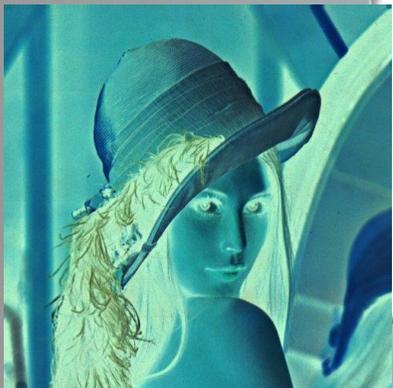




# Programming

von anwendeng



# List & Slicing for Python

---



串列清單 (list)：收集一連串資料的容器

中括號括起的資料結構 (data structure), 便是所謂的串列 (list)。

```
dices=['', 'DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png',  
print(type(dices))
```

## 建立串列

串列稱為容器 (container), 收納的值叫元素 (element)。

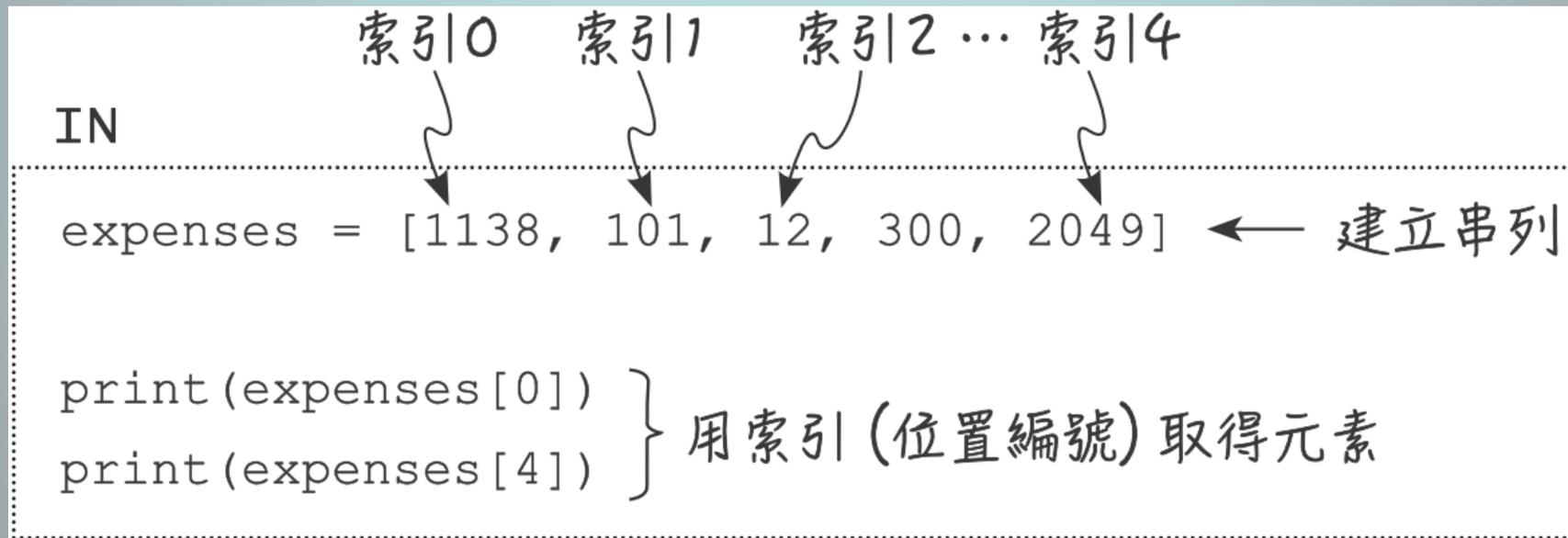
注意中括號 [] 內的值得用逗號分隔

```
IN  
lst = [10, 20, 30, 40, 50]  
print(lst)
```

