b)

● 輸出範例:100

### B-6-2.輸入格式:每行有多個數字

- 語法: a, b = map(int, input().split())
- 範例:輸入只有1行,有3個數字a,b,c,請輸出(a\*b)%c。
- 輸入範例: 11 23 19
- 輸出範例:6

```
a, b, c = map(int, input().split())
print((a*b) % c)
```

### 輸入與輸出練習

```
**P1-1. 圓面積計算**
```

#### [題敍]

給定半徑 r, 計算圓面積。(圓周率 pi = 3.14)

#### [輸入說明]

只有一個整數 r。

#### [輸出說明]

輸出圓面積。

#### [輸入範例]

5

#### [輸出範例]

78.5

\*\*P1-2. 矩形面積計算\*\*

#### [題敍]

給定長與寬. 計算矩形面積。

#### [輸入說明]

只有一行包含 2 個整數 a, b, 以空白間隔。

#### [輸出說明]

輸出矩形面積。

#### [輸入範例]

101 201

#### [輸出範例]

20301

















## 條件判斷

- Python 程式語言是一行一行執行的,要讓程式的執行能 夠因情況而轉彎,那就需要用到條件判斷式來控制流程。
- 在 Python 語言中,就提供了 if 、 else 、 elif 這三種語法來協助我們實現各種條件判斷和流程控制。

#### C-1-1.if敘述

if 語法

if 條件式:

程式區塊

- 條件式可不用括號(),條件式後需搭配冒號:
- 程式區塊以縮排方式處理,同一層縮排視為同一程式區塊



```
# 奇偶數判斷
num = int(input())
if num%2 == 0:
    print('%d is even' %(num))
if num%2 != 0:
    print('%d is odd' %(num))
```

[→ 60 60 is even

## C-1-2.if-else敘述

if-else 語法

if 條件式:

程式區塊1

else:

程式區塊2

```
# 奇偶數判斷
num = int(input())
if num\%2 == 0:
  print('%d is even' %(num))
else:
 print('%d is odd' %(num))
```

77 is odd

### C-1-3.if-elif-else敘述

if-elif-else 語法

```
# 成績等第判斷
score = int(input())
if score \geq 90:
 print('A')
elif score >= 80:
 print('B')
elif score >= 70:
 print('C')
elif score >= 60:
 print('D')
else:
 print('F')
```



## 條件判斷練習

SecurityFocusOnline CTF

[Python101]

IF 潤年

IF幸福企業的獎金制度

skyoj 程式練習

P1 奇偶命運大不同

P2 一中半島爭奪戰

## 迴圈

- 在處理資料的時候,時常會需要重複執行某些相同的步驟, 如計算全校每位同學的段考平均。
- 迴圈 (loop) 的作用是讓指定的某段敘述在條件符合的情況下一直重覆執行。
- 在程式設計中,可以利用迴圈來進行重覆性的資料輸入、處 理與輸出。

## C-2-1.while敘述

#### while 語法

while 條件式:

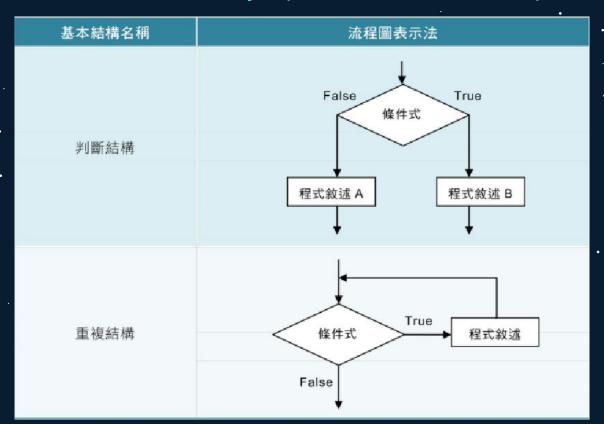
程式區塊

- m, n = int(input()), int(input())
  while(n>0):
   m, n = n, m%n
  print(m)
- **□**→ 42 75 3

```
n = int(input())
while(n > 0):
print(n%10)
n = n//10
```



## C-2-2.if與while的差異



## C-3-1.for敘述

for 語法

for 變數 in 序列:

欲執行的程式區塊

- for迴圈的變數會依序走訪序列中的元素
- 序列可為range函式、字串(string)、表列(list)、元組(tuple)、字典(dict)、集合(set)

## C-3-2.for+range()

for 語法

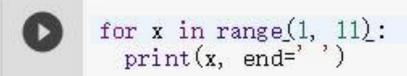
for 變數 in range([起始值,] 終止值[, 遞增值]):

