

# List

hydai 8.15

# 想像一下

- 今天有三個資料要計算，所以我們會這樣寫
- $a = 1$
- $b = 2$
- $c = 3$
- .....那如果你有**成千上萬**筆資料呢？

# 再想像一下

- 不知道使用者要輸入多少資料又要怎麼辦呢？
- `in = input().....`
- `a = in[0]`
- `b = in[1]`
- .....???????? 這樣真的行嗎？

「神說要有 List 於是就有了 List」

– 來自神秘的訊息，因為太神祕所以我也忘了從哪來的了

# 繼續用你的想像力！

- 我們現在有個異次元的櫃子
- 他可以有無限的格子
- 第一層可以放東西
- 第二層也可以放東西
- 依此類推

# 也就是說這個櫃子

- 我們會用一個數字代表東西放在第幾層
- 這個數字就叫做 **index(索引)**
- 櫃子裡可以放任何東西

# 讓我們跨出櫃子

- 把櫃子換成 List 就會變成
- 放在第 **0** 層的變數、放在第 **1** 層的變數...
- 這個數字就是 **index(索引)** 而變數就是可以塞資料的地方

# List 長這樣

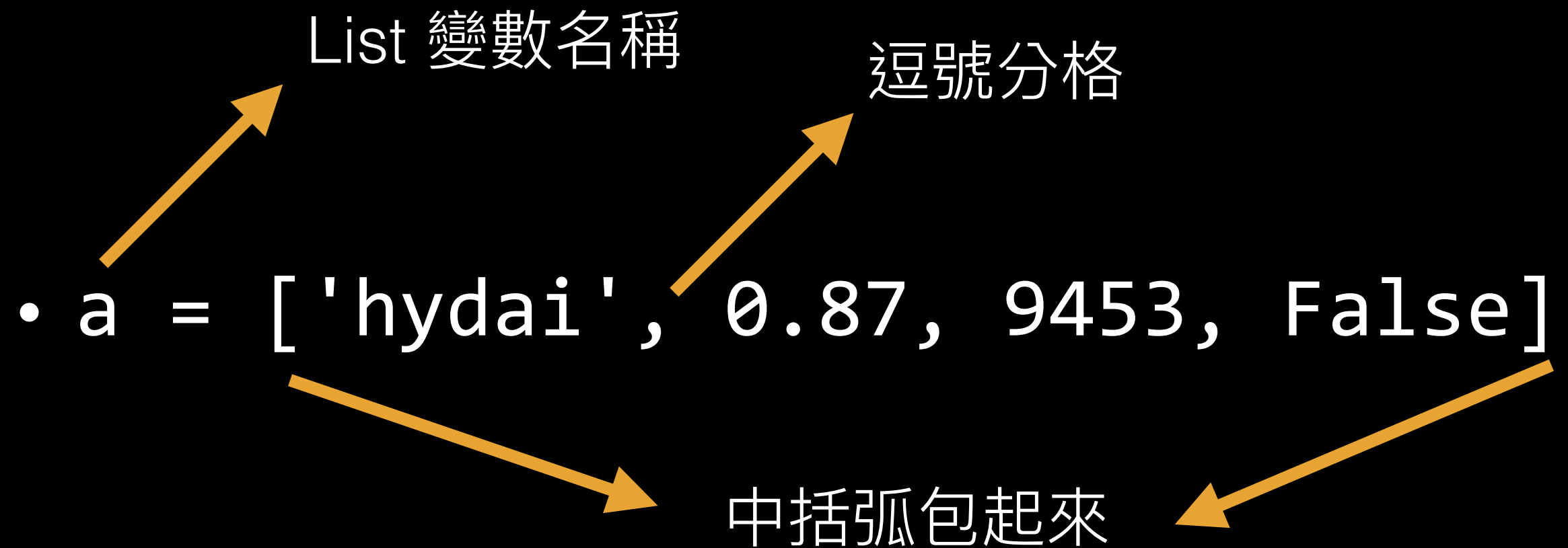
- `a = ['hydai', 0.87, 9453, False]`
- `a` 就是一個 List，~~不是櫃子，也不是什麼綠豆糕~~
- 第 0 格放的是 'hydai' 這個字串
- 第 1 格放的是 0.87 這個浮點數
- 第 2 格放的是 9453 這個整數
- 第 3 格放的是 False 這個布林值



# 有了格子的概念

- `a = ['hydai', 0.87, 9453, False]`
- 我們可以用 **`a[第幾格]`** 來取出變數
- `a[0]` 代表 'hydai'
- `a[3]` 代表 False

# List 的語法



# 試試看

```
In [10]: a = ['hydai', 54689, 94.87, True]
print(a)
print(a[0])
print(a[1])
print(a[2])
print(a[3])

['hydai', 54689, 94.87, True]
hydai
54689
94.87
True
```

# 那去拿一個不存在的格子呢？

```
a = ['hyundai', 54689, 94.87, True]
print(a)
print(a[0])
print(a[1])
print(a[2])
print(a[3])
print(a[4]) # 這行會發生什麼事呢？
```

