Python
字串基本 String Basic

臺北科技大學資訊工程系

字串基本操作

- 使用 + 運算子　　結合字串
- 使用 *　　複製字串
- 使用 []　　擷取子字串

```
def strOp():
    print("Hello" + " World")
    name = "小明" * 2
    print(name)
    Hi = "Hi~ 您好!"
    print(Hi[1])
    print(Hi[4])
```

```
Hello World

小明小明

i
您
```

字串基本操作

- 擷取字串[start:end:step]，
  - start ~ (end - 1), each step
  - start ~ (前一個), each step
  - 在 start, end, 放 -1 ，表示倒數過來第一個

```
def strGet():
    words = "This is a book, that is a cat~"
    print(words)
    print(words[::])
    print(words[5:])
    print(words[-4::])
    print(words[-14:20:1])
    print(words[-6:5:-1])
    print(words[-4:-10:-2])
```

```
This is a book, that is a cat~
This is a book, that is a cat~
is a book, that is a cat~
cat~
that
a si taht ,koob a s
cas
```

Exercise

- 輸入兩個英文句子 A, B
  - 將兩個英文句子 A, B 串聯成 C
  - 將 C 倒著輸出
  - C 從第五個字到最後第三個字，每隔兩個字輸出

```
def compute():
    A = input()
    B = input()          # 輸入兩個英文句子 A, B
    C = A + B            # 將兩個英文句子A, B串聯成 C
    print(C[::-1])       # 將 C 倒著輸出
    data = C[4:-2:2]     # C 從第五個字到最後第三個字，每隔兩個字輸出
    print(data)

compute()
```

輸入:
This is a book
That is a cat
輸出:
tac a si tahTkoob a si sihT
saboTa sac

資料型別-字串操作

- len()　　取得長度
- .replace()　　替換
- .split()　　切割字串，回傳 str list

```
def strOps():
    words = "小明今天去學校，小明遲到。"
    wordLength = len(words)
    print(wordLength)
    sentence = words.replace("小明","湯姆")
    print(words)
    print(sentence)
    wordsSplit = words.split("明")
    print(wordsSplit)
```

```
13

小明今天去學校，小明遲到。
湯姆今天去學校，湯姆遲到。

['小', '今天去學校，小', '遲到。']
```

Exercise

- 將兩個英文句子 A, B 串聯成 C
- 將 C 其中的 x 替換成 y，變成 D
- 輸出 C, D 長度的加總
- 輸出 C 前三個字，每一個字重複輸出三次。

```
def compute(A, B, x, y):
    C = A + B                          # 將兩個英文句子 A, B 串聯成 C
    D = C.replace(x, y)                # 將 C 其中的 x 替換成 y，變成 D
    print('D =', D)
    print(len(C) + len(D))             # 輸出 C, D 長度的加總
    data = C.split(' ')    # 輸出 C 前三個字 data[0], data[1], data[2]
    print(data)
    print(data[0]*3, data[1]*3, data[2]*3) # 每一個字重複輸出三次

def test():
    A = 'This is a book'   # 輸入兩個英文句子 A, B
    B = 'That is a cat'
    x = 'a'  # 輸入兩個英文字 x, y
    y = 'an'
    compute(A, B, x, y)

test()
```

```
C = This is a bookThat is a cat
D = This is an bookThant is an cant
58
['This', 'is', 'a', 'bookThat', 'is', 'a', 'cat']
ThisThisThis isisis aaa
```

字串存取、長度與複製

- 單一字元，是長度為一的字串。
- 字串可用雙引號"或用單引號'進行標示
- 使用索引位置[]可以存取元素
- len()　　取得長度
- ＊　　複製字串
- ＋　　串接

```
a = 'Hello \'ho '
b = "World "
c = "Bob said 'hey there.' "
print(a)
print(b + c)
print(a[1], a[4])
print(len(a))
print(b*3)
```

```
Hello 'ho
World Bob said 'hey there.'
e o
10
WorldWorldWorld
```

字串相等

- == operator or __eq__()

```
s1 = "apple"
if (s1 == "apple"): print("s1 is apple")
else:   print("s1 is NOT apple")

s2 = "apple"
if (s2 == "apple"): print("s2 is apple")
else: print("s2 is NOT apple")

s3 = "Apple"
if (s3 == "apple"): print("s3 is apple")
else: print("s3 is NOT apple")

s4 = "apple"
if (s1.__eq__(s4)): print("s1 equals to s4")
else: print("s1 does NOT equal to s4")
```

```
s1 is apple
s2 is NOT apple
s3 is NOT apple
s1 equals to s4
```