欲執行的程式區塊

- 起始值預設為0, 遞增值預設為1
- 起始值 ≤ range()的範圍 < 終止值 (左閉右開)</li>

## C-3-3.for+range()範例

- for x in range(6):
  print(x, end='')
  - C→ 012345



1 2 3 4 5 6 7 8 9 10

- for x in range(3, 10, 2):
  print(x, end='')
- C→ 3579

## C-3-4.序列為字串(string)範例

```
s = 'From A to A+'
for x in s:
  print(x)
```

```
s = input()
for x in s:
   print(chr(ord(x)+3), end='')

TNFSH
WQIVK
```

```
註:ord() ⇒ 將字元轉為ASCII編碼(整數)
chr() ⇒ 將ASCII編碼(整數)轉為字元
```

#### C-4.break敘述

·break語法:跳出當前迴圈

```
#break範例
pre = post = 0
while (True):
  m, n = int(input()), int(input())
  if (m == n): break
  elif (m > n): pre += 1
  else: post += 1
print('前者大:%d、後者大:%d' %(pre, post))
30
10
40
10
20
20
前者大:2、後者大:0
```

## C-5.continue敘述

·continue語法:略過下方語法,直接執行下一輪迴圈·

```
#印出區間所有7的倍數
m, n = int(input()), int(input())
if (m > n):
  m, n = n, m
for i in range (m, n+1):
  if (i%7 != 0):
    continue
  print(i, end=' ')
50
150
             84 91 98 105 112 119 126 133 140 147
```

## C-6.巢狀迴圈範例

#印出九九乘法表

```
for i in range (1, 10):
      for j in range (1, 10):
        print('%d*%d=%2d' %(i, j, i*j), end=' ')
                #換行
      print()
\Gamma
             1*2= 2
                      1*3=
                               1*4= 4
                                        1*5= 5
                                                 1*6= 6
                                                          1*7= 7
                                                                    1*8= 8
                                                                             1*9= 9
    1*1=
                                                          2*7=14
    2*1=
             2*2= 4
                      2*3=
                               2*4= 8
                                        2*5=10
                                                 2*6=12
                                                                    2*8=16
                                                                             2*9=18
         2
                           6
                                                                    3*8=24
    3*1=
             3*2= 6
                      3*3= 9
                               3*4=12
                                         3*5=15
                                                  3*6=18
                                                           3*7=21
                                                                             3*9=27
         3
    4*1= 4
             4*2= 8
                      4*3=12
                               4*4=16
                                        4*5=20
                                                 4*6=24
                                                          4*7=28
                                                                    4*8=32
                                                                             4*9=36
    5*1= 5
             5*2=10
                      5*3=15
                               5*4=20
                                        5*5=25
                                                 5*6=30
                                                           5*7=35
                                                                    5*8=40
                                                                             5*9=45
                                        6*5=30
                                                                             6*9=54
    6*1= 6
             6*2=12
                      6*3=18
                               6*4=24
                                                 6*6=36
                                                           6*7=42
                                                                    6*8=48
                      7*3=21
                                                                    7*8=56
    7*1=
             7*2=14
                                         7*5=35
                                                           7*7=49
                                                                             7*9=63
         7
                               7*4=28
                                                  7*6=42
    8*1= 8
             8*2=16
                      8*3=24
                                         8*5=40
                                                 8*6=48
                                                           8*7=56
                                                                    8*8=64
                                                                             8*9=72
                               8*4=32
                                                           9*7=63
                                                                             9*9=81
    9*1= 9
             9*2=18
                      9*3=27
                               9*4=36
                                        9*5=45
                                                 9*6=54
                                                                    9*8=72
```

## 迴圈練習

SecurityFocusOnline CTF

[Python101]

FOR your summation

skyoj 程式練習

P3 盗墓筆記

P4 Collatz conjecture









# 內建函數與物件方法









## D-1.內建函數與物件方法差別

- 內建函數:print()、input().
  - 一段「被命名」的程式碼,這段程式碼可以用於執行某個特定任務
- 物件:str、list
  - Python為物件導向程式語言,物件包含資料型態、類別、模組
- 物件方法:str.split()、list.append()
  - 附屬於物件之下的函數, 需以物件變數後接, 再接函數名稱

## D-2.常用內建函數(Built-in Functions)

abs()	chr()	eval()	float()
input()	int()	len()	list()
map()	max()	min()	ord()
pow()	print()	range()	sorted()
str()	sum()	type()	

## D-2-1.abs()

- 功用:回傳絕對值
- 語法:abs(x)
- 參數說明:x 為整數
- 範例:

```
>>> abs(-5)
5
>>> abs(66)
66
```

## D-2-2.chr()

- 功用: ASCII值轉字元
- 語法:chr(i)
- 參數說明: i 為ASCII值
- 備註:與chr()函數相反者為ord()

•	範例:	>>>	chr(97)		
		>>>	chr(65)		

二進位	十進位	十六進位	圖形	二進位	十進位	十六進位	冒形
0100 0000	64	40	0	0110 0000	96	60	*
0100 0001	65	41	A	0110 0001	97	61	a
0100 0010	66	42	В	0110 0010	98	62	ь
0100 0011	67	43	С	0110 0011	99	63	с
0100 0100	68	44	D	0110 0100	100	64	d
0100 0101	69	45	Е	0110 0101	101	65	е
							_

