

1. Python설치시 기본 탑재된 Tool을 모두 고르시오

IDEL

PyCharm

VSCode

Colab

2. 다음 결과는?

a = “안녕하세요”

print("a")

a

3. 결과는?

a = 30

b = 20

c = 10

if(a<b):

 a = 5

if(a<c):

 print("1")

else:

 print("2")

2

4. 빈칸에 들어갈 코드는?

```
score = {"a":2,"b":3}
```

```
print(?????) score.keys()
```

결과 : dict_keys(['a', 'b'])

5. for문을 완성하세요

```
fruits = ['사과', '바나나', '딸기', '포도']
```

```
??????
```

```
for i in fruits:
```

```
    print(i)
```

출력:

사과

바나나

딸기

포도

6. 두 수를 입력받아 곱한 값을 반환하는 multiply() 함수를 만드세요.

```
?????def multiply(a,b):
```

```
    return a*b
```

```
a = 10
```

```
b = 3
```

```
Print(multiply(a,b))
```

결과 : 30

7. 결과예측하시오

```
a = 10
```

```
b = 20
```

```
def s():
```

```
    global a
```

```
    b = 3
```

```
    print(a,b)
```

```
    a = 5
```

```
    return
```

```
s()
```

```
print(a,b)
```

10 3

5 20

8. 결과는?

a = [10,20,30,40]

```
def swap(lista):
```

```
    lista[1] = lista[0]
```

```
    lista[0] = lista[1]
```

```
    lista[0],lista[2] = lista[2],lista[0]
```

```
    return lista
```

```
resulta = swap(a)
```

```
print(resulta)
```

[30, 10, 10, 40]

9. 결과는?

```
class Car:
```

```
    wheels = 4
```

```
    def __init__(self, brand):
```

```
        self.brand = brand
```

```
car1 = Car("현대")
```

```
car2 = Car("기아")
```

```
print(car1.brand, car1.wheels)
```

```
print(car2.brand, car2.wheels)
```

```
Car.wheels = 6
```

```
print(car1.wheels, car2.wheels)
```

현대 4

기아 4

6 6

10. 결과는?

```
class Student:
```

```
    #__slots__ = ['name', 'age']
```

```
    def __init__(self, name, age):
```

```
        self.name = name
```

```
        self.age = age
```

```
s = Student("홍길동", 20)
```

```
s.grade = 90
```

```
print(s.name,s.age)
```

11.빈칸에 공통으로 들어갈 단어는?

```
import tkinter
```

```
root = tkinter.Tk()
```

```
button1 = tkinter.Button(root, text="A")
```

```
button2 = tkinter.Button(root, text="B")
```

```
button3 = tkinter.Button(root, text="C")
```

```
button4 = tkinter.Button(root, text="D")
```

```
button1._____(row=0, column=0)
```

```
button2._____(row=0, column=1)
```

```
button3._____(row=1, column=0)
```

```
button4._____(row=1, column=1)
```

```
root.mainloop()
```

결과:

A B

C D

grid

12.

1. 파일 열기 모드 w와 a의 차이점은?
2. with 문을 사용하여 파일을 처리하는 이유는?

13. 빈칸을 채우세요

```
text = "Hello World Python"
```

```
words = text.____() split
```

```
print(words)
```

```
fruits = ["apple", "banana", "cherry"]
```

```
com = "-".____(fruits) join
```

```
print(com)
```

결과

```
['Hello', 'World', 'Python']
```

```
apple-banana-cherry
```


14. 결과는?

```
import re

text = "The rain in Spain"

result = re.findall(r"^\bSWw+", text)

print(result)
```

Spain

15. 다음 중 이터러블(Iterable)과 이터레이터(Iterator)에 대한 설명으로 옳은 것은?

1. 리스트는 이터레이터이므로 next() 함수를 직접 호출할 수 있다.
2. 제너레이터는 이터레이터를 생성하는 함수이며, yield 키워드를 사용한다.
3. 이터러블 객체는 반드시 __next__ 메서드만 구현하면 된다.
4. iter() 함수는 이터레이터를 이터러블로 변환한다.

16. 결과는?

```
from collections import deque

dq = deque([[1,2]])

dq.append(1)

dq.append(2)

print(dq.pop())
```

```
print(dq.popleft())
```

16. 결과는?

```
graph = {'A': ['B', 'C'], 'B': ['D', 'E'], 'C': ['F'], 'D': [], 'E': ['F'], 'F': []}
```

```
def dfs(graph, start):
```

```
    visited = []
```

```
    stack = [start]
```

```
    while stack:
```

```
        node = stack.pop()
```

```
        if node not in visited:
```

```
            visited.append(node)
```

```
            stack.extend(reversed(graph[node]))
```

```
    return visited
```

```
print(dfs(graph,'A'))
```

17. 빈칸 채우기

```
def bfs(graph, start):
```

```
    visited = []
```

```
    stack = ?????
```

```
    while stack:
```

```
        ?????
```

```
        if node not in visited:
```

```
            visited.append(node)
```

```
            stack.extend(reversed(graph[node]))
```

```
    return visited
```

```
print(bfs(graph,'A'))
```

출력 : ['A', 'C', 'B', 'F', 'E', 'D']

18. 옳은 것을 모두 고르세요

1. turtle.Screen().bgpic("image.gif")는 배경 이미지 설정이 가능하며, GIF 포맷만 지원한다.

2. `turtle.getshapes()`는 'arrow', 'turtle', 'circle', 'square', 'triangle', 'classic'을 기본으로 반환한다.
3. `turtle.textinput("입력", "이름")`은 그래픽 창에 입력 대화상자를 띄운다.
4. `turtle.write()`의 font 매개변수는 튜플 형태로 (폰트명, 크기, 스타일)을 지정해야 한다.
5. `turtle.ondrag(turtle.goto)`를 사용하면 마우스로 거북이를 드래그해 선을 그릴 수 있다.

18. 출력 도형은 무슨 모양일까?

```
import turtle
```

```
t = turtle.Turtle()
```

```
for i in range(6):
```

```
    t.forward(100)
```

```
    t.right(60)
```

```
turtle.exitonclick()
```

19.출력은?

```
import pandas as pd
```

```
import numpy as np
```

```
data = {
```

```
    'ID': [1, 2, 3, 4, 5],
```

```
    'Department': ['HR', 'Engineering', 'HR', 'Engineering', 'Finance'],
```

```
    'Salary': [70000, 120000, 80000, 110000, 90000]
```

```
}
```

```
df = pd.DataFrame(data)
```

```
a = df.groupby('Department')
```

```
print(a.mean()["ID"].sum()) 10
```

```
print(a.max()["Salary"].min()) 80000
```

```
dep id sal
```

```
hr 1,3 70000, 80000
```

```
en 2,4 120000 110000
```

```
fi 5 90000
```

20. 각각 주용도에 맞게 3개의 짝을 지으세요

- statsmodels 2 - - ① 머신러닝 모델 구현
- scikit-learn 1 - - ② 통계 분석
- scipy 3 - - ③ 과학 계산