字串資料型別特性

- 字串是不可變的
  - 無法直接在字串中，插入子字串。
  - >>> name = 'Henny'
  - >>> name[0] = 'P’ ✗

- 若要改變字串，須用字串函式，如：replace() or slice
  - .replace()替換

```
name = 'Hernny'
data = name.replace('n', 'o')
print(name)
print(data)
full = name[:3] + 'ing'
print(full)
```

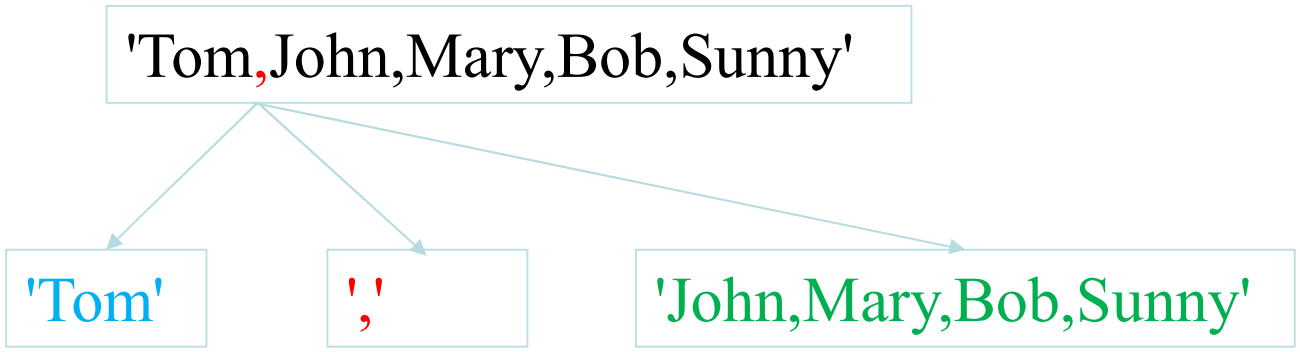
```
Hernny
Herooy

Hering
```

字串替換與切割

- str.partition(sep)
  - 字串切割，只切割第一個符合 sep 引數的子字串，形成 3-tuple

```
names = 'Tom,John,Mary,Bob,Sunny'
print(names.partition(','))
print(names.partition(',')[-1])
print(names.partition(',')[0])
print(names.partition(',')[1])
print(names.rpartition(',')) # 從後面
```

```
('Tom', ',', 'John,Mary,Bob,Sunny')
John,Mary,Bob,Sunny
Tom
,
('Tom,John,Mary,Bob', ',', 'Sunny')
```

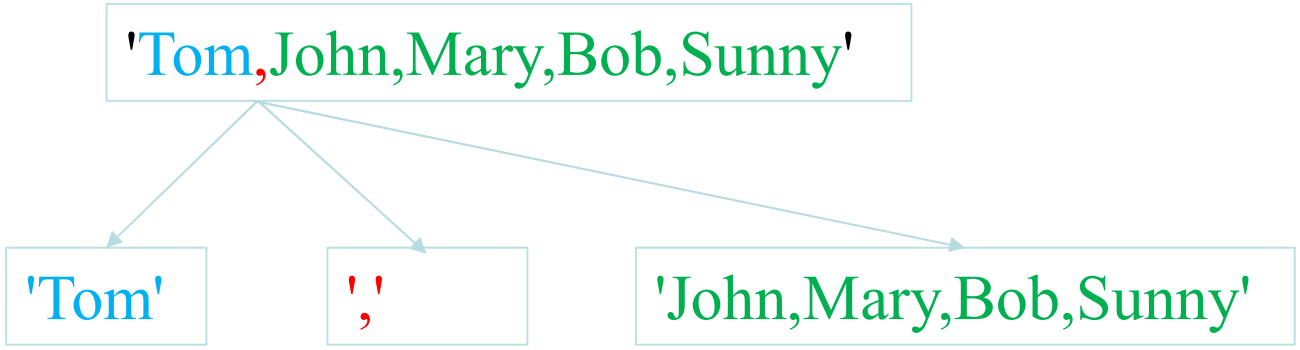


字串替換與切割

- str.partition(sep)
  - 字串切割，只切割第一個符合 sep 引數的子字串，形成 3-tuple

```
names = 'Tom,John,Mary,Bob,Sunny'
print(names.partition(','))
print(names.partition(',')[-1])
print(names.partition(',')[0])
print(names.partition(',')[1])
print(names.rpartition(',')) # 從後面
```

```
('Tom', ',', 'John,Mary,Bob,Sunny')
John,Mary,Bob,Sunny
Tom
,
('Tom,John,Mary,Bob', ',', 'Sunny')
```