## D-2-3.eval()

• 功用:執行字串運算式,並回傳結果

● 語法:eval(expression)

● 參數說明: expression 為字串運算式

範例: >>> x = 7 >>> eval('3\*x') 21 >>> eval('pow(x, 2)')

## D-2-4.float()

• 功用:將整數或字串轉成浮點數資料型態

● 語法:float(x)

• 參數說明: x 為整數或字串

● 範例: >>> float(100) 100.0 >>> float('-123.45\n') -123.45 >>> float('1e-3') 0.001

## D-2-5.input()

- 功用:以字串格式讀入整行資料
- 語法:input([prompt])
- 參數說明: prompt 為輸入提示字, 可省略
- 範例: >>> s = input('Please input your name ---> ') Please input your name ---> Elon Mask >>> s 'Elon Mask'

## D-2-6.int()

- 功用:將不同進制表達字串轉成10進制整數
- 語法:int(x, base=10)
- 參數說明:x 為不同進制字串, base 為進制(預設為10進制)

#### D-2-7.len()

- 功用:回傳物件的長度(元素個數)
- 語法:len(s)
- 參數說明: s 為序列, 如: string, list, range, dict
- 範例: >>> len('Taiwan can help')
  15
  >>> len([1, 2, 3, 4, 5, 6])

### D-2-8.list()

- 功用:將字串、元組等可迭代序列轉換成列表(list)
- 語法:list([iterable])
- 參數說明: iterable 為可迭代序列, 如: string, range, tuple
- 範例:
   >>> print(a)
   []
   >>> print(list('Taiwan can help'))
   ['T', 'a', 'i', 'w', 'a', 'n', ' ', 'c', 'a', 'n', ' ', 'h', 'e', 'l', 'p']
   >>> print(list(range(7)))
   [0, 1, 2, 3, 4, 5, 6]

## D-2-9.map()

- 功用:將可迭代序列丟入函數運算,回傳一個迭代器
- 語法: map(function, iterable, ...)
- 參數說明: function 為函數名, iterable 為可迭代序列
- 範例: >>> print(map(int, ['20', '30', '40']))
   <map object at 0x0000018ABB35E590>
   a, b, c = map(int, ['20', '30', '40'])
   >>> print(a, b, c)
   20 30 40

#### D-2-10.max()

- 功用:回傳最大值
- 語法: max(arg1, arg2, ...)、max([iterable])
- 參數說明: iterable 為可迭代序列
- 範例: >>> max(50, -30, 80, -100) 80
   >>> max("Taiwan can help") 'w'
   >>> max([10, 20, 30, 40, 50]) 50
   >>> max([[1,0], [1,1], [2,1], [2, 0]]) [2, 1]

## D-2-11.min()

- 功用:回傳最小值
- 語法: min(arg1, arg2, ...)、min([iterable])
- 參數說明: iterable 為可迭代序列

## D-2-12.ord()

● 功用:字元轉ASCII值

● 語法:ord(c)

● 參數說明: ⋵為字元

● 備註:與ord()函數相反者為chr()

• 範例: >>> ord('A') 65 >>> ord('z')

二進位	十進位	十六進位	冒形	二進位	十進位	十六進位	冒形
0100 0000	64	40	0	0110 0000	96	60	*
0100 0001	65	41	A	0110 0001	97	61	a
0100 0010	66	42	В	0110 0010	98	62	ь
0100 0011	67	43	С	0110 0011	99	63	с
0100 0100	68	44	D	0110 0100	100	64	d
0100 0101	69	45	Е	0110 0101	101	65	е

## D-2-13.pow()

• 功用:回傳指數值

• 語法:  $pow(base, exp[, mod]) \Rightarrow base^{exp} \% mod$ 

● 參數說明: base 為底數、exp 為指數、mod 為模數

• 範例: >>> pow(10, 5) 1000000 >>> pow(7, 5, 11) 10

## D-2-14.print()

• 功用:輸出

● 語法: print(objects, sep='', end='\n')

- 參數說明: objects 為python物件、sep 為分隔字元(預設空白)、end 為結束字元(預設換行)
- 範例: >>> a, b, c = 50, 'Taiwan', [1, 2, 'AA'] >>> print(a, b, c) 50 Taiwan [1, 2, 'AA']

## D-2-15.range()

- 功用:回傳一個計數範圍的迭代類別
- 語法:range(stop)、range(start, stop[, step])
- 參數說明: start 為計數起始值(預設為0)、stop 為計數終止值(不包含stop)、step 為遞增值(預設為1)
- 範例: >>> list(range(5)) [0, 1, 2, 3, 4] >>> list(range(1, 10, 2)) [1, 3, 5, 7, 9]

## D-2-16.sorted()

- 功用:回傳一個排序好的列表
- 語法:sorted(iterable, key=None, reverse=False)
- 參數說明: iterable 為可迭代序列、reverse 為排序規則(False為遞增、True為遞減)
- 範例: >>> a = [5, 2, 3, 1, 4]
   >>> b = sorted(a, reverse=True)
   >>> print(b)
   [5, 4, 3, 2, 1]

