

04. 구조와 모듈

총점 12/17 ?

내용이 많은 만큼 꼼꼼히 공부해봅시다 > <

✗ 활성화 레코드에는 함수의 정보(반환값, 매개변수, 지역변수, 반환값, 반환 주소 등)이 기록되는데, 활성화 레코드가 처리되는 순서를 적으시오. (답안 예시: 1234) *

0/1

- 1) 함수로 건너뛰다.
- 2) 반환 주소를 스택에 저장한다.
- 3) 함수의 실제 매개변수를 스택에 저장한다.
- 4) 스택의 최상위 인덱스를 함수의 지역 변수에 필요한 총량만큼 늘린다.

3421

✗

정답

3241

✓ 파이썬이 패키지(package)로 취급하는 것은? *

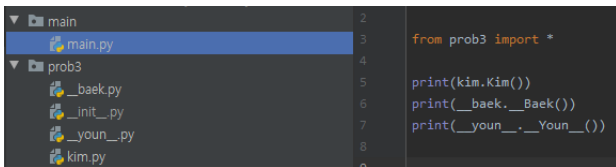
1/1

- ☐ 모듈
- ☐ __init__.py
- ☐ 모듈이 있는 디렉터리
- ☒ 모듈과 __init__.py가 있는 디렉터리
- ☐ 모르겠다.

✓

✓ 다음 main.py가 동작하기 위해 __init__.py에 들어갈 내용은? *

1/1



- ☐ 아무것도 적지 않아도 가능하다.
- ☐ __init__ = ""
- ☐ __init__ = ["kim", "__baek", "__youn__"]
- ☐ __all__ = ""
- ☒ __all__ = ["kim", "__baek", "__youn__"]
- ☐ __all__ = ["kim.py", "__baek.py", "__youn__.py"]
- ☐ 모르겠다.

✓

✓ 바이트 컴파일 코드(byte-compiled code)에 대한 설명으로 옳지 않은 것은? *

1/1

- ☐ 표준 모듈을 많이 사용하는 프로그램의 시작 시간(로딩 시간)을 줄이기 위한 것이다.
- ☐ python 3.5부터는 .pyo 대신 .pyc를 사용한다.
- ☒ 용량이 작고 리버스 엔지니어링이 빨라 스크립트 공유에 용이하다.
- ☐ 모르겠다.

✓

✓ 다음 스크립트의 실행 결과로 옳은 것은? *

1/1

```
# main/1_sys_example.py
import sys

def main():
    for idx, arg in enumerate(sys.argv[1:]):
        text = "(" + str(idx) + ", " + arg + ")"
        print(text, end=" ")
    if name == "main":
        main()

# on terminal
# $ python main/1_sys_example.py a b c d e
```

- ☐ (0, b) (1, c) (2, d) (3, e)
- ☒ (0, a) (1, b) (2, c) (3, d) (4, e) ✓
- ☐ (0, 1_sys_example.py) (1, a) (2, b) (3, c) (4, d) (5, e)
- ☐ (0, main/1_sys_example.py) (1, a) (2, b) (3, c) (4, d) (5, e)
- ☐ 모르겠다.

✓ 다음 중 파이썬에서 '거짓'을 나타내는 것이 아닌것은? *

1/1

- ☒ "False" ✓
- ☐ 0
- ☐ None
- ☐ *
- ☐ list()
- ☐ tuple()
- ☐ set()
- ☐ dict()
- ☐ 모두 거짓이다.
- ☐ 모르겠다.

✓ 다음 중 "True"가 출력되는 것은? *

1/1

```
a = [1, 2, 3]
def f(a):
    while a:
        yield a.pop()

func = f(a)
if func == func.__iter__():
    print("True")
else:
    print("False")

if func.__next__() == 1:
    print("True")
else:
    print("False")

if next(func) == 3:
    print("True")
else:
    print("False")

if func.__iter__().__next__() == 2:
    print("True")
else:
    print("False")
```

- ☒ func == func.__iter__()
- ☐ func.__next__() == 1
- ☐ next(func) == 3
- ☐ func.__iter().__next__() == 2
- ☐ 모르겠다.

✓

✓ 다음 함수의 실행 결과로 옳은 것은? *

1/1

```
def f(x): return x % 2 != 0 or x % 3 != 0
print(list(filter(f, range(2, 25))))
```

- ☐ [6, 12, 18, 24]
- ☐ [5, 7, 11, 13, 17, 19, 23]
- ☐ [2, 3, 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24]
- ☒ [2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23]
- ☐ 모르겠다.