字串替換與切割

- str.replace(old, new[, count])
  - 替換 string 中特定字串，
  - old 被替換的字串，
  - new 要替代的字串，
  - count 指定代替的數目

```
str = "It is istring example"
print (str)
print (str.replace("is", "was"))
print (str.replace("is", "was", 1))
```

```
It is istring example
It was wastring example
It was istring example
```

字串替換與切割

- string.split(sep, maxsplit)
  - 由左至右，將 string 字串以 sep 字串為分隔，進行分割
  - 若找不到符合 sep 的值，回傳整個字串

```
s = input('Enter 2 numbers separated by a space: ').split('a')
print(s)
```

```
Enter 2 numbers separated by a space: 123 456
['123 456']
```

  - 若找到符合 sep 的值，回傳 str list 字串串列

```
num1, num2 = input('Enter 2 numbers separated by a space: ').split()
num1 = int(num1)
num2 = int(num2)
print(num1, num2)
```

```
Enter 2 numbers separated by a space: 123 456
123 456
```

```
names = 'Tom,John,Mary,Bob,Sunny'
print(names.split(','))
```

```
['Tom', 'John', 'Mary', 'Bob', 'Sunny']
```

字串替換與切割

- string.split(sep, maxsplit)

```
def strOps():
    words = "小明今天去學校，小明遲到。"
    wordLength = len(words)
    print(wordLength)
    sentence = words.replace("小明","湯姆")
    print(words)
    print(sentence)
    wordsSplit = words.split("明")
    print(wordsSplit)
```

```
13

小明今天去學校，小明遲到。
湯姆今天去學校，湯姆遲到。

['小', '今天去學校，小', '遲到。']
```

字串替換與切割

- str.splitlines(keepends) # 將一行文字切割下來
  - 將字串進行切割
  - 以 "\n" 和 "\r" 作為分割的區隔符號
  - 回傳序列資料型別
  - keepends 引數預設 False，不會連同 '\n' 一併回傳
  - keepends 引數設為 True ，　會連同 '\n' 一併回傳

```
s = "Thank you for the music\nWelcome to the jungle\nMa"
print(s.splitlines())
print(s.splitlines(True))
```

```
['Thank you for the music', 'Welcome to the jungle', 'Ma']
['Thank you for the music\n', 'Welcome to the jungle\n', 'Ma']
```

Exercise

- 計算字串有多少字

```
s = 'Given a string and count how many words in the string?'
words = s.split()
print(words)
print(len(words))
```

```
['Given', 'a', 'string', 'and', 'count', 'how', 'many', 'words', 'in', 'the', 'string?']
11
```

- 計算字串有多少非重複字，可使用集合

```
s = 'Given a string and count how many words in the string'
words = s.split()
nonDuplicate = set(words)
print(nonDuplicate)
print(len(nonDuplicate))
```

```
{'words', 'the', 'Given', 'count', 'many', 'a', 'how', 'and', 'string', 'in'}
10
```

字串檢查

- in
- not in

```
txt = "The rain in Spain stays mainly in the plain"
x = "ain" in txt
print(x)
```

True

```
txt = "The rain in Spain stays mainly in the plain"
x = "ain" not in txt
print(x)
```

False

Exercise

- 檢查 'e' 是否在 'Umbrella'

```
s = 'Umbrella'
if 'e' in s:
  print("True")
else:
  print("False")
```

```
True
```

- 檢查輸入是否為 'python'

```
s = input('Enter a string: ')
if s == 'python':
  print("True")
else:
  print("False")
```

```
Enter a string: C
False
```

字串判斷

- str.isalpha(): 判斷是否為全為 字母(a~z, A~Z)
- str.islower(): 判斷是否為全為 小寫字母
- str.isspace(): 判斷是否為全為 空白
- str.isalnum(): 判斷是否為全為 字母或數字 *alphanumeric*

```
s = '505'
s1 = '2com'
s2 = "\u00B2" # unicode 十六進位
print(s.isdigit())
print(s2.isnumeric())
Print(s.isdecimal())
Print(s1.isalnum()) # alphanumeric
print(s1.isalpha())  # a~z, A~Z
```

```
True
True
True
True
False
```

```
s = "50800"
print(s.isdigit())
s = "abcd"
print(s.islower())
s = "  "
print(s.isspace())
```

```
True

True

True
```

