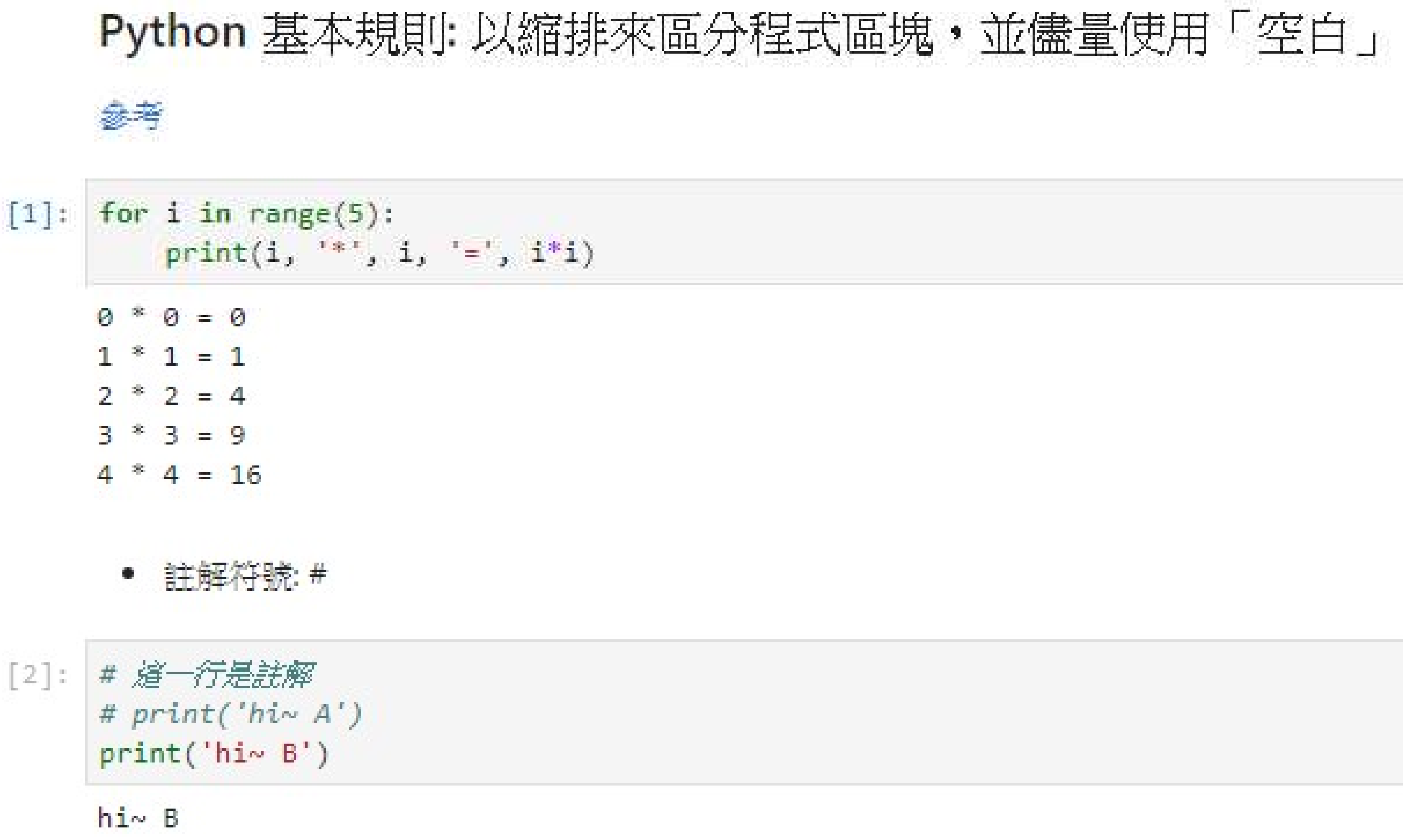
D01-2
基本語法練習

Speaker: Malo

參考資料

- A Byte of Python
- https://python.swaroopch.com/
- Python3 vs Python2？不用再想了，就是Python3了！

基本語法



Python 基本規則: 以縮排來區分程式區塊，並儘量使用「空白」

參考

[1]:

```
for i in range(5):
    print(i, '*', i, '=', i*i)
```

```
0 * 0 = 0
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
```

