

# SC2

## 1. Definitions and Short Answers

1. What is a **comment** in a program and what is its purpose?



註解可以讓人了解這支程式的功能

2. What is an **operator**? Give some examples of **arithmetic operators** in Python.



運算子。 +, -, \*, /, //, \*\*

3. What is a **comparison operator**? What are possible results of a comparison?



>, <, ==, !=

運算結果為True或False

4. What is a **logical operator**? What are possible results of a logical operation?



&, |, ^, not

運算結果為True或False

5. What is 20 in **hexadecimal** representation? in **octal** representation?



0x14, 0o24

6. Why does Python support two **division operators**? What is their difference?



/, //

前者是浮點數除法，後者是整數除法

7. What is the difference between **'12'** and **12** in Python?



前者是string，後者是數字

8. What is the difference between **x = 3** and **x == 3** in Python?



前者是assignment，後者是判斷式(結果為True or False)

9. Assuming the variable **y** has been assigned the integer value of 4, which of the following are legal in Python and what do they do? which are illegal in Python?

- **y = 4** 合法
- **4 = y** 非法
- **y == 4** 合法
- **4 == y** 合法
- **'y' = y** 非法
- **'y' == '4'** 合法
- **'4' = y** 非法

10. Assume variable **x** has integer value 3, and variable **y** has integer value of 4. What is the result of the following operator expressions, if they are legal in Python? Which of the following are not legal?

- **x \* y** 合法
- **'x' \* y** 合法
- **x \* 'y'** 合法
- **'x' \* 'y'** 非法

- $x + y$  合法
- $'x' + 'y'$  合法
- $'x' + y$  非法
- $x + 'y'$  非法

11. What is the data type of `['Sun','Mon','Tue','Wed','Thu','Fri','Sat']`?



list

12. if `L = ['Sun','Mon','Tue','Wed','Thu','Fri','Sat']`, then what are the **values** of the following **expressions** if they are legal Python? Which are illegal?

- `L[3]`



`'Wed'`

- `L[1:5]`



`['Mon', 'Tue', 'Wed', 'Thu']`

- `L[5:1]`



`[]`

- `L(2:3)`



`'list' object is not callable`

- `L[1,2,3]`



list indices must be integers or slices, not tuple

- `L{3}`



invalid syntax

- `L[1-5]`



`'Wed'`

- `L['3']`



list indices must be integers or slices, not str

13. Assume `T = ('Sun','Mon','Tue','Wed','Thu','Fri','Sat')`, which of the following are allowed in Python, and what are their output or effect? Which are not allowed, for what reasons?

- `print(T[3])`



`Wed`

- `print(T(3))`



`TypeError: 'tuple' object is not callable`

- `print(T{3})`



SyntaxError: invalid syntax

- `T[3] = 'WED'`



TypeError: 'tuple' object does not support item assignment

- `T[3] == 'WED'`



True

- `print(T[3:5])`



`('Wed', 'Thu')`

- `print(T[3, 5])`



TypeError: tuple indices must be integers or slices, not tuple

- `print(T['3'])`



TypeError: tuple indices must be integers or slices, not str

14. Assume `S = {'Sun','Mon','Tue','Wed','Thu','Fri','Sat'}`, which of the following are allowed in Python, and what are its output or effect? Which are not allowed, for what reasons?

- `print(S[3])`



TypeError: 'set' object is not subscriptable

- `print(S(3))`



TypeError: 'set' object is not callable

- `print(S{3})`



SyntaxError: invalid syntax

- `S[3] = 'WED'`



TypeError: 'set' object does not support item assignment

- `S[3] == 'WED'`



TypeError: 'set' object is not subscriptable

- `print(S[3:5])`



TypeError: 'set' object is not subscriptable

- `print(S[3, 5])`



TypeError: 'set' object is not subscriptable

- `print(S['3'])`



TypeError: 'set' object is not subscriptable

15. Assume  $D = \{\text{'Sun':0, 'Mon':1, 'Tue':2, 'Wed':3, 'Thu':4, 'Fri':5, 'Sat':6}\}$ , which of the following are legal in Python, and what are their values?