字串判斷

- isdigit()
  - True: Unicode數字，byte數字(單字節)，全形數字(雙字節)
  - False: 漢字數字
  - Error: 無
- isdecimal()
  - True: Unicode數字，，全形數字(雙字節)
  - False: 漢字數字
  - Error: byte數字(單字節)
- isnumeric()
  - True: Unicode數字，全形數字(雙字節)，漢字數字
  - False: 無
  - Error: byte數字(單字節)

```
a = "\u0030"  # unicode for 0
b = "\u00B2" # unicode for ²

print(a.isdigit())
print(b.isdigit())
```

```
True
True
```

```
a = "\u0030" # unicode for 0
b = "\u0047" # unicode for G

print(a.isdecimal())
print(b.isdecimal())
```

```
True
False
```

字串判斷

```
num = "1"  # unicode
print(num.isdigit())          # True
print(num.isdecimal())        # True
print(num.isnumeric())        # True

num = " １ " # 全形
print(num.isdigit())          # True
print(num.isdecimal())        # True
print(num.isnumeric())        # True

num = "四" # 漢字
print(num.isdigit())          # False
print(num.isdecimal())        # False
print(num.isnumeric())        # True

num = b"1" # byte string
print(num.isdigit())          # True
print(num.isdecimal())        # AttributeError 'bytes' object has no attribute 'isdecimal'
print(num.isnumeric())        # AttributeError 'bytes' object has no attribute 'isnumeric'
```

```
>>> print('中文'.encode('utf-8'))
b'\xe4\xb8\xad\xe6\x96\x87'

>>> print(b'\xe4\xb8\xad\xe6\x96\x87'.decode('utf-8'))
中文
```

字串轉換

- string.lower()
  - 將string內的字母從大寫轉換小寫
- string.upper()
  - 將string內的字母從小寫轉為大寫

```
s = "Hello my FRIENDS"
print(s.lower())
print(s.upper())
```

```
hello my friends
HELLO MY FRIENDS
```

字串轉換

- string.capitalize ()
  - 將字串內第一個單字第一個字母[a-z]轉換成大寫
- string.title()
  - 將字串內所有單字第一個字母[a-z]轉換成大寫

```
s = "Hello my FRIENDS"
print(s.capitalize())
print(s.title())
```

```
Hello my friends
Hello My Friends
```

- string.swapcase()
  - 將 string 字串裡的字母大小寫互轉

```
s = "Hello my FRIENDS"
print(s.swapcase())
```

```
hELLO MY friends
```

Exercise

- 使用者輸入任意整數 n，
- 當輸入值 n 不為整數，
  - 提示使用者輸入資料型別錯誤，且不斷重新讓使用者輸入，
- 若輸入值 n 為整數，
  - 將其 print 至螢幕上，例如 n = 100

```
while(True):
  n = input('請輸入一個整數：')
  if n.isdigit():
    print(n)
    break
  else:
    print('型別錯誤，請輸入整數')
```

Exercise

- 計算字串大小寫數字與特殊符號
  - input_str = "P@#yn26at^&i5ve"

```
def findDigitsCharsSymbols(inputString):
  charCount = 0
  digitCount = 0
  symbolCount = 0
  for char in inputString:
    if char.isalpha():     #    if char.islower() or char.isupper():
      charCount += 1
    elif char.isnumeric():
      digitCount += 1
    else:
      symbolCount += 1
  print("Chars = ", charCount, "Digits = ", digitCount, "Symbol = ", symbolCount)

findDigitsCharsSymbols("P@#yn26at^&i5ve")
```

```
Chars =  8          Digits =  3          Symbol =  4
```

字串切片

- 擷取字串[start:end:step]，start~(end-1), each step
  - -1 表示倒數過來第一個

```
def strGet():
    words = "This is a book, that is a cat~"
    print(words)
    print(words[::])
    print(words[5:])
    print(words[-4::])
    print(words[-14:20:1])
    print(words[-6:5:-1])
    print(words[-4:-10:-2])
```

```
This is a book, that is a cat~
This is a book, that is a cat~
is a book, that is a cat~
cat~
that
a si taht ,koob a s
cas
```

字串切片

- 切片 Slicing

```
b = "Hello, World!"
print(b[2:5])
```

llo

```
b = "Hello, World!"
print(b[-5:-2])
```

orl

- string.split() 將字串依指定的字串符號切割，回傳 str list

```
s3 = "This is a sentence."
s3_split = s3.split(' ') # 空格為切割符號
print (s3_split)
```

```
['This', 'is', 'a', 'sentence.']
```