## D-2-17.str()

- 功用:將資料轉成字串
- 語法:str(object)
- 參數說明: objects 為python物件
- 範例: >>> a, b = 13579, [1, 3, 5, 7, 9] >>> print(str(a), str(b)) 13579 [1, 3, 5, 7, 9]

### D-2-18.sum()

- 功用:對可迭代的序列(必須為整數資料)求總和.
- 語法:sum(iterable[, start=0])
- 參數說明: iterable 為可迭代序列、start 為總和起始值(預設0)
- 範例: >>> sum([1, 2, 3, 4, 5])
  15
  >>> sum([1, 2, 3, 4, 5], 100)
  115

# D-2-19.type()

• 功用:回傳物件類型

● 語法:type(object)

● 參數說明: objects 為python物件

# D-3.物件方法

留待字串、列表介紹

## 內建函數與物件方法練習

SecurityFocusOnline CTF

[Python101]

16進位制的計算

8進位的閱讀

skyoj 程式練習

P5 [字串] 車牌等級

















### E-1.字串 str 簡介

- 資料格式:以''或""將字元資料含括起來
- 字串中的資料(稱之為元素)是有序排列,從0開始編號
- 字串中的元素不允許變更
- 範例:s = 'abcdefgh'、ans = ''(ans為空字串)

### E-2.字串的元素編號(索引值index)

範例:s = 'abcdefghi'

index 左→右	0	1	2	3	4	5	6	7	8
字串 s	a	Ъ	С	d	e	f	g	.h	· ji k
index 右→左	-9	-8	-7	-6	-5	-4	-3	-2	-1

#### E-3.字串運算子:+, \*

- +:str 串接
- \*:str 重複
- 範例:

```
>>> a = 'Happy'
>>> b = a + ' ' + 'Python'
>>> print(b)
Happy Python
```

```
>>> c = 'ABC' * 3
>>> print(c)
ABCABCABC
```

#### E-4.字串運算子: []

- []:截取 str 部份字元
- 用法1 [index]: 取用 index 所在字元
- 用法2 [ start: stop: step]: 截取索引值為start到stop-1的字元, 並

依step值跳躍取用(step值-1代表反轉取用)

範例: >>> a = 'I love Python' print(a[3]) o print(a[2:6]) love

#### E-5.字串運算子: in

- in:成員運算子,回傳 True/False
- 格式: <something> in <sequence>
- 範例:s = 'abcdefghi' 'd' in s ⇒ True
  - 'z' in s ⇒ False

### E-6.字串常用內建函數

• a = 'TNFSH', b = 12345

通用函數	描述	範例	x 值
len( <str>)</str>	計算元素個數	x = len(a)	5
list( <seq>)</seq>	將 <seq>轉成列表(list)</seq>	x = list(a)	['T', 'N', 'F', 'S', 'H']
ord( <chr>)</chr>	將列表中所有元素加總	x = ord(a[1])	78
str( <obj>)</obj>	將 obj 轉為字串	x = str(b)	'12345'
sorted( <str>)</str>	回傳依遞增排序的列表	x = sorted(a)	['F', 'H', 'N', 'S', 'T']

### E-7.字串專用物件方法

• a = 'TNFSH', b = ' 10 20 30 '

	通用函數	描述	範例	x 值	
$\cdot \Big $	<str>.lower() <str>.upper()</str></str>	回傳轉小寫/大寫的字串	x = a.lower()	'tnfsh'	
(30)	<str>.replace(old, new)</str>	將 <str>中的old子字串以new子字 串取代,回傳取代完新字串。</str>	x = a.replace('FSH', . 'GS')	'TNGS'	
•	<str>.split([sep])</str>	字串以 sep 分割, 回傳列表。 sep 預設空白。	x = b.split()	['10', '20', '30']	
	移除字串頭尾的 chars 子字串。 chars 預設空白。		x = b.strip()	'10 20 30'	

# E-7-1.字串專用物件方法 replace()

- 功用: 把字串中的 old 替換成 new, 回傳替換後新字串。
- 語法: str.replace(old, new[, max])
- 參數說明: max 為替換次數(省略則替換次數不限制)
- 範例: >>> x = 'TNFSH\_IN\_TAINAN-TAINAN\_400TH\_ANNIVERSARY' >>> y = x.lower().replace('\_', ' ').replace('-', ', ') y 'tnfsh in tainan, tainan 400th anniversary'

## E-7-2.字串專用物件方法 split()

- 功用:對字串分割,回傳分割後的列表list
- 語法:str.split(sep='')
- 參數說明: sep 為分割字元(可省略, 預設為空白)
- 範例: >>> a, b = '10 20 30 40', 'AA,BB,CC,DD' >>> print(a.split()) ['10', '20', '30', '40'] >>> print(b.split(',')) ['AA', 'BB', 'CC', 'DD']

#### 字串練習

SecurityFocusOnline CTF

[Python101]

字串的逆向工程

skyoj 程式練習

P6 神文

P7 雙進制迴文

P8 [實作1] 位數總和













S.