- $D[3]$  非法
- $D[\text{'Thu'}]$  4
- $D[0:3]$  非法
- $D[2, 6]$  非法
- $D\{\text{'Sun'}\}$  非法
- $D(0)$  非法
- $D\{3\}$  非法
- $D(\text{'Sun'})$  非法

16. What is the value of  $\{2, 3, 4\} \mid \{3, 4, 5\}$ ?



$\{2, 3, 4, 5\}$

17. What is the value of  $\{2, 3, 4\} \& \{3, 4, 5\}$ ?



$\{3, 4\}$

18. Suppose you have the following sequence of Python statements:

$x = 3$   $y = 2$

**if**  $x > y$ :

`print("x is bigger than y")`

**elif**  $x == y$ :

`print("x and y are the same")`

**else:**

```
    print("x is smaller than y")
```

What is printed?



x is bigger than y

19. What is wrong with the following code, which is supposed to compute the total of a list of numbers?

```
L = [3, 2, 6, 5]
```

```
for i in L:
```

```
    total = total + i
```

```
print(total)
```

How can it be fixed?



要初始化, total = 0

20. What is the difference between

```
x = 0
```

```
while x < 100:
```

```
    x = x + 1
```

and

```
x = 0
```

```
if x < 100:
```

```
    x = x + 1
```

?





while會重複執行，直到條件不滿足。  
if只判斷那一次，執行至多一次。

21. What is an example of a **function** in Python? How do you **call** a function? What is a **parameter**?
22. What is an example of calling a (built-in) function that **returns a value**?
  - does input() return a value? YES
  - does print() return a value? NO
  - what other built-in functions do you know that returns a value? open() list()
23. Python supports two kinds of **loops**. What are they?



for-loop, while-loop

24. What is a **suite**? What is the pronunciation of "suite"?



function body

25. What does **import** math do? How do you call the cos function (cosine) defined in the math module in Python?



載入math這個內建的模組  
math.cos(0)

26. To read a file, it is common to see fh = open('filename'). What kind of data is fh called? Give an example of using fh for accessing (e.g., reading or writing) a file.



要看filename的格式  
fh.readlines()

27. if `s = 'hello'`, Python supports two styles of “calls” (or “invocation”):

- `len(s)` is an example of a **function call**
- `upper()` is another form of call. What kind of call is it?



method call

28. How are **class** and **instance** related to each other?



`instance(實例化)`: 建出一個實體的class

29. Why is it incorrect to split the statement `f = a + b * 2 + c / 2 - 4 * d` onto two separate lines as the following

```
f = a + b * 2 + c / 2
```

```
- 4 * d ?
```

How can it be fixed so Python will accept it?



改成

```
f = a + b * 2 + c / 2 \
```

```
- 4 * d
```

30. If you want to **swap** the values of two variables `x` and `y`, why can't you just do `x = y` `y = x` ? Give two different ways you can swap their values correctly in Python.



1. `(x, y) = (y, x)`

2.

```
temp = y
```

```
y = x
```

```
x = temp
```

31. What is a **keyword** in Python? Give some example keywords in Python.



['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

32. Which of the following are legal and illegal **identifiers** in Python?

- myname
- my\_name
- \_myname
- MyName
- myname\_
- my-name 非法
- my11name
- myname11
- 11myName
- my\_11Name
- \_11myName
- @myname 非法
- my@name 非法
- myname@ 非法
- in 非法
- out
- \_in
- \_out
- IN

- OUT
- and 非法
- or 非法
- but
- function
- integer
- number
- class 非法
- instance
- global 非法
- local 非法
- you+me 非法
- I\_love\_\$\$ 非法

33. What is an example of a **snake-case** identifier? a **camel-case** identifier?



snake-case: hello\_world

camel-case: goodByeWorld