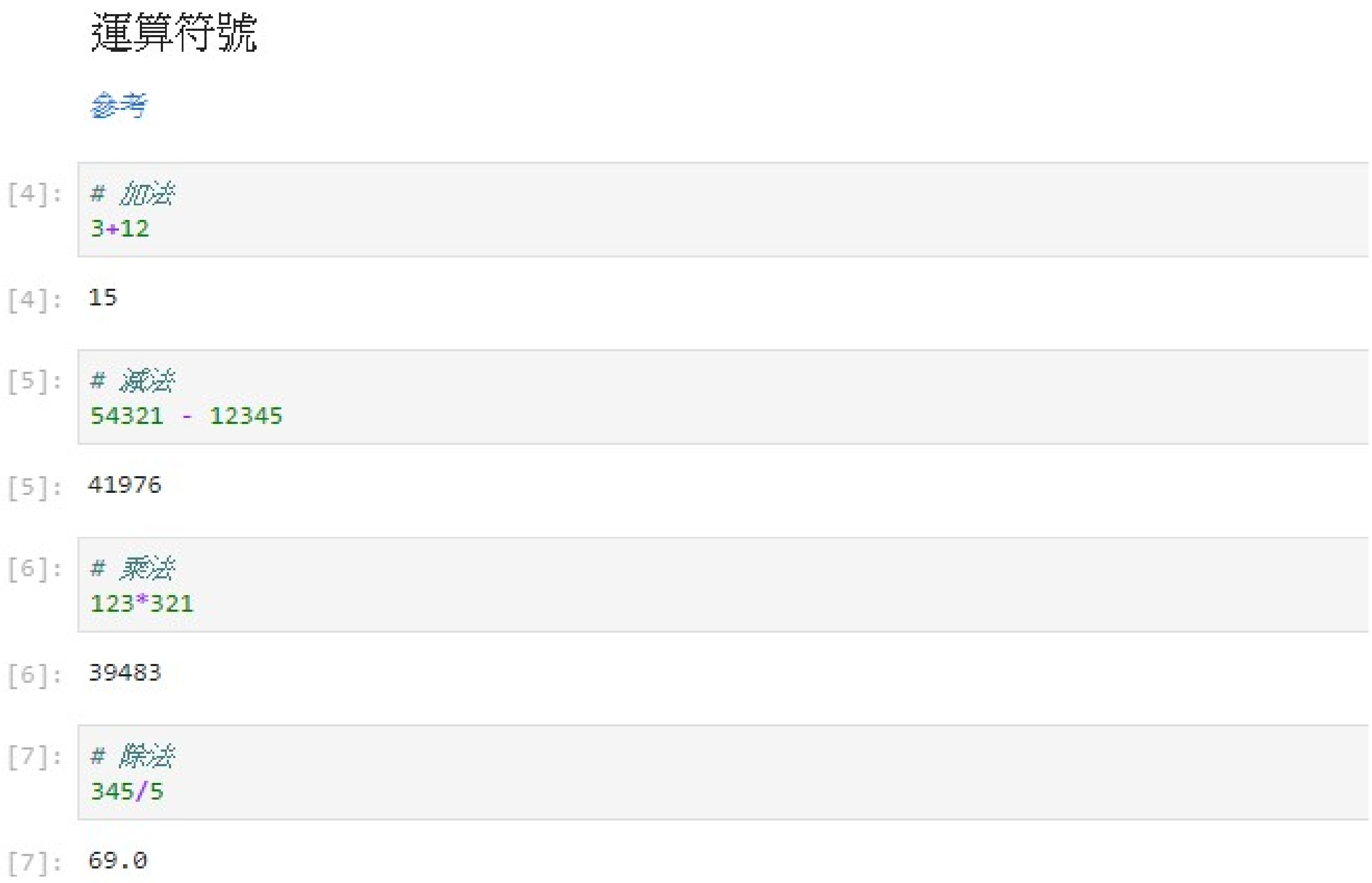
- 註解符號: #

[2]:

```
# 這一行是註解
# print('hi~ A')
print('hi~ B')
```

```
hi~ B
```

基本語法

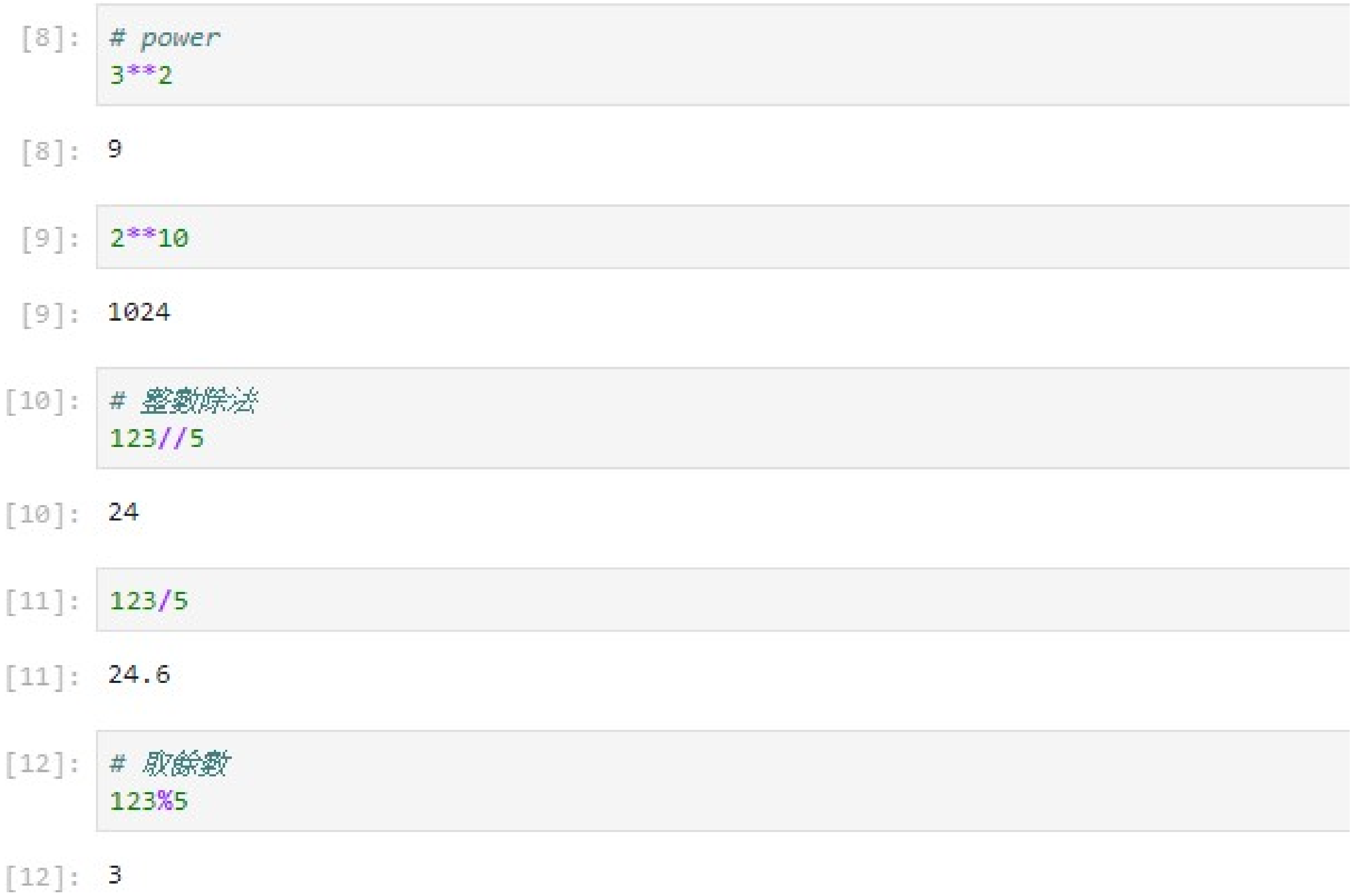


運算符號

參考

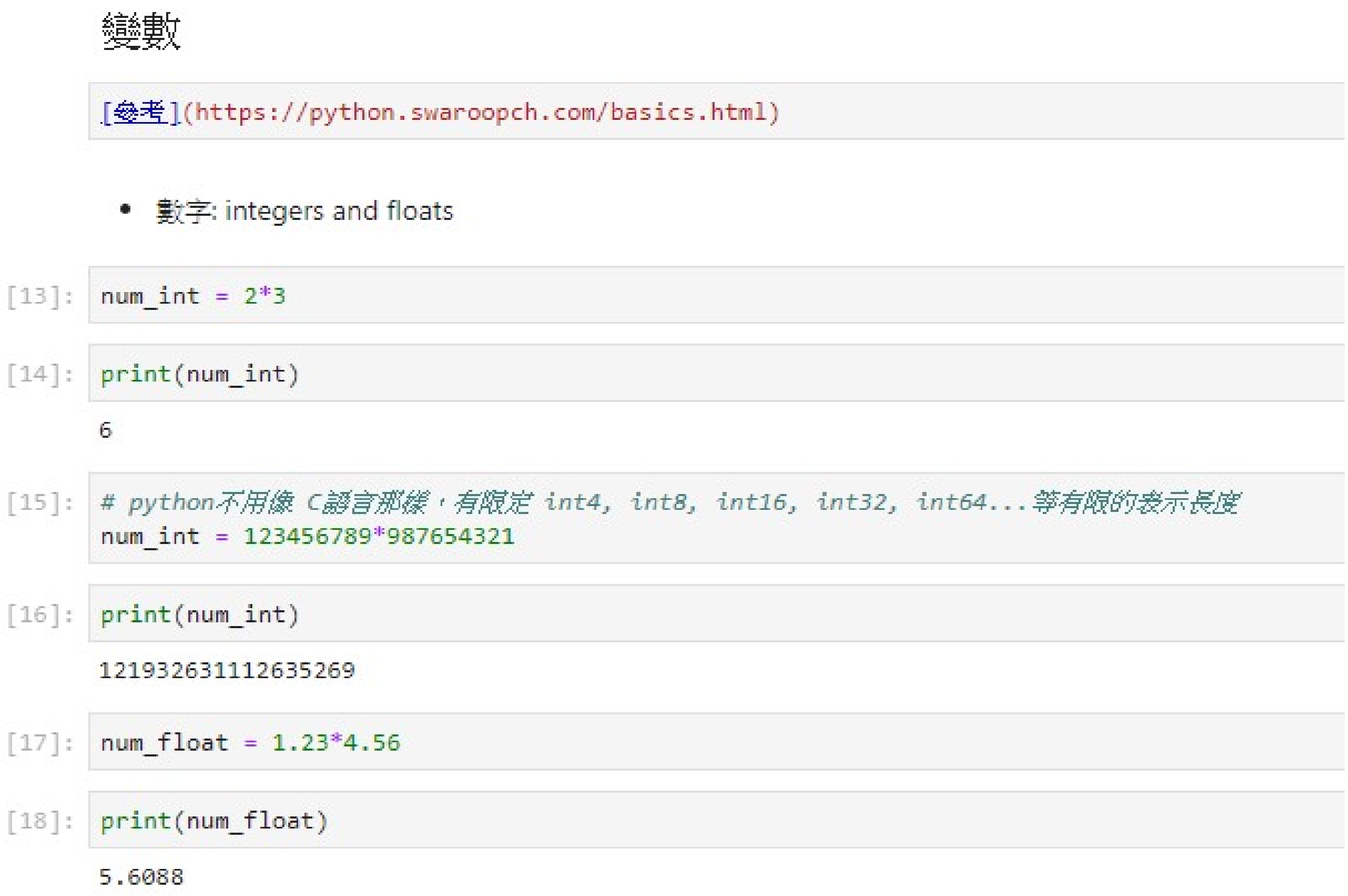
```
[4]: # 加法
     3+12

[4]: 15

[5]: # 減法
     54321 - 12345

[5]: 41976

[6]: # 乘法
     123*321

[6]: 39483

[7]: # 除法
     345/5

[7]: 69.0
```

基本語法



```
[8]: # power
     3**2

[8]: 9

[9]: 2**10

[9]: 1024

[10]: # 整數除法
      123//5

[10]: 24

[11]: 123/5

[11]: 24.6

