

選擇題

1. 在函式外定義的變數稱為？

- a. 區域變數
- b. 唯一變數
- c. 全域變數
- d. 以上皆非

2. 以下哪個函式標題是正確的？

- a. `def Demo(P, Q = 10)`
- b. `def Demo(P = 10, Q = 20)`
- c. `def Demo(P = 10, Q)`
- d. 選項 a 和選項 b 皆是

3. 以下程式的輸出為何？

```
x = 10
def f():
    x = x + 10
    print(x)
f()
```

- a. 20
- b. 10
- c. `UnboundLocalError`
- d. 以上皆非

4. 以下程式的輸出為何？

```
def Func_A(P = 10, Q = 20):
    P = P + Q
    Q = Q + 1
    print(P, Q)
Func_A(Q = 20, P = 10)
```

- a. Error: P and Q are not defined
- b. 20 10
- c. 10 20
- d. 30 21

5. 以下程式的輸出為何？

```
def test():  
    x = 10
```

```
x = 11  
test()  
print(x)
```

- a. 10
- b. 11
- c. 無意義的值
- d. 以上皆非

6. 如果函式不回傳任何值，那麼預設情況下函式會回傳哪種型態的值？

- a. int
- b. double
- c. str
- d. None

7. 以下程式的輸出為何？

```
def test():  
    global x  
    x = 'A'
```

```
x = 'Z'  
test()  
print(x)
```

- a. Z
- b. A
- c. 無意義的值
- d. 以上皆非

8. 以下程式的輸出為何？

```
def test(x):  
    x = 200
```

```
x = 100  
test(x)  
print(x)
```

- a. 100
- b. 無意義的值
- c. 200
- d. 以上皆非

9. 以下程式的輸出為何？

```
def test(x):  
    p = 90
```

```
p = 50  
print(test(p))
```

- a. 90
- b. 50
- c. Error
- d. None

10. 以下程式的輸出為何？

```
def evaluate_expression_1(Z):  
    Z = Z + 5  
    def evaluate_expression_2(Z):  
        print('Hello')  
        return Z  
    return Z  
value = 10  
print(evaluate_expression_1(value))
```

- a. Hello 10
- b. 10
- c. 15 Hello
- d. 15

11. 以下程式的輸出為何？

```
def evaluate_expression_1():  
    global x  
    x = x - 5  
  
    def evaluate_expression_2():  
        global x  
        return x + 3  
  
    return evaluate_expression_2()  
  
x = 10  
print(evaluate_expression_1())
```

- a. 5
- b. 8
- c. 10
- d. 13

12. 以下程式的輸出為何?

```
def perform_multiplication(Num1, Num2):  
    Num2 = Num1 * Num2  
    return Num1, Num2
```

```
Num2, Num1 = perform_multiplication(5, 4)  
print(Num , Num2)
```

- a. 5 4
- b. 5 20
- c. 20 5
- d. 4 5

13. 以下程式的輸出為何?

```
def Display(Designation , Salary):  
    print(Designation , Salary)  
Display('Manager', 25000)  
Display(300000, 'Programmer')
```

- a. Error:Type Mismatch
- b. Manager 25000
300000 Programmer
- c. 300000 Programmer
Manager 25000
- d. 以上皆非

14. 以下程式的輸出為何?

```
def func1():  
    print("Outer function")  
    def func2():  
        print("Inner function")  
func2()
```

- a. Outer function
- b. Inner function
- c. Outer function, Inner function
- d. NameError

15. 以下程式的輸出為何？

```
def func(a, b = 40, c = 50):  
    print(a, b, c)
```

```
func(100, "Welcome")
```

a. 100 Welcome 50

b. 100 40 50

c. Error

d. Welcome 40 50

16. 以下程式的輸出為何？

```
def compute(arg):  
    arg += '3'  
    arg *= 3  
    return arg
```

```
print(compute( 'abc' ))
```

a. abc3

b. abcabcab

c. abc3abc3abc3

d. Error

17. 以下程式的輸出為何？

```
x = 100  
y = 200  
def demo():  
    global y  
    x = 450  
    y = 900  
demo()  
print(x, y)
```

a. 100 200

b. 450 900

c. 100 900

d. 450 200

18. 以下程式的輸出為何？

```
def arg_demo(p = 5, *args):  
    print(args)  
arg_demo(5, 6, 7, 8)
```

- a. (5, 6, 7, 8)
- b. Error
- c. (6, 7, 8)
- d. [6, 7, 8]