✓

✓ 네모 안에 공통적으로 들어갈 메서드는? *

1/1

```
import collections

minus_one_dict = collections.????(lambda: -1)
point_zero_dict = collections.????(lambda: (0, 0))
message_dict = collections.????(lambda: "No message")
```

- ☐ Counter
- ☒ defaultdict
- ☐ OrderedDict
- ☐ deque
- ☐ 모르겠다.

✓

✓ 다음 코드의 출력 결과로 옳은 것은? *

1/1

```
a = [1, 2, 3, 4, 5]
b = ['a', 'b', 'c', 'd']
print(list(zip(a, b)))
```

- ☐ [1, 2, 3, 4, 5, 'a', 'b', 'c', 'd']
- ☐ [(1, 2, 3, 4, 5), ('a', 'b', 'c', 'd')]
- ☒ [(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd')]
- ☐ [(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd'), (5, ')]
- ☐ 모르겠다.

✓

✓ 다음은 open() 메서드의 모드에 대한 설명이다. 옳지 않은 것은? *

1/1

- ☐ r : 읽기 모드
- ☐ w : 쓰기 모드(동명 파일이 이미 있다면, 그 파일을 지운 후 내용을 새로 쓴다.)
- ☐ a : 추가 모드(동명 파일이 이미 있다면, 그 파일 끝에 내용을 추가한다.)
- ☒ rw : 읽기와 쓰기 모드
- ☐ t : 텍스트 모드
- ☐ b : 바이너리 모드
- ☐ 모르겠다.

✓

✓ 다음 네모에 알맞은 메서드는? *

1/1

```
# test.txt
안녕하세요..
저는
김인중 입니다..

# test.py
with open("test.txt", encoding="utf8") as f:
    text = f.
    print(text)

# 출력 결과
안녕하세요..
저는
김인중 입니다.]
```

- ☒ read()
- ☐ readline()
- ☐ readable()
- ☐ readlines()
- ☐ 모르겠다.

✓

✓ 다음 중 파이썬 객체를 가져와서 문자열 표현으로 변환(직렬화)하는 모듈은? *

1/1

- ☐ shutil
- ☒ pickle
- ☐ struct
- ☐ 모르겠다.

✓

✕ 다음 코드의 출력 결과로 옳은 것은? *

0/1

```
# test.txt
안녕하세요..
저는
김인중 입니다..

# test.py
import string, sys

try:
    f = open('test.txt', encoding='utf8')
    s = f.readline()
    i = int(string.strip(s))
    print("에러 없음")

except IOError as err: # No such file or directory
    print(err)
except ValueError as err2: # invalid literal for int() with base 10: '안녕하세요.\n'
    print("데이터를 숫자로 변환할 수 없습니다.")
except AttributeError as err3: # module 'string' has no attribute 'strip'
    print("예외 발생: {0}".format(sys.exc_info()[0]))
```

- ☐ [Errno 2] No such file or directory: 'test.txt'
- ☒ 데이터를 숫자로 변환할 수 없습니다.
- ☐ 예외 발생: <class 'AttributeError'>
- ☐ 에러 없음
- ☐ 모르겠다.

정답

- ☒ 예외 발생: <class 'AttributeError'>

✕ 다음 코드는 파일을 읽어서 모든 빈 줄을 제거하고 저장하는 코드이다. 빈 칸을 채우시 .../3
오. *

```
# 파일을 읽어서 모든 빈 줄을 제거하는 코드
import sys

def read_data(filename):
    lines = []
    with open(filename) as fh:
        for line in fh:
            if line.strip():
                lines.append(line)
    return lines

def write_data(lines, filename):
    fh = None
    with open(filename, "w") as fh:
        for line in lines:
            fh.write(line)

def remove_black_lines():
    if len(sys.argv) < 2:
        print("Usage: python {0} [file ...]".format(sys.argv[0]))

    if __name__ == "__main__":
        remove_black_lines()
```

의견 보내기

```
for filename in sys.argv[1:]:
    lines = read_data(filename)
    if lines:
        write_data(lines, filename)
```

이 콘텐츠는 Google이 만들거나 승인하지 않았습니다. - [서비스 약관](#) - [개인정보처리방침](#)