[12]: # 取餘數
      123%5

[12]: 3
```

基本語法



變數

```
[參考](https://python.swaroopch.com/basics.html)
```

- 數字: integers and floats

```
[13]: num_int = 2*3
```

```
[14]: print(num_int)
```

```
6
```

```
[15]: # python不用像 C語言那樣，有限定 int4, int8, int16, int32, int64...等有限的表示長度
      num_int = 123456789*987654321
```

```
[16]: print(num_int)
```

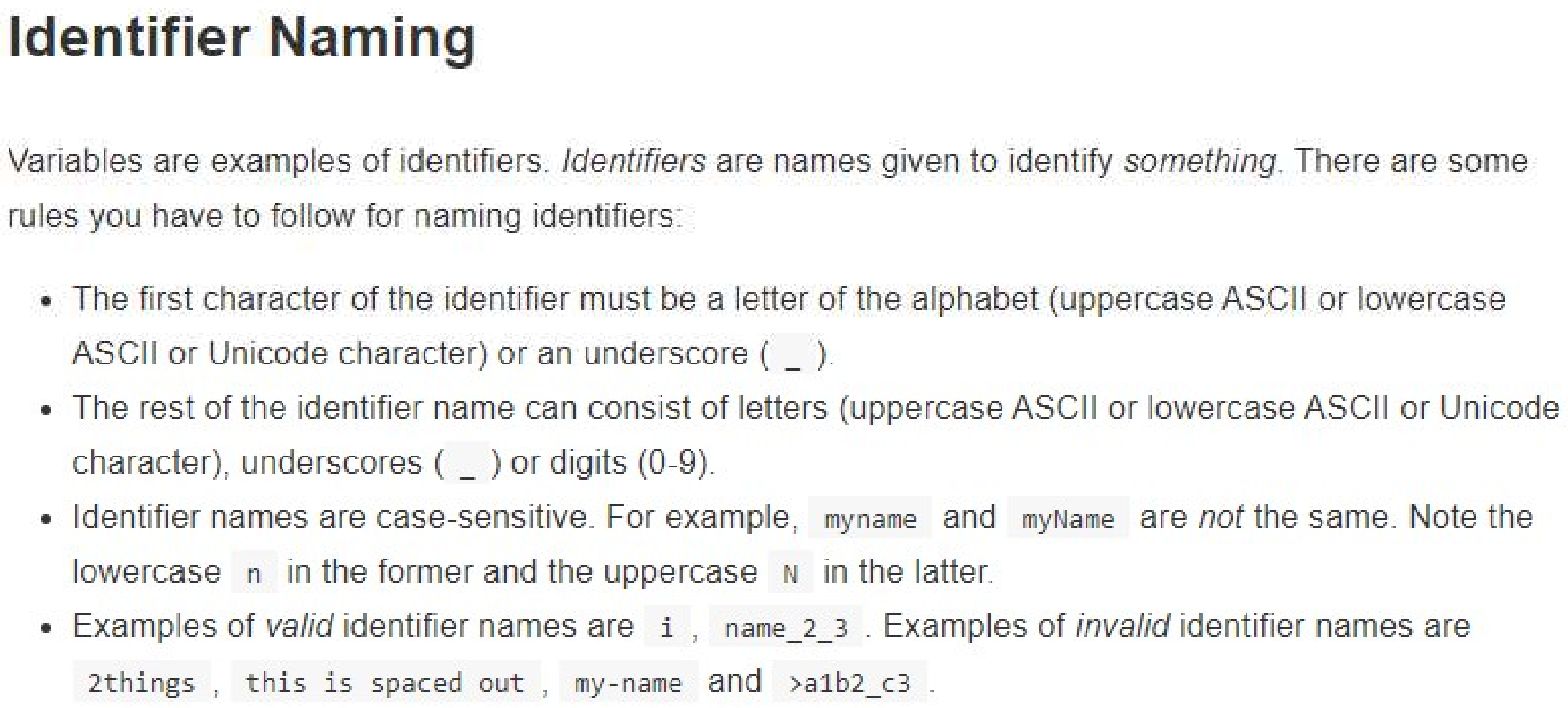
```
121932631112635269
```

```
[17]: num_float = 1.23*4.56
```

```
[18]: print(num_float)
```

```
5.6088
```