#### F-1.列表(list)

- 資料格式:以[]將不同型態的資料含括起來,以,分隔
- 列表中的資料(稱之為元素)是有序排列,從0開始編號
- 格式:表列名稱 = [元素0, 元素1, ...]
- 範例:

取用列表元素	替換列表中元素	宣告空的列表
data = [ 'John', [95, 118], 'May', 100 ] print(data[1]) ⇒ [95, 118]	data[1] = 20 print(data) ⇒ ['John', 20, 'May', 100]	a = []

#### F-2.列表list運算子:+, \*

- +:list 串接
- \*:list 重複
- 範例:

```
>>> a, b = [1, 2, 3], [10, 20, 30]
>>> a += b
>>> print(a)
[1, 2, 3, 10, 20, 30]
```

```
>>> c = ['A', 'B', 'C']
>>> c *= 2
>>> print(c)
['A', 'B', 'C', 'A', 'B', 'C']
```

#### F-3.列表list運算子: []

- []:截取 list 部份元素
- 用法1 [index]:取用 index 所在元素
- 用法2 [ start : stop : step ]: 截取索引值為start到stop-1的元素,並
   依step值跳躍取用(step值-1代表反轉取用)
- 範例: >>> a = [0, 1, 2, 3, 4, 5]
  >>> print(a[0])
  0
  >>> print(a[1:3])

```
>>> print(a[:4])
[0, 1, 2, 3]
>>> print(a[2:])
[2, 3, 4, 5]
>>> print(a[:4:2])
[0, 2]
```

## F-4.列表list運算子: in

- in:成員運算子,回傳 True/ False
- 格式:<something> in <sequence>
- 範例:x = [10, 20, 30] 10 in x ⇒ True
  - 50 in  $x \Rightarrow False$

# F-5.列表(list)常用內建函數

a = [ 'John', [95, 118], 'May', 100 ]

b = [89, 119, 55]

通用函數	描述	範例 結果	
len( <list>)</list>	計算元素個數	len(a)	4
list( <seq>)</seq>	將 <seq>轉成列表(list)</seq>	list('TNFSH')	['T','N','F','S','H']
sum( <list>)</list>	· 將列表中所有元素加總 註:僅限列表元素皆為數字	sum(b)	263
max( <list>) min(<list>)</list></list>	找出列表中最大(小)值 若為字串,則依字典序排列	max(b) min(b)	119 55
sorted( <list>)</list>	回傳依遞增排序的列表	c = sorted(b) print(c)	[ 55, 89, 119 ]

## F-6.列表(list)專用物件方法

a = [ 'John', [95, 118], 'May', 100 ]

	通用函數	描述	範例	結果	
	<li>st&gt;.clear( )</li>	清除列表中所有元素	a.clear()	П	
$\cdot [$	<li>st&gt;.append(<obj>)</obj></li>	將 <obj>加到<list>尾端</list></obj>	b.append(35)	[ 89, 119, 55, 35 ]	
	<li>sort()</li>	將列表中元素遞增排序 (直接於原列表操作)	b.sort() print(b)	[ 55, 89, 119 ] ·· ·	
	<li>st&gt;.reverse()</li>	·.reverse() 反轉列表元素		[ 55, 119, 89 ]	
	<li><li>!st&gt;.extend(<list1>)</list1></li></li>	st>.extend( <list1>) 將<list1>合併至<list>尾端</list></list1></list1>		[ 'John', [95, 118], 'May', 100, 89, 119, 55 ]	

### F-6-1.列表(list)常用物件方法 clear()

● 功用:將列表(list)清空

語法: list.clear()

• 參數說明:

• 範例: >>> a = [0, 1, 2, 3, 4, 5] >>> a.clear() >>> print(a)

## F-6-2.列表(list)常用物件方法 append()

• 功用:將物件加至列表(list)尾端

語法: list.append(x)

● 參數說明:x 為欲加至列表尾端的物件

範例: >>> a = [0, 1, 2, 3, 4, 5]>>> a.append(100)>>> print(a)[0, 1, 2, 3, 4, 5, 100]

#### F-6-3.列表(list)常用物件方法 sort()

- 功用:將列表(list)元素排序
- 語法: list.sort(key=None, reverse=False)
- 參數說明: reverse 為排序規則(False為遞增、True為遞減)
- >>> a = [7, 2, 5, 3, 8] >>> a = [7, 2, 5, 3, 8] 範例: >>> a.sort() >>> print(a) [2, 3, 5, 7, 8]
  - >>> a.sort(reverse=True) [8, 7, 5, 3, 2]

### F-6-4.列表(list)常用物件方法 reverse()

● 功用:反轉列表(list)元素

語法:list.reverse()

• 參數說明:

• 範例: >>> a = [0, 1, 2, 3, 4, 5] >>> a.reverse() >>> print(a) [5, 4, 3, 2, 1, 0]

## F-6-5.列表(list)常用物件方法 extend()

- 功用:在列表(list)末尾添加另一個序列
- 語法: list.extend(seq)
- 參數說明: seq 為添加的序列, 可以為list、tuple、set
- 範例: >>> a, b = [0, 1, 2, 3, 4, 5], ['AAA', 'BBB']
   >>> a.extend(b)
   >>> print(a)
   [0, 1, 2, 3, 4, 5, 'AAA', 'BBB']

## F-6-6.列表(list)常用物件方法 copy()

- 功用:複製列表(list) \*注意與=的差異
- 語法: list.copy()
- 節例:

  >>> a = [0, 1, 2, 3, 4, 5]
  >>> b = a
  >>> a[0] = -1
  >>> print(a, b)
  [-1, 1, 2, 3, 4, 5] [-1, 1, 2, 3, 4, 5]
  >>> a = [0, 1, 2, 3, 4, 5]
  >>> c = a.copy()
  >>> a[0] = -1

[-1, 1, 2, 3, 4, 5] [0, 1, 2, 3, 4, 5]

>>> print(a, c)

b = a, 則a, b兩變數記憶體位置相同 a 做任何改變, b 也會變

c = a.copy(), 則a, c兩變數記憶體位置不同 a 做改變, c 不影響

# F-7.列表建構 list comprehension

- 建構 list 的方法, 語法分成三部分
- 語法 [ expression for item in list (if condition) ]
- 說明 expression可為函數或運算式、condition為條件式(可省略)
- 範例:

```
>>> numbers = []
>>> for x in range(10):
... numbers.append(x * 3)
...
...
>>> print(numbers)
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27]
```

```
>>> numbers = [x*3 for x in range(10)]
>>> print(numbers)
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27]
```

#### F-7-1.列表建構範例 I

範例:給定已知以空白間隔的數字資料

```
#將數字資料以空白分割成列表,再遍歷列表,將每個元素轉整數新增至空列表

a = '34 55 89 66 91 69 38 77 26 14 54'

b = a.split()

c = []

for x in b:

    c.append(int(x))

print(c)
```

```
#簡化為列表建構
a = '34 55 89 66 91 69 38 77 26 14 54'
c = [int(x) for x in a.split()]
print(c)
```

#### F-7-2.列表建構範例 II

範例:使用者輸入一串數字(以空白分隔)

```
#將輸入資料以空白分割成列表,再遍歷列表,將每個元素轉整數新增至空列表

a = input()

b = a.split()

c = []

for x in b:

    c.append(int(x))

print(c)
```

```
#簡化為列表建構
c = [int(x) for x in input().split()]
print(c)
```

# F-8.列表list與字串str比較

比較項目	列表list	字串str	
範例	a = ['AB', 55, [0,1], 7]	b = 'I love Python'	
含括各元素的符號	[ ]	''或""	
元素的資料型態	不限資料型態	字元	
各元素是否有序排列	是(索引值從0開始)	是(索引值從0開始)	
修改元素內容	可 (a[2] = 99)	不可 (b[②	

#### 列表練習

SecurityFocusOnline CTF

[Python101]

黄金比例與費氏數列

skyoj 程式練習

P9 [B3]七星彩

P10 [APCS-1]修補圍籬

P11 Shopaholic













函數、模組、套件







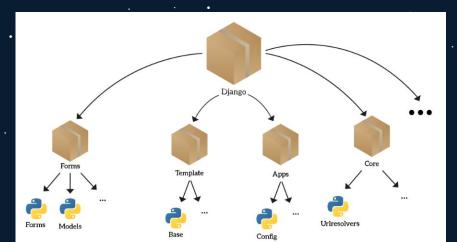


#### G-1.函數、模組、套件的差別

函數(function):一段程式碼

模組(module):一個 python 檔案, 包含 class、variable、function

套件(package):一個目錄, 包含 subpackage、module:



### G-2.函數(function)

- 函數為一段程式碼程式碼的集合,可視為一個獨立的程式區段
- 函數可重覆呼叫使用,為結構化程式語言的重要元素
- python 函數可區分為三大類:
  - .○ 內建函數::print()、input()、int()、len()、str()...
  - 第三方公司所開發的模組module
  - 自定義函數

### G-3.模組(module)

- 模組為一個python程式檔案,包含了變數、類別、函數等
- 範例: random模組
- 匯入模組語法:import 模組名稱
  - ·○ import random # 匯入random模組
- 使用模組內函數語法:模組名稱.函數名稱
  - random.randint(1, 100) # 產生一個1-100間的隨機數

## G-4.套件(package)

- 套件為一個python程式資料夾,包含了多個python檔案(模組)
- 範例: <u>pwnlib套件</u>
- . 匯入套件語法: import 套件名稱.模組名稱
  - import pwnlib.tubes.ssh # 匯入ssh模組