字串連結

- Join all items in a tuple into a string

```
a = ("how", "are", "you")
x = "-".join(a)
print(x)
```

```
how-are-you
```

- Join all items in a tuple into a string, using a hash character 井字號 as separator:

```
myTuple = ("John", "Peter", "Vicky")
x = "#".join(myTuple)
print(x)
```

```
John#Peter#Vicky
```

Exercise 1

- 輸入兩個英文句子 A, B，兩個英文單字 x, y
  - 將兩個英文句子 A, B 串聯成 C
  - 將 C 其中的 x 替換成 y，變成 D
  - 輸出 C, D 長度的加總
  - 輸出 C 前三個字，每一個字重複輸出三次。
  - C 從第五個字到最後第三個字，每隔兩個字輸出
  - 將 C 倒著輸出

Input

英文句子 A : This is a book.
英文句子 B : That is a cat .
英文單字 x: is
英文單字 y: was

```
def f(A, B, x, y):
    C = A + B
    D = C.replace(x, y)
    print(len(C)+len(D))
    R = C.split()
    print((R[0]+' ')*3, (R[1]+' ')*3, (R[2]+' ')*3)
    print(C[::-1])
    print(C[5:-2:2])
```

```
57
This This This  is is is  a a a
tac a si tahTkoob a si sihT
i  okhti
```

Exercise 2

- output?

```
def f(data, s):
    length = len(data)
    for i in range(length): # i = 0, 1, 2, 3, 4
        newData = data[0:i] + s + data[i+1:length]
        print(newData)

f('abced','X')
```

```
Xbced
aXced
abXed
abcXd
abceX
```

Exercise 3

- 輸入 s1, s2, 在 s1 中間加入 s2 後輸出

```
def appendMiddle(s1, s2):
  middleIndex = int(len(s1) /2)
  print("Original Strings are", s1, s2)
  middleThree = s1[:middleIndex] + s2 + s1[middleIndex:]
  print("After appending new string in middle", middleThree)

appendMiddle("Chrisdem", "IamNewString")
```

```
Original Strings are Chrisdem IamNewString
After appending new string in middle ChriIamNewStringsdem
```

Exercise 4

- 程式
  - 輸入 My name is Michele
  - 輸出 Michele is name My

解答

```
def main():
    words = input('Enter: ')
    wordsSplit = words.split(' ')
    print(wordsSplit)
    wordsSplit = wordsSplit[::-1];
    print(wordsSplit)
    wordsSplit = ' '.join(wordsSplit)
    print(wordsSplit)

main()
```

```
Enter: My name is Michele

['My', 'name', 'is', 'Michele']

['Michele', 'is', 'name', 'My']

Michele is name My
```

Exercise 4

- 輸入字串，判斷是否為迴文?
- Example of palindrome: madam

```
wrd = input("Please enter a word: ")
wrd = str(wrd)
rvs = wrd[::-1]
# print(rvs)
if wrd == rvs:
    print("This word is a palindrome")
else:
    print("This word is NOT a palindrome")
```

```
Please enter a word: abcba
This word is a palindrome
```

```
def reverse(word):
    x = ''
    for i in range(len(word)):
        x += word[len(word)-1-i]
    return x

word = input('Give me a word:\n ')
x = reverse(word)
if x == word:
    print('This is a Palindrome')
else:
    print('This is NOT a Palindrome')
```

```
Ggive me a word: abcba
This is a Palindrome
```

join()

- string.join(iterable)

```
myList = ["name","John","country","Norway"]
mySeparator = 'TEST'

x = mySeparator.join(myList)

print(x)
```

```
nameTESTJohnTESTcountryTESTNorway
```

```
myTuple = ("John", "Peter", "Vicky")

x = "#".join(myTuple)

print(x)
```

```
John#Peter#Vicky
```

多行字串

- 指定多行字串
  - 三個單引號/雙引號，或 \ +

```
a = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua."""
print(a)
```

```
a = '''Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.'''
print(a)
```

```
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

With line break

多行字串

- 指定多行字串
  - 三個單引號/雙引號，或 \ +

```
a = "Lorem ipsum dolor sit amet, \
consectetur adipiscing elit, \
sed do eiusmod tempor incididunt \
ut labore et dolore magna aliqua."
print(a)
```

```
a = ("Lorem ipsum dolor sit amet," +
"consectetur adipiscing elit," +
"sed do eiusmod tempor incididunt" +
"ut labore et dolore magna aliqua.")
print(a)
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
```

No line break

END