基本語法

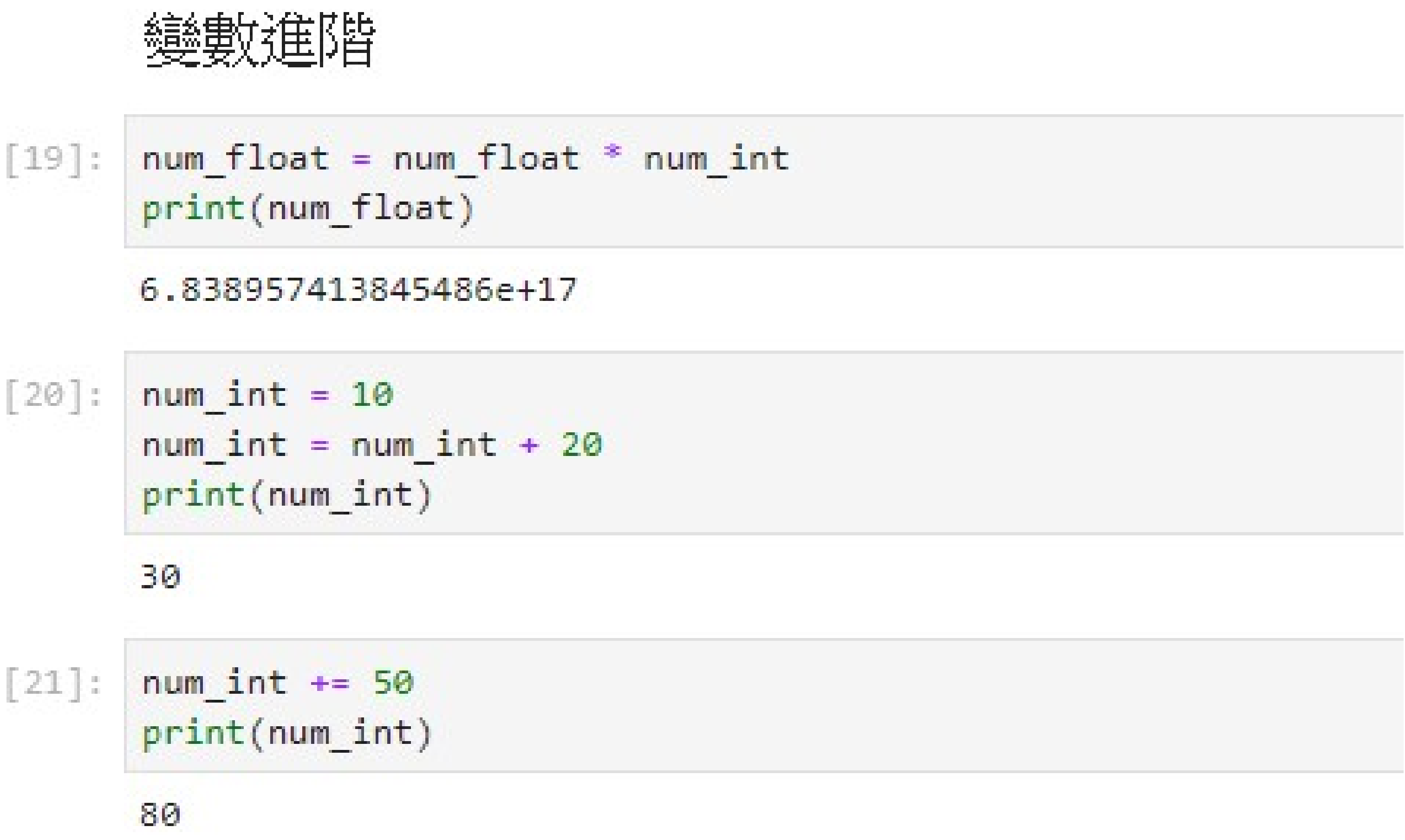


Identifier Naming

Variables are examples of identifiers. *Identifiers* are names given to identify *something*. There are some rules you have to follow for naming identifiers:

- The first character of the identifier must be a letter of the alphabet (uppercase ASCII or lowercase ASCII or Unicode character) or an underscore (`_`).
- The rest of the identifier name can consist of letters (uppercase ASCII or lowercase ASCII or Unicode character), underscores (`_`) or digits (0-9).
- Identifier names are case-sensitive. For example, `myname` and `myName` are *not* the same. Note the lowercase `n` in the former and the uppercase `N` in the latter.
- Examples of *valid* identifier names are `i` , `name_2_3` . Examples of *invalid* identifier names are `2things` , `this is spaced out` , `my-name` and `>a1b2_c3` .