### G-5.import 語法

1. import [套件.]模組
import random
random.randint(1, 100)

2. from 模組/套件 import 函數/模組 · from random import randint randint(1, 100)

3. import 模組 as 別名
import random as rd
rd.randint(1, 100)

4. from 模組/套件 import \* (較不建議)
from random import \* ·
randint(1, 100)

## G-6. 匯入第三方公司開發模組 (套件)範

範例:產生 5 個 1~100 之間的隨機數, 存放於列表中

```
# python code
import random
a = [random.randint(1, 100) for _ in range(5)]
print(a)

# result
[37, 85, 73, 2, 13]
```

#### G-7. 自定義函數

宣告

def 函數名稱([參數1, 參數2, ...]): #參數可省略, 亦可指定預設值

語法

執行程式區塊 · [return 回傳值]

#若無回傳值,無需return。

#若多個回傳值可用逗號隔開。

呼叫

函數

[變數=] 函數名稱([引數1,引數2, ...])

#### G-8. 自定義函數範例

```
範例:計算 1<sup>3</sup> - 2<sup>3</sup> + 3<sup>3</sup> - 4<sup>3</sup> + ...... n<sup>3</sup>
# function
def calc(n):
     \cdotans = 0.
  : for i in range (1, n+1):
            if i\%2 == 1: ans = ans + i**3
            else: ans = ans - i**3
      return ans
# main program
x = calc(11)
                                 # result
                                  756
print(x)
```

#### 函數練習

SecurityFocusOnline CTF

[Python101]

函數的虛擬碼實作

skyoj 程式練習

P12 [Uva11417] GCD





















#### H-1.數字系統轉換

- int(<str>[, base=10]): 其它進位表達式(字串)轉10進位整數
   範例: int('A6', 16) ⇒ 166、int('356') ⇒ 356
- hex(<int>):10進位數字轉16進位表達式(字串)。範例:hex(34) ⇒ '0x22'
- str(<int>):10進位數字轉10進位表達式(字串)。
   範例:str(34) ⇒ '34'
- bin(<int>):10進位數字轉2進位表達式(字串)。範例:bin(34) ⇒ '0b100010'

## H-2.ASCII編解碼

ord(<chr>>): ASCII編碼, 將字元轉成ASCII碼(10進位整數)。
 範例: ord('A') ⇒ 65

chr(<int>): ASCII解碼, 將ASCII碼(10進位整數)轉成字元。
 範例: chr(65) ⇒ 'A'

### H-3.Base64編解碼

需匯入 base64模組, 語法: import base64

● b64encode(<bytes-like str>):Base64編碼。

範例: base64.b64encode(b'ABCD') ⇒ b'QUJDRA=='

● b64decode(<bytes-like str>): Base64解碼。

範例: base64.b64decode(b'TWFu') ⇒ b'Man'

# H-4.字元字串與bytes字串的差別

比較項目	字元字串	bytes字串
字串元素	Unicode字元	0~255的整數(8位元)
用處	一般文字處理	網路資料、電腦檔案等 二進位資料
前綴	u 或 U (可省略)	b 或 B
範例	'Hello'	b'Hello'

### H-5.字元字串與bytes字串範例

```
# 字元字串
s, t = 'Hello',
for x in s:
  t += chr(ord(x)+3)
print(t)
 輸出 Khoor
```

```
# bytes字串
s, t = b'Hello', ''
for x in s:
t += chr(x+3)
print(t)
 輸出 Khoor
```

## H-6.字元字串與bytes字串互轉

<str>.encode():字元字串轉成bytes字串。

範例: 'ABCD'.encode() ⇒ b'ABCD'

● *<bytes-like str***>.decode()**: bytes字串轉成字元字串。

範例:b'TWFu'.decode() ⇒ 'TWFu'

### 編解碼應用練習

SecurityFocusOnline CTF

[編碼102] 全

[Crypto102] CRY11, CRY14



















### I-1.解謎式CTF - PPC

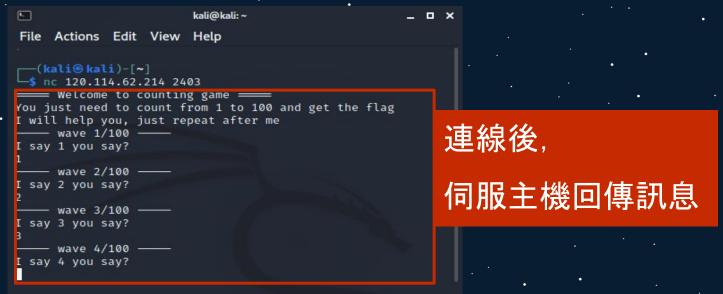
- PPC 全名 Professionally Program Coder
- 部份CTF初賽會新增 PPC 類型題目
- 主要以程式解題,與一般程式設計競賽不同
  - <mark>程式競賽</mark>:程式碼送至評分主機, 評分主機會將測試資料輸入給程. 式, 比對程式執行結果與答案是否相同
  - CTF競賽:透過程式與伺服器連線,以即時互動方式給出答案,若答對,則伺服器端會給出.flag