19. 以下程式的輸出為何？

```
S1 = "Welcome to JAVA Programming"  
S2 = S1.replace("JAVA","Python")  
print(S2)
```

- a. Welcome to JAVA Programming
- b. Welcome to Python Programming
- c. Welcome to JAVA Python Programming
- d. 以上皆非

20. 以下程式的輸出為何？

```
Str1 = "Hello"  
Str2 = Str1[:-1]  
print(Str2)
```

- a. ello
- b. Hello
- c. el
- d. Hell

21. 以下程式的輸出為何？

```
Str1 = "The Sum of {0:b} and {1:b} is {2:b}".format(2, 2, 4)  
print(Str1)
```

- a. The Sum of 10 and 10 is 0100
- b. The Sum of 2 and 2 is 100
- c. The Sum of 10 and 10 is 100
- d. The Sum of 2 and 2 is 4

22. 以下程式的輸出為何？

```
Str1 = "ABBCCDEEBBFFERBBJJUIBB"  
print(Str1.count("BB"), end=' ')  
print(Str1.count("BB", 1), end=' ')  
print(Str1.count("BB", 2), end=' ')  
print(Str1.count("BB", 3), end=' ')
```

a. 4 4 3 3

b. 4 3 4 3

c. 3 4 3 4

d. 4 4 4 3

23. 以下程式的輸出為何？

```
Str1 = "Python Programming"  
Str1[0] = "J"  
print(Str1)
```

a. Jython Programming

b. Jython

c. Jython Jrogramming

d. Error

24. 以下程式的輸出為何？

```
S = "Programming"  
for char in S:  
    print(char, end="")
```

a. Programming

b. P r o g r a m m i n g

c. Error

d. 以上皆非

25. 以下程式的輸出為何？

```
S = "ILOVEWORLD"
for ch in range(0, len(S), 3):
    print(S[ch], end=" ")
```

a. I V O D

b. I O W L

c. I V W L

d. I L O V

26. 以下程式的輸出為何？

```
def countbc(word):
    print(word)
    count = 0
    for bc in word:
        if(bc == 'bc'):
            count = count + 1
    return count
print("Number of 'bc' = ", countbc("abcbabcaaa"))
```

a. 0

b. 10

c. 2

d. 1

27. 如何用以下的列表將' UK' 提取出來？

```
Countries = [ 'India' , ' USA' , ' UK' ]
```

a. Countries[2]

b. Countries[-1:]

c. 選項 a 和選項 b 都是

d. 只有選項 a

28. 以下程式的輸出為何？

```
a = '\t\t\tPython\n\n'
print(a.strip())
```

- a. Python\n
- b. Python\n\n
- c. Python
- d. \t\tPython

29. 以下程式的輸出為何？

```
S = 'ABC'
n = 1
for ch in S:
    print(ch * n)
    n = n + 1
```

- a. ABC
- b. A
BB
CCC
- c. 1
22
333
- d. A BB CCC

30. 以下哪個選項並非有效操作？

- a. "Hello" + "World"
- b. "Hello" + 123
- c. '\$' * 5
- d. 以上皆非

31. 以下程式的輸出為何？

```
str1 = 'I love Python programming. Python is very easy. '  
print(str1.find("Python"), ' ', str1.rfind("Python"))
```

a. True, True

b. 7, 27

c. 6, 26

d. 以上皆非

32. 以下敘述式的輸出何者為 True？

a. "Python".isalpha()

b. "".isalpha()

c. "123".isalpha()

d. "1a2b3c".isalpha()

程式題(共五題)

1.

隨堂練習 (使用巢狀for 迴圈)

- 撰寫一個程式，使用星星排列並顯示出以下的圖案：

```
*  
* *  
* * *  
* * * *  
* * * * *
```

2.

Assignment (使用巢狀for 迴圈)

- 撰寫一個程式，使用星星排列並顯示出以下的圖案：

```
      *
    * * *
  * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
```

3.

Assignment (使用巢狀for 迴圈)

- 撰寫一個程式，使用數字排列並顯示出以下的圖案：

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

4.

Assignment

- 撰寫一個程式，使用數字排列並顯示出以下的圖案：

```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

5.

Complementing DNA

Please take the following sequence and replace A with T, T with A, C with G, and G with C.

```
my_dna =
"ACTGATCGATTACGTATAGTATTTGCTATCATACATATATATCGATGCGTT
CAT"
```

Hint:

(1) Use temporary alphabet of four letters.

or

(2) Taking advantage of the fact that the replace method is case-sensitive. You can make all the replaced bases in lower cases.