基本語法



變數進階

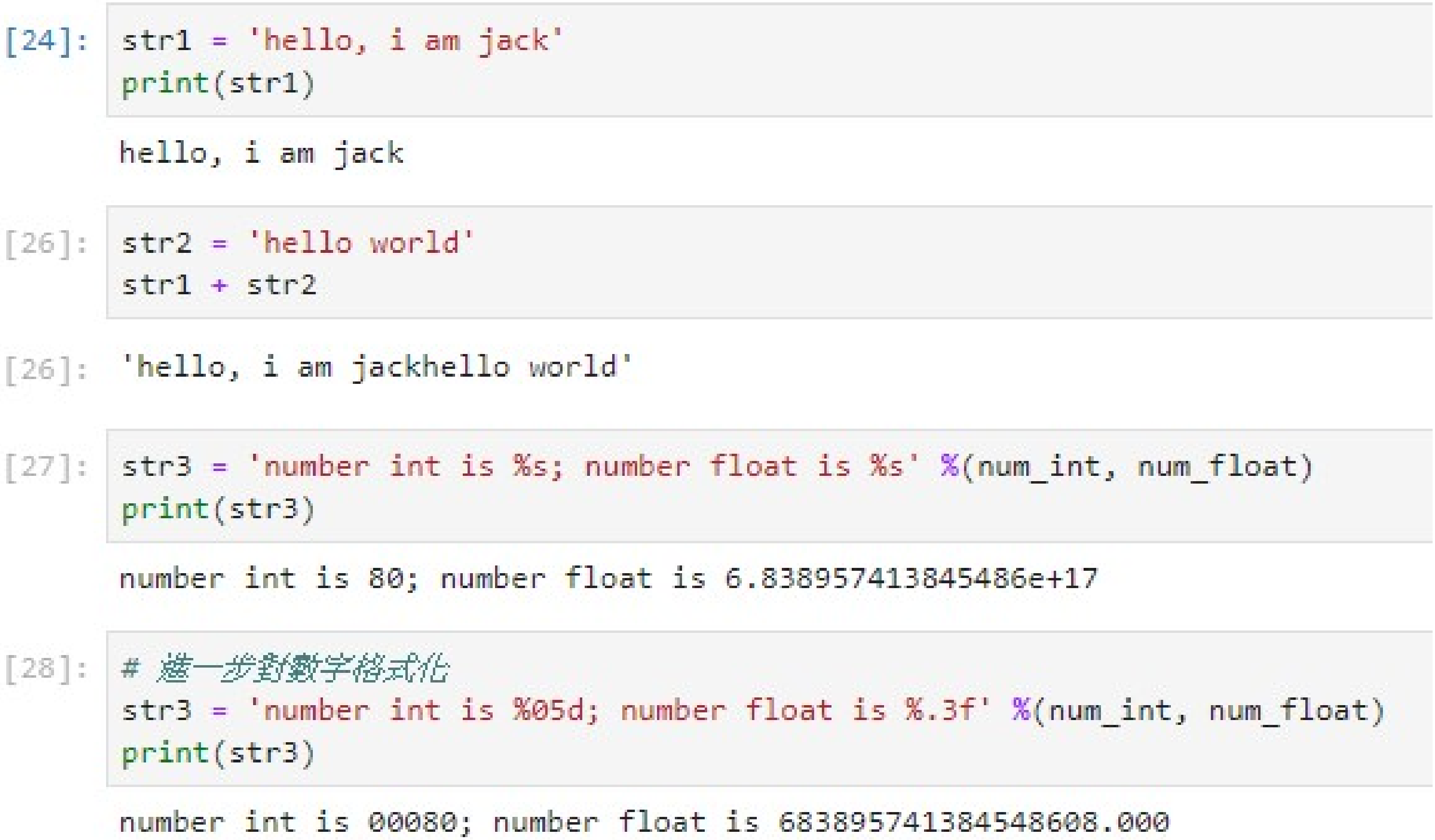
```
[19]: num_float = num_float * num_int
      print(num_float)

6.838957413845486e+17
```

```
[20]: num_int = 10
      num_int = num_int + 20
      print(num_int)

30
```

```
[21]: num_int += 50
      print(num_int)

80
```