# 不作死就不會死

```
-----  
--  
IndexError                                Traceback (most recent call las  
t)  
<ipython-input-11-5c7192474e1a> in <module>()  
      5 print(a[2])  
      6 print(a[3])  
----> 7 print(a[4]) # 這行會發生什麼事呢？  
  
IndexError: list index out of range
```

# 如何知道 List 有多長？

- 用 `len([List 變數名稱])`
- `a = [1, 3]`
- `len(a) = 2`

List + For loop

# 因為可以拿到 List 長度

- `for [變數名稱] in range(0, len(list)):`
- `[縮排] blahblah`



# 試試看

```
In [12]: a = ['hyundai', 54689, 94.87, True]
         for i in range(0, len(a)):
           print(a[i])
```

```
hyundai
54689
94.87
True
```

可以更簡潔

# in

- 因為有 in 這個關鍵字
- in 有「在...之中」的意思
- in range(0, 3) 代表在 0~2 之中

# 所以可以這樣寫

```
In [13]: a = ['hydai', 54689, 94.87, True]
         for i in a:
             print(i)
```

```
hydai
54689
94.87
True
```

# 練習題

- 現在有一個 list  $a = [1, 3, 5, 7, 9]$
- 1. 請對每個元素都平方後印出來
- 2. 且在遍歷之後， $a$  會變成  $[1, 9, 25, 49, 81]$

更多 List 的操作

# 建造一個空的 List

- `a = list()`
- `a = []`
- 這兩個都可以

# 動態增加元素

- `list.append(x)`
  - 把 `x` 塞到 `list` 的最後面
- `list.insert(i, x)`
  - 把 `x` 塞到 `i` 這個位置上



# 試試看

```
In [14]: a = [1, 3, 5]
print(a)
a.append(7)
print(a)
a.insert(1, 2)
print(a)
```

[1, 3, 5]  
[1, 3, 5, 7]  
[1, 2, 3, 5, 7]

# 拿掉特定位置的元素

- `list.pop()`
  - 把 `list` 的最後一格丟掉
- `list.pop(i)`
  - 把 `list` 的第 `i` 格丟掉

# 試試看

```
In [19]: a = [1,3,5,3,1]
print(a)
a.pop()
print(a)
a.pop(2)
print(a)
```

```
[1, 3, 5, 3, 1]
[1, 3, 5, 3]
[1, 3, 3]
```

# 拿掉某個格子

- `list.remove(x)`
  - 會把第一個出現的 `x` 拿掉
- `list.clear()`
  - Python 3.3 以後，這邊我們先用 `del a[:]`
  - 全部清光光～

# 試試看

```
In [15]: a = [1, 3, 5, 3, 1]
print(a)
a.remove(3)
print(a)
a.remove(3)
print(a)
a.clear()
print(a)
```

```
[1, 3, 5, 3, 1]
[1, 5, 3, 1]
[1, 5, 1]
[]
```

# 練習題

- 讓我們輸入五個普物段考成績（整數）
- 請個別輸出這五個普物段考成績
- 請輸出這五個成績的平均
- 請將這五個成績各自開根號乘以十以後輸出
- 算出新的成績平均
- Hint: 開根號可以用  $x^{**0.5}$

# 練習題的輸入與輸出

輸入

```
10  
30  
50  
70  
90
```

輸出

```
List:  
10  
30  
50  
70  
90  
Average: 50.0  
List after **0.5 * 10:  
31.622776601683796  
54.772255750516614  
70.71067811865476  
83.66600265340756  
94.86832980505139  
Average: 67.12800858586283
```

與常見函式的結合



# 試試看

- `max(list)`
- `min(list)`
- `sum(list)`

切片 Slice

# 拿出櫃子的一部份

- 語法：`list[start : end]`
- 拿到 list 的 `start`, `start+1`, ..., `end-2`, `end-1`

# 試試看

```
In [20]: a = [0, 1, 2, 3, 4, 5]  
print(a[0:6])  
print(a[3:5])
```

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

# 更多細節

- `list[start : end]`，`start` 跟 `end` 都可以省略不寫
- `start` 的預設為 0
- `end` 的預設為 `len(list)`
- `list[: end]` 代表 `0~end-1`
- `list[start : ]` 代表 `start~len(list)-1`
- `list[: ]` 代表 `0~len(list)-1`

# 試試看

```
In [21]: a = [0, 1, 2, 3, 4, 5]  
print(a[ :4])  
print(a[3: ])  
print(a[ : ])
```

```
[0, 1, 2, 3]
```

```
[3, 4, 5]
```

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

# 練習題

- 輸入十個整數
- 之後會有五次的詢問
- 每次會輸入開始的位置跟結束的位置
- 請輸出從開始位置到結束位置的總和

# 範例輸出入

Input List:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

From: 3

To: 10

Sum from 3 to 10 = 49

From: 1

To: 2

Sum from 1 to 2 = 2

From: 6

To: 8

Sum from 6 to 8 = 15

From: 0

To: 10

Sum from 0 to 10 = 55