### I-2.PPC 題目範例

X Challenge 1 Solves count 50 你會數一到一百嗎? 題目會給定一個連線指令 nc 120.114.62.214 2403 Submit Flag

#### I-3.PPC 題目解題步驟

- 1. 先於 Shell 輸入連線指令看看回傳什麼訊息
- 2. 再依回傳訊息撰寫程式連線



### I-4.pwntools 套件

- 一個基於CTF框架和漏洞利用的 Python 套件
- 旨在讓使用者簡單快速的編寫exploit (漏洞利用)
- 安裝 pwntools 套件
  - \$ pip install pwntools

# I-5. 匯入 pwntools 套件

依據官方說明文件

from pwn import \*

## I-6.pwntools 進行 nc 連線

格式:

連線名稱 = remote('IP address', port)

範例:r = remote('120.114.62.214', 2403)

### I-7-1.remote連線後接收資料

- recvn(N):接受 N 個字元<sup>→</sup>
- recvline():接收一行資料
- recvlines(N):接收 N 行資料
- recvuntil(some\_string):接收到 some\_string 為止

## I-7-2.remote連線後接收資料(範例)

```
from pwn import * # 匯入pwntools套件
```

```
r = remote('120.114.62.214', 2403)  # 進行 nc 連線
```

```
r.recvline() # 從連接端接收一行資料
```

r.recvlines(7) # 從連接端接收七行資料

r.recvuntil(b'answer: ') # 從連接端接收資料; 直到 'answer: '字串

#字串 'answer: '加上前綴b, 代表此為Bytes字串(python3字串格式)

### I-8-1.remote連線後傳送資料

- send(payload): 發送payload(需為字串)
- **sendline(payload)**: 發送payload, 並換行(末尾\n)
- sendafter(some\_string, payload):接收到 some\_string 後, 發送 payload
- sendlineafter(some\_string, payload):接收到 some\_string 後, 發送 payload並換行

## I-8-2.remote連線後傳送資料(範例)

```
from pwn import * # 匯入pwntools套件
```

```
r = remote('120.114.62.214', 2403) # 進行 nc 連線
```

```
for i in range(100): # 跑 100 次迴圈
```

r.recvlines(2) # 從連接端接收2行資料

r.sendline(str(i+1).encode()) # 將i+1轉字串傳送至連接端

# encode()函數: 將文字字串轉換成Bytes字串

## I-9.remote連線取得最後flag方法

1. interactive():與 shell 互動,輸入 logout 離開連接端 shell 範例:r.interactive() # 回到 shell,可取得 flag

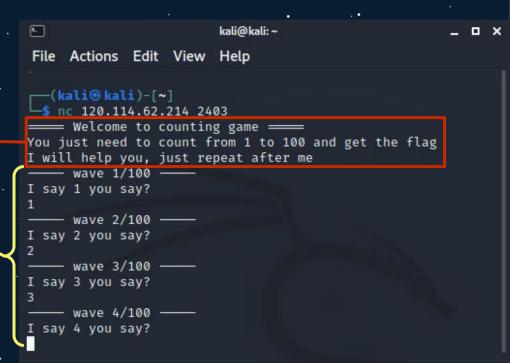
2. 用 recvline() 接收 flag 後再印出

範例:a=r.recvline() #將接收的flag 存到變數a

print(a.decode()) # 將a轉成字元字串再輸出至螢幕

#### I-10-1.題目count - 解題範例

- 1. 先以 shell 連線, 看看從連接端接收什麼訊息·
- 首先有3行題目敘述,
   接著有100個詢問,
   每個詢問有2行,
   第1行為計數、第2行為問題
- 3. 改以 python 程式碼, 透過 remote() 來進行 nc 連線



#### I-10-2.題目count - 解題範例

r.close()

1. gedit 編修一個python檔案

```
(ksu® Kali-20212-64)-[~]
$ gedit count.py
```

· 2. 存檔後, 編譯該檔案

```
(ksu® Kali-20212-64)-[~]
$ python3 count.py
```

3. 執行完畢, flag 會出現在螢幕上

```
from pwn import *
r = remote('120.114.62.214', 2403)
r.recvlines(3)
for i in range(100):
 r.recvlines(2)
  r.sendline(str(i+1).encode())
print(r.recvline().decode())
```

### I-11.remote()連線注意事項

- 1. 透過 python 的 remote() 函數連線, 傳送接收資料皆為本機程式與連接端之間傳遞, 不會於董幕上顯示過程。
- 2. **若要於螢幕上顯示過程**,可於 remote()函數加上第三個參數 r = remote('120.114.62.214', 2409, level='debug') 或者使用 print()函數,將資料顯示於螢幕上 r.sendline(ans.encode())

3. 程式結尾, 請將 remote() 連線結束 r.close()

print(ans)

#### PPC練習

SecurityFocusOnline CTF

[PPC101] 全