



# Python

## 列表與for迴圈

賴璉錡

lclai.t11@o365.fcu.edu.tw

# List 列表

- List是Python中很常用的一種資料結構，可以將多種形態的資料同時儲存在一個可修改的序列中。
- List的建立方式只要將逗號分隔的數值使用[]括起來
  - Ex: [1, 2, 3, 'a', 'b', 'c', True, False]
- List有順序性，每個list都有索引值index，並且從0開始

index	0	1	2	3	4	5	6	7
value	1	2	3	'a'	'b'	'c'	True	False

# 索引和切片

- Python可以透過索引值來取得某個位置的元素
  - Ex: `my_list[0]`
- List有slice的方式來取得部分原list的片段

```
my_list = [1, 2, 3, 'a', 'b', 'c', True, False]

my_list[0] # 1
my_list[:3] # [1, 2, 3]
my_list[2:] # [3, 'a', 'b', 'c', True, False]
my_list[2:4] # [3, 'a']
my_list[:-1] # [1, 2, 3, 'a', 'b', 'c', True]
```

## Practice8: List 操作 練習

- 新增一個列表，並且使用 `print()` 方法來看以下索引值操作之結果

```
my_list = [1, 2, 3, 'a', 'b', 'c', True, False]
first_value = my_list[0]
fifth_value = my_list[4]

print(first_value)
print(type(first_value))
print(fifth_value)
print(type(fifth_value))
print(my_list[2])
print(my_list[:3])
print(my_list[2:])
print(my_list[2:4])
print(my_list[:-1])
```

# List 新增、插入、刪除、修改

- `append()`：在列表的尾端加入一個元素
  - `my_list = [1, 2, 3]`
  - `my_list.append(5)`
  - `print(my_list) # [1, 2, 3, 5]`
- `insert()`：在列表中的任意位置插入一個元素
  - `my_list = [1, 2, 3]`
  - `my_list.insert(2, 5)`
  - `print(my_list) # [1, 2, 5, 3]`

# List 新增、插入、刪除、修改

- Remove()：刪除列表中的元素
  - `my_list = [1, 2, 3, 2, 2]`
  - `my_list.remove(2)`
  - `print(my_list) # [1, 3, 2, 2]`
  - `my_list.remove(2)`
  - `print(my_list) # [1, 3, 2]`
- 修改列表中的某個index的值
  - `my_list = [1, 2, 3, 2, 2]`
  - `my_list[3] = 5`
  - `print(my_list) # [1, 2, 3, 5, 2]`

## Practice9: List 新增、插入、刪除、修改

- 新增一個列表，進行新增、插入、刪除、修改操作練習

```
my_list = [1, 'apple', 3]
my_list.append(5)
print(my_list)
```

```
my_list = [1, 'apple', 3]
my_list.insert(2, 5)
print(my_list)
```

```
my_list = [1, 'banana', 3, 2, 2]
my_list.remove("banana")
print(my_list)
my_list.remove(2)
print(my_list)
```

```
my_list = [1, 'apple', 3, 2, 2]
my_list[1] = 5
print(my_list)
```

# List的一些方法

- `len()` : 返回列表的長度
  - Ex: `len(my_list)` # 8
- `sort()` : 排序列表中的元素
  - `age_list = [5, 8, 6, 3, 10, 15, 2, 5]`
  - `age_list.sort()`
  - `print(age_list)` # [2, 3, 5, 5, 6, 8, 10, 15]
- `reverse()` : 反轉列表中的元素順序
  - `age_list.reverse()`
  - `print(age_list)` # [15, 10, 8, 6, 5, 5, 3, 2]



# List的一些方法

- `count()`：返回指定元素在列表出現的次數
  - `age_list = [5, 8, 6, 3, 10, 15, 2, 5]`
  - `age_list.count(5) # 2`
- `index()`：返回指定元素在列表中第一次出現的索引
  - `age_list = [5, 8, 6, 3, 10, 15, 2, 5]`
  - `age_list.index(6) # 2`

## Practice10: List方法

- 新增一個列表，並且使用 `print()` 方法來看以下方法操作之結果

```
my_list = [1, 2, 3, 'a', 'b', 'c', True, False]
len(my_list)

age_list = [5, 8, 6, 3, 10, 15, 2, 5]

age_list.sort()
print(age_list)

age_list.reverse()
print(age_list)

age_list = [5, 8, 6, 3, 10, 15, 2, 5]
age_list.index(6)
```

# range()

- range () 方法可以產生一個可以迭代 (iterable) 的物件
- range(stop)
- range(start, stop[, step])
  - range(10) = range(0, 10, 1)
    - 從0開始 (0有算) 數到10結束 (10不算) , 每次加1
    - 結果: [0, 1, 2, 3, 4, 5, 6 , 7 , 8 , 9] # 10個數字
  - range(2, 20, 2)
    - 從2開始 (2有算) 數到20結束 (20不算) , 每次加2
    - 結果: [2, 4, 6, 8, 10, 12, 14 , 16 , 18 ] # 9個數字

## Practice 11: range方法

- 使用 `range` 方法產生 `range` 的物件，並將其轉換成 `list` 後用 `print()` 印出

```
rag = range(10)
list_rag = list(rag)
print(list_rag)

rag2 = range(2, 10, 5)
list_rag2 = list(rag2)
print(list_rag2)

rag3 = range(20, 0, -4)
list_rag3 = list(rag3)
print(list_rag3)

rag4 = range(20, -1, -4)
list_rag4 = list(rag4)
print(list_rag4)
```

# For 迴圈

- For 迴圈可以從頭開始迭代每一個在列表中的元素

```
my_list = [1, 2, 3, 'a', 'b', 'c', True, False]
for element in my_list:
    print(element)
```

```
1
2
3
a
b
c
True
False
```

element 是指在遍歷的過程中每個元素的代稱

## Practice12: for loop

- 新增一個list後，使用for迴圈print出每一個元素

```
my_list = [1, 2, 3, 'a', 'b', 'c', True, False]
for element in my_list:
    print(element)
```

```
1
2
3
a
b
c
True
False
```

# Practice13: for loop

- 目前有兩個列表，分別是姓名與年齡

- `name_list = ['Joe', 'Mary', 'Andy', 'Dan']`
- `age_list = [15, 23, 35, 48]`

- 請依照順序印出
  - <姓名>今年<年紀>歲
- 提示：
  - 因為兩個list同樣長度，所以for迴圈設定相同次數，並將次數設成取值的索引值
  - 取值方法可參考[p3](#)

```
name_list = ['Joe', 'Mary', 'Andy', 'Dan']  
age_list = [15, 23, 35, 48]
```

```
Joe 今年 15 歲  
Mary 今年 23 歲  
Andy 今年 35 歲  
Dan 今年 48 歲
```