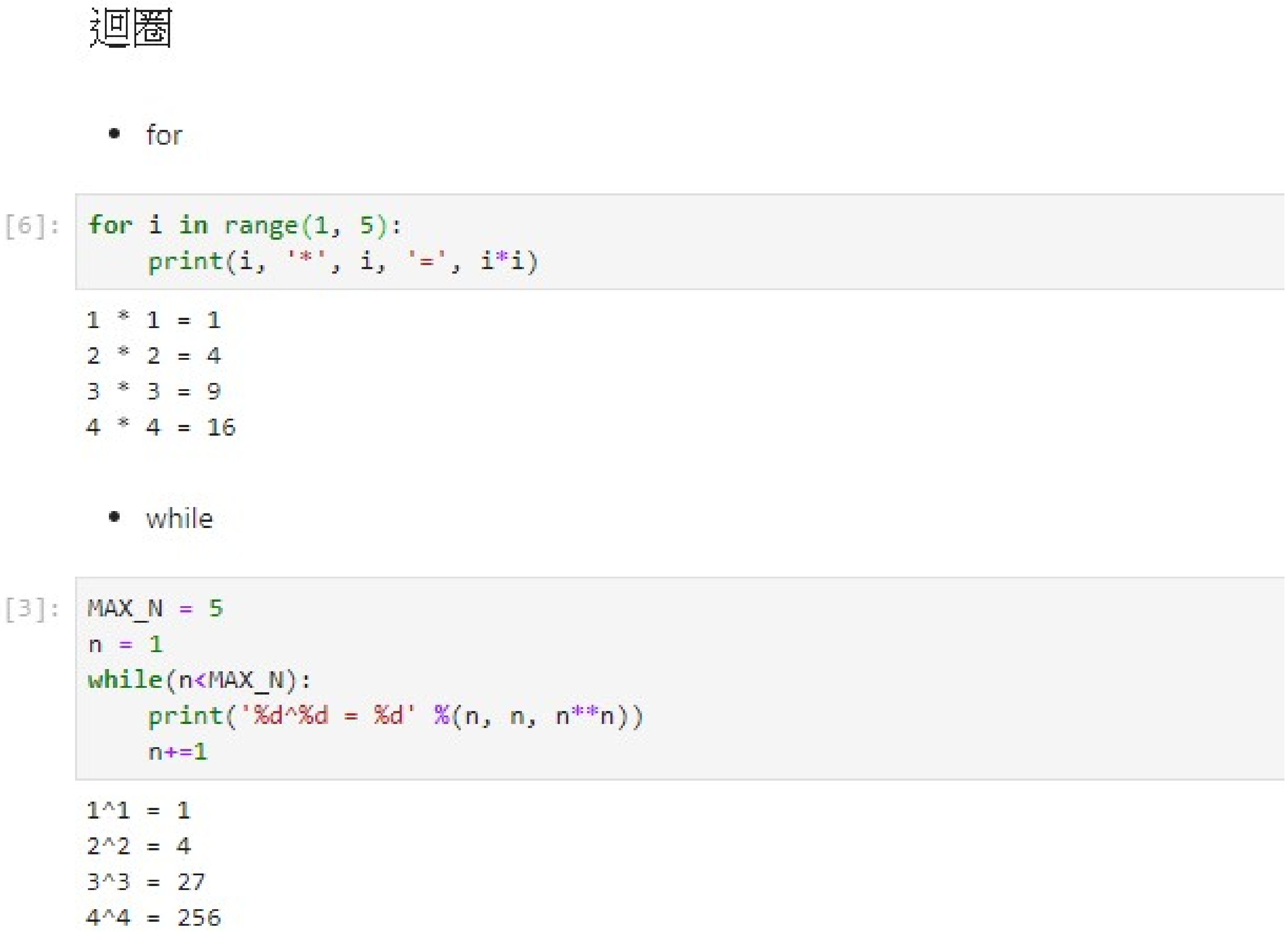
基本語法

字串



```
[24]: str1 = 'hello, i am jack'
      print(str1)

      hello, i am jack

[26]: str2 = 'hello world'
      str1 + str2

[26]: 'hello, i am jackhello world'

[27]: str3 = 'number int is %s; number float is %s' %(num_int, num_float)
      print(str3)

      number int is 80; number float is 6.838957413845486e+17

[28]: # 進一步對數字格式化
      str3 = 'number int is %05d; number float is %.3f' %(num_int, num_float)
      print(str3)

      number int is 00080; number float is 683895741384548608.000
```

基本語法



迴圈

- for

[6]:
```
for i in range(1, 5):
    print(i, '*', i, '=', i*i)
```

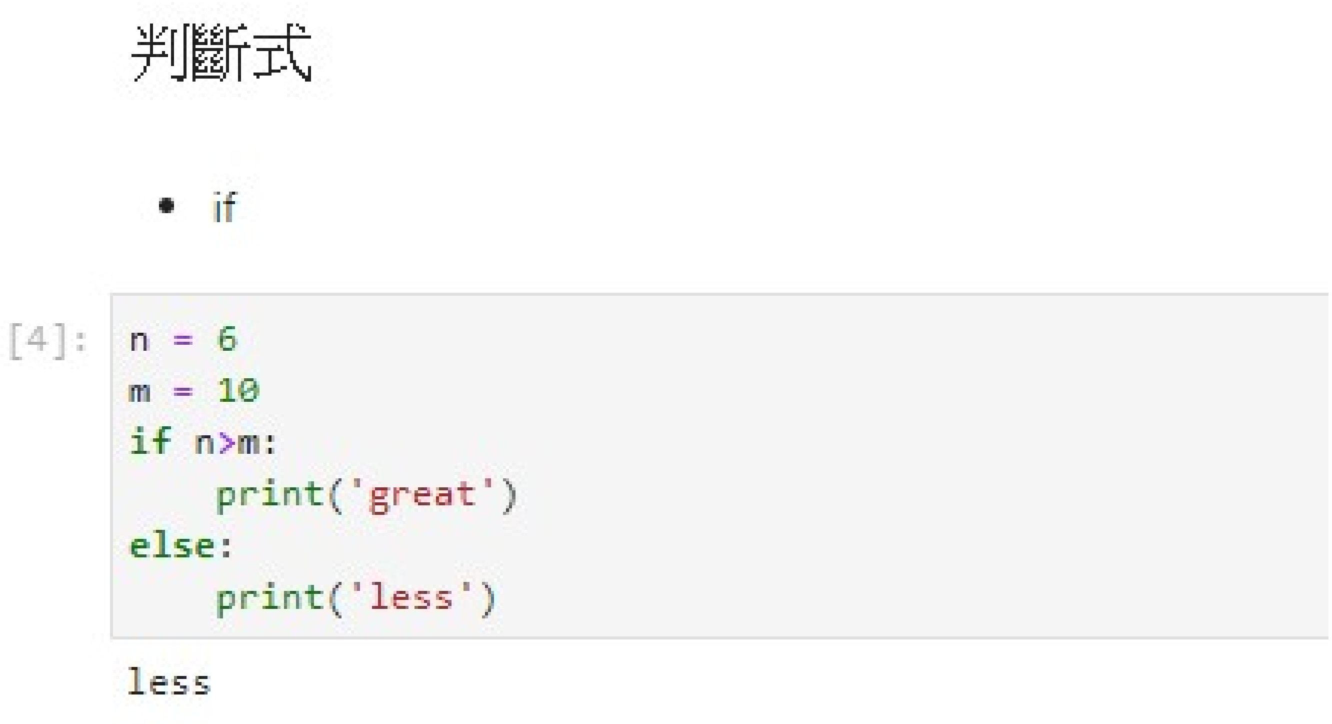
```
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
```

- while

[3]:
```
MAX_N = 5
n = 1
while(n<MAX_N):
    print('%d^%d = %d' %(n, n, n**n))
    n+=1
```

```
1^1 = 1
2^2 = 4
3^3 = 27
4^4 = 256
```