建立串列變數 lst

索引就是元素位置的編號，最末元素的索引就是  $N - 1$ 。

用索引 (index) 存取串列元素



也可以用索引來更改元素的值

## 用索引 (index) 存取串列元素

IN

```
expenses[2] = 120
```

```
print(expenses)
```

OUT

```
[1138, 101, 120, 300, 2049]
```

第二個元素值已從12改成120了

串列切片 slicing：擷取串  
列中某範圍的一些元素

```
print(dices[2])  
print(dices[-1])  
print(dices[0:6:2])  
print(dices[6:0:-1])
```

資料組名稱[索引|起始值：索引|結束值（不含）：  
索引|值改變量]

使用切片slicing

```
dices=['', 'DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png', 'DICE06.png']  
print(type(dices))
```

```
<class 'list'>
```

```
print(dices[1:5])
```

```
dices=['', 'DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png', 'DICE06.png']
```

```
print(dices[2])  
print(dices[-1])  
print(dices[0:6:2])  
print(dices[6:0:-1])
```

DICE02.png

DICE06.png

['', 'DICE02.png', 'DICE04.png']

['DICE06.png', 'DICE05.png', 'DICE04.png', 'DICE03.png', 'DICE02.png', 'DICE01.png']

IN

```
expenses = [1138, 101, 12, 300, 2049] ← 採用原本 expenses 的值
```

```
print (expenses[0:3])
```

語法

串列[起始索引:起始索引+擷取個數:索引間隔]

IN

```
print (expenses[1:]) ← 等於 [1:5]
```

```
print (expenses[:3]) ← 等於 [0:3]
```

```
print (expenses[:]) ← 等於 [0:5]
```

```
print (expenses[::]) ← 等於 [0:5]
```

```
print (expenses[::2]) ← 取整個串列, 但每隔 2 個元素取一次
```

```
print (expenses[::-1]) ← 取整個串列, 但倒著順序取 (使串列顛倒)
```

# 串列資料的增刪、加總與排序

新增元素：`append()`

刪除及清空元素：`remove()`、`clear()`

`insert`

元素總和：`sum()`

元素數量：`len()`

最小與最大值：`max()`、`min()`

元素排序：`sorted()` 與 `sort()`

# 新增元素：append()

語法

串列名.append(新資料)

把新的元素(不限型別)附加在串列尾端。

```
[13]: dices=['', 'DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png', 'DICE06.png']
```

```
[14]: dices.append([0,1,2])
print(dices)
print(dices[-1][0])
```

```
['', 'DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png', 'DICE06.png', [0, 1, 2]]
0
```

```
[15]: dices.remove(dices[0])
print(dices)
```

```
['DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png', 'DICE06.png', [0, 1, 2]]
```

# 刪除及清空元素：remove()、clear()

IN

```
expenses.clear()  
print(expenses)
```

# insert(index, new element)

```
5]: dices.remove(dices[0])
print(dices)

['DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png', 'DICE06.png', [0, 1, 2]]

6]: dices.insert(0, 543)
print(dices)

[543, 'DICE01.png', 'DICE02.png', 'DICE03.png', 'DICE04.png', 'DICE05.png', 'DICE06.png', [0, 1, 2]]
```

# 元素總和：sum()

IN

```
expenses = [1138, 101, 12, 300, 2049] ← expenses 剛剛已被清空,  
現在把它 key 回來  
print(sum(expenses))
```

# 元素數量：len()

IN

```
expenses = [1138, 101, 12, 300, 2049]
```

```
print(len(expenses))
```

既繫數道總和與元素個數，要算平均就簡單了：

**IN**

```
print(sum(expenses) / len(expenses))
```

**OUT**

```
720.0
```

最小與最大值：max()、min()

**IN**

```
print (min (expenses) )
```

```
print (max (expenses) )
```

# 元素排序：sorted() 與 sort()

IN

```
s = sorted(expenses)
```

```
print(s, expenses) ← 把兩個串列印出來看
```

OUT

```
[12, 101, 300, 1138, 2049] [1138, 101, 12, 300, 2049]
```

*s 已排序好了*

*原本的 expenses 不變*

# 程式實作

von anwendeng

感謝觀賞

Herzlichen Dank für die  
Aufmerksamkeit

von anwendeng