基本語法



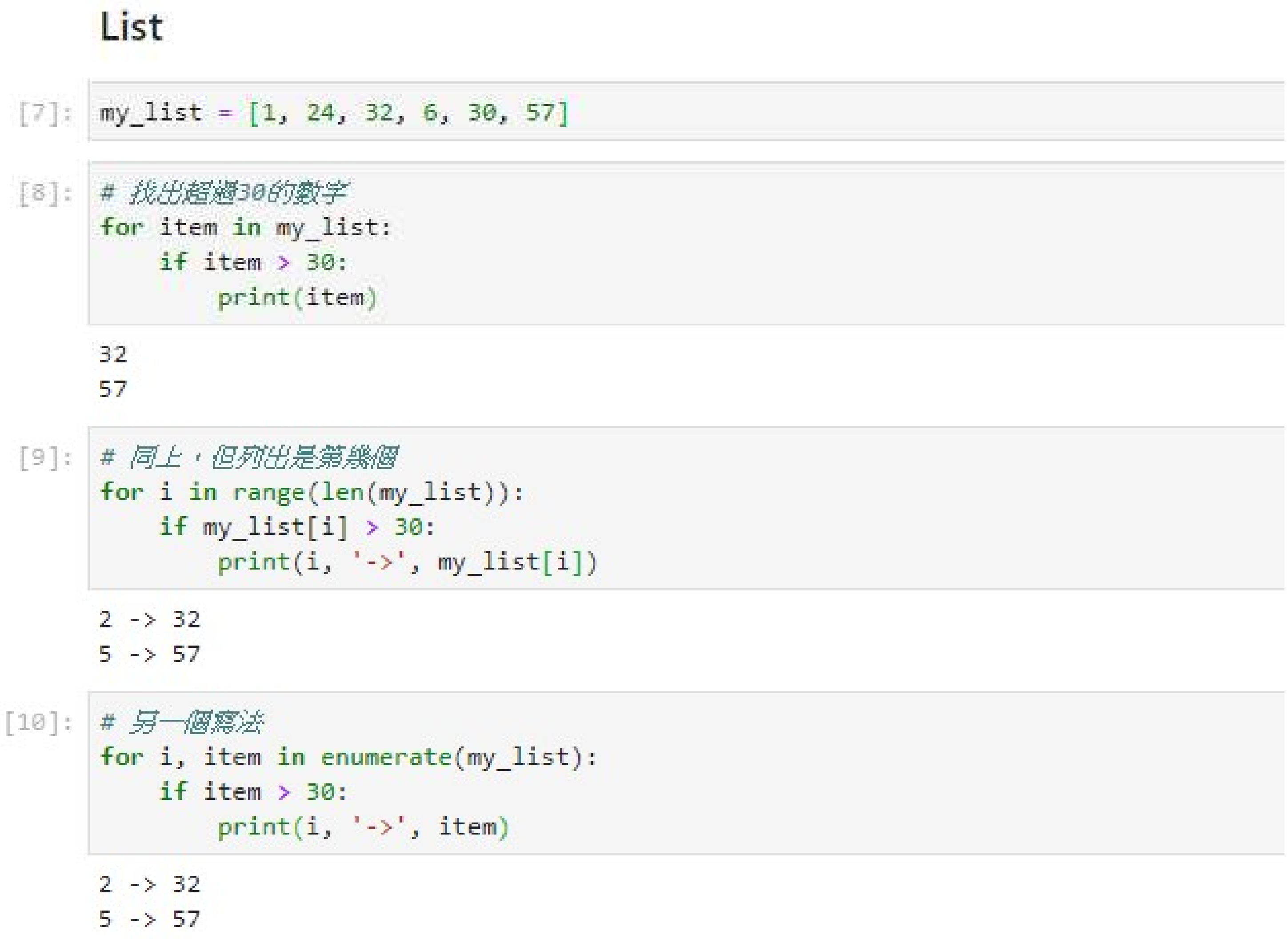
判斷式

- if

```
n = 6
m = 10
if n>m:
    print('great')
else:
    print('less')
```

```
less
```

基本語法



List

```
[7]: my_list = [1, 24, 32, 6, 30, 57]
```

```
[8]: # 找出超過30的數字
     for item in my_list:
         if item > 30:
             print(item)
```

```
32
57
```

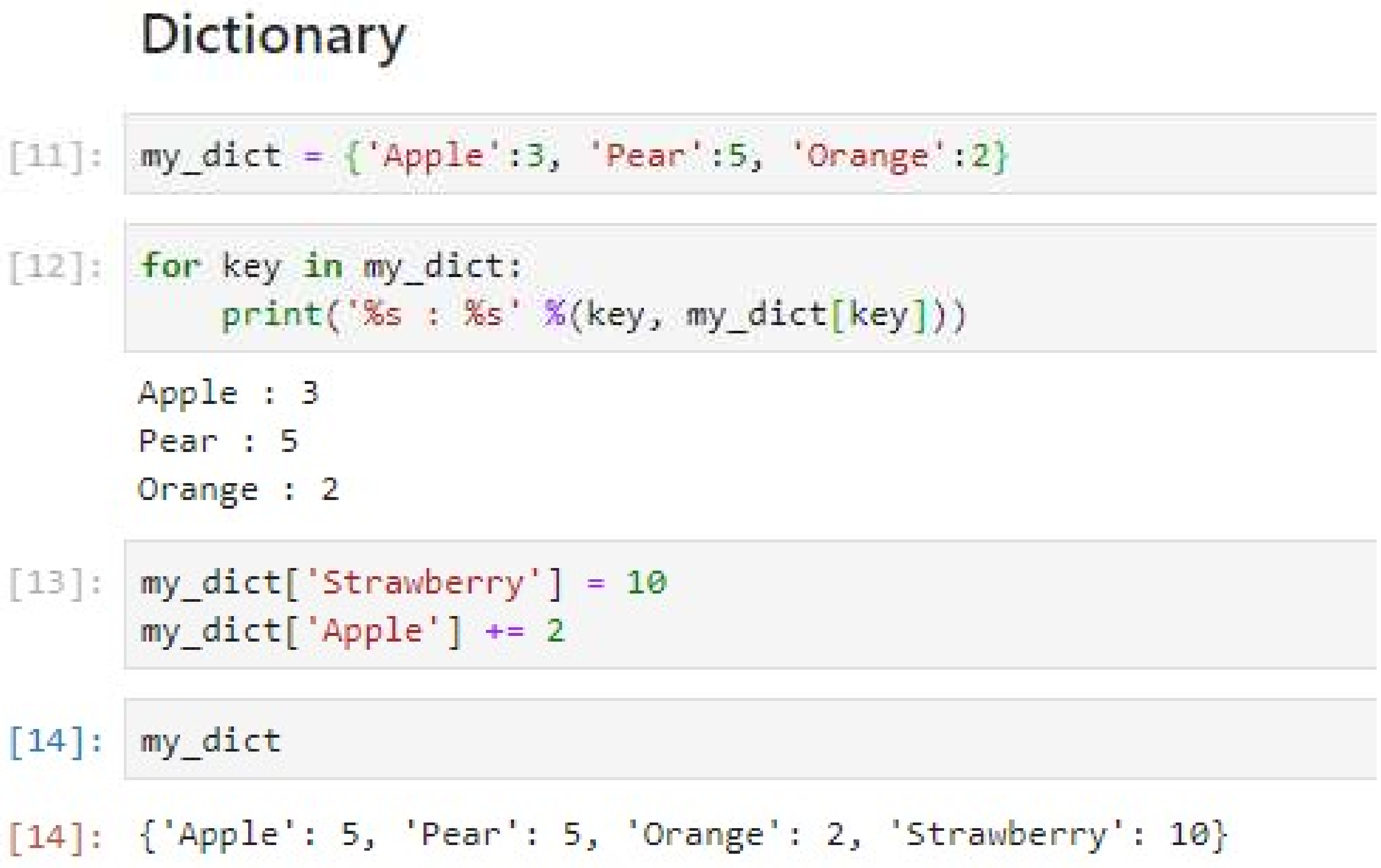
```
[9]: # 同上，但列出是第幾個
     for i in range(len(my_list)):
         if my_list[i] > 30:
             print(i, '->', my_list[i])
```

```
2 -> 32
5 -> 57
```

```
[10]: # 另一個寫法
      for i, item in enumerate(my_list):
          if item > 30:
              print(i, '->', item)
```

```
2 -> 32
5 -> 57
```

基本語法



Dictionary

```
[11]: my_dict = {'Apple':3, 'Pear':5, 'Orange':2}
```

```
[12]: for key in my_dict:
          print('%s : %s' %(key, my_dict[key]))
```

```
Apple : 3
Pear : 5
Orange : 2
```

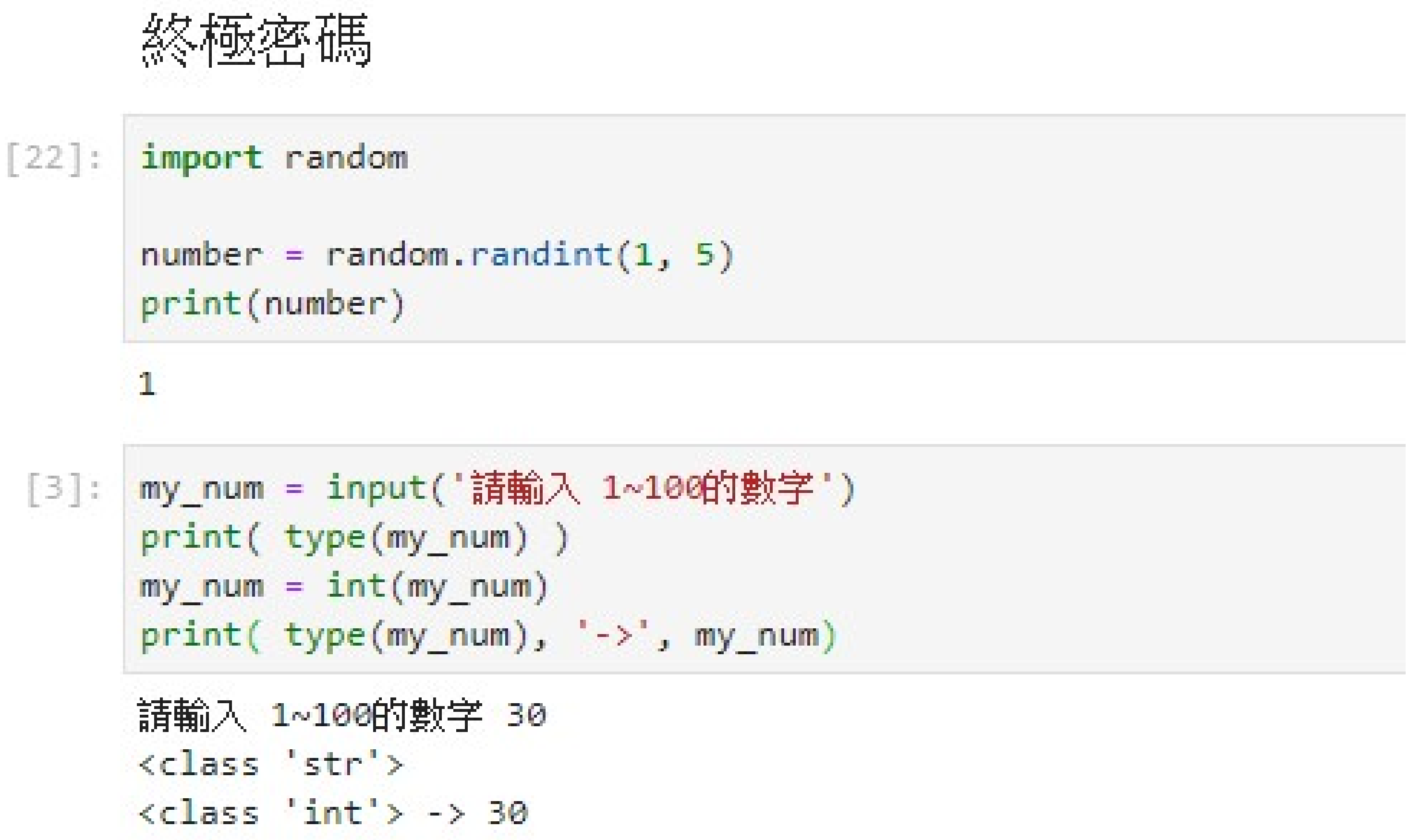
```
[13]: my_dict['Strawberry'] = 10
      my_dict['Apple'] += 2
```

```
[14]: my_dict
```

```
[14]: {'Apple': 5, 'Pear': 5, 'Orange': 2, 'Strawberry': 10}
```

練習題

- 以下為提示資料，請測試後，完成一個終極密碼的遊戲



終極密碼

```
[22]: import random

      number = random.randint(1, 5)
      print(number)
```

```
1
```

```
[3]: my_num = input('請輸入 1~100的數字')
     print( type(my_num) )
     my_num = int(my_num)
     print( type(my_num), '->', my_num)
```

```
請輸入 1~100的數字 30
<class 'str'>
<class 'int'> -> 30
```