1. 다음 빈 칸에 공통으로 들어갈 정답으로 올바른 것은?

( ) 파이썬은 매일 ( ) 이라는 짧은 시간을 투자해서
누구나 쉽고 재미있게 파이썬을 배울 수 있도록 하는 컨텐츠이다.

① 1분 ② 2분 ③ 3분 ④ 4분

1. 다음 빈 칸에 공통으로 들어갈 정답으로 올바른 것은?

( ) 파이썬은 매일 ( ) 이라는 짧은 시간을 투자해서
누구나 쉽고 재미있게 파이썬을 배울 수 있도록 하는 컨텐츠이다.

① 1분 ② 2분 ③ 3분 ④ 4분

정답 ①
매일 1분 정도는 괜찮죠? 다음 시간부터 제대로 시작해보아요 ^^

2. Visual Studio Code 에서 파이썬 개발을 위한 환경 설정 중 가장 필요 없는 단계는?

① 공식 홈페이지에서 파이썬 설치 파일을 다운로드 및 실행한다.

② Visual Studio Code 를 설치한다.

③ Visual Studio Code 에서 Python extension 을 설치한다.

④ 설치 과정에 문제가 발생하지 않도록 기도한다.

2. Visual Studio Code 에서 파이썬 개발을 위한 환경 설정 중 가장 필요 없는 단계는?

① 공식 홈페이지에서 파이썬 설치 파일을 다운로드 및 실행한다.

② Visual Studio Code 를 설치한다.

③ Visual Studio Code 에서 Python extension 을 설치한다.

④ 설치 과정에 문제가 발생하지 않도록 기도한다.

정답 ④
기도가 필요할 때도 있지만 아직은 아니에요

3. 다음 중 자료형의 종류와 예시가 올바르게 짝지어지지 않은 것은?

① 숫자 자료형 (예 : 1, 0, -500)

② 문자 자료형 (예 : '작은 따옴표 문자', ''큰 따옴표 문자'')

③ 숫자 자료형 (예 : '3.1982', '10.0')

④ 불리안 자료형 (예 : True, False)

3. 다음 중 자료형의 종류와 예시가 올바르게 짝지어지지 않은 것은?

① 숫자 자료형 (예 : 1, 0, -500)

② 문자 자료형 (예 : '작은 따옴표 문자', ''큰 따옴표 문자'')

③ 숫자 자료형 (예 : '3.1982', '10.0')

④ 불리안 자료형 (예 : True, False)

정답 ③

숫자 자료형이 되기 위해서는 작은 따옴표 없이 3.1982 나 10.0 으로 해야 돼요

4. 다음 중 ★ 에 들어갈 수 있는 변수 선언 방법으로 올바른 것은?

```
(      ★      )
print(age) # 실행 결과 : 7
```

① 7 = age　　② '7' = age　　③ age = 7　　④ age : 7

4. 다음 중 ★ 에 들어갈 수 있는 변수 선언 방법으로 올바른 것은?

```
(      ★      )
print(age) # 실행 결과 : 7
```

① 7 = age　② '7' = age　③ age = 7　④ age : 7

정답 ③

변수 선언은 등호(=) 기준으로 왼쪽에 변수 이름을, 오른쪽에 값을 입력하면 돼요

5. 다음 중 변수 이름으로 사용할 수 없는 것은?

① name_

② _name

③ Name1

④ 2name

5. 다음 중 변수 이름으로 사용할 수 없는 것은?

① name_

② _name

③ Name1

④ 2name

정답 ④

변수의 이름은 숫자로 시작할 수 없어요

6. 다음 형 변환 중 잘못된 것은?

① int('3.14')

② str(3.14)

③ int('3')

④ str(3)

6. 다음 형 변환 중 잘못된 것은?

① int('3.14')

② str(3.14)

③ int('3')

④ str(3)

정답 ①

'3.14'는 먼저 실수로 변환한 뒤에 정수로 변환해야 해요 > int(float('3.14'))

7. 다음 문장을 파이썬 코드로 표현하면?

A 는 B 보다 크거나 같고, C 는 D 보다 작거나 같다

① A >= B and C <= D

② A => B and C =< D

③ A >= B or C <= D

④ A => B or C =< D

7. 다음 문장을 파이썬 코드로 표현하면?

A 는 B 보다 크거나 같고, C 는 D 보다 작거나 같다

① A >= B and C <= D

② A => B and C =< D

③ A >= B or C <= D

④ A => B or C =< D

정답 ①

비교 연산 시에는 부등호를 먼저 써야 해요. and 와 or 의 차이는 아시죠? ^^

8. 다음 코드의 실행 결과로 올바른 것은?

```
print(3 > 5)
print(bool(None))
print(bool('False'))
```

① False
True
True

② False
False
False

③ False
False
True

④ True
True
True

8. 다음 코드의 실행 결과로 올바른 것은?

```
print(3 > 5)
print(bool(None))
print(bool('False'))
```

① False
True
True

② False
False
False

③ False
False
True

④ True
True
True

정답 ③ 3 은 5 보다 크지 않으니 False, None 은 '없는 값'이니 False,
'False' 는 빈 문자열이 아닌 'False' 라는 값을 가지기 때문에 True 랍니다

9. 다음 코드의 실행 결과로 올바른 것은?

```
print('파이썬은')
#print('처음에는')
print('쉬워요')
```

① 파이썬은 ② 처음에는 ③ 파이썬은 쉬워요 ④ 파이썬은 처음에는 쉬워요

9. 다음 코드의 실행 결과로 올바른 것은?

```
print('파이썬은')
#print('처음에는')
print('쉬워요')
```

① 파이썬은

② 처음에는

③ 파이썬은 쉬워요

④ 파이썬은 처음에는 쉬워요

정답 ③

두 번째 줄이 #으로 주석처리 되었으므로 '처음에는' 부분은 출력되지 않아요

10. 다음 실행 결과를 얻기 위해서 ★ 에 들어갈 수 없는 코드는?

```
fruit = 'apple'
print(        ★        )
# 실행 결과 : apple
```

① fruit[:] ② fruit[0:] ③ fruit[:5] ④ fruit[:-1]

10. 다음 실행 결과를 얻기 위해서 ★ 에 들어갈 수 없는 코드는?

```
fruit = 'apple'
print(        ★        )
# 실행 결과 : apple
```

① fruit[:]　② fruit[0:]　③ fruit[:5]　④ fruit[:-1]

정답 ④ -1 은 마지막 위치의 인덱스 값이며 슬라이싱은 콜론(:) 뒤에 적힌 위치의 '직전까지' 출력하므로 실행결과는 apple 이 아닌 appl 이랍니다

11. regret 변수에 저장된 문자열의 길이를 확인하기 위한 방법으로 올바른 것은?

```
regret = '아니 1분만에 파이썬을 어떻게 배우냐고'
```

① len(regret)
② regret.len()
③ sizeof(regret)
④ count(regret)

11. regret 변수에 저장된 문자열의 길이를 확인하기 위한 방법으로 올바른 것은?

```
regret = '아니 1분만에 파이썬을 어떻게 배우냐고'
```

① len(regret)
② regret.len()
③ sizeof(regret)
④ count(regret)

정답 ①
len() 함수를 이용하면 문자열 또는 리스트 등의 길이를 확인할 수 있어요

12. 승민이는 실수로 CapsLock 버튼을 누른 채 영어 숙제를 해서 모든 문장의 대소문자가 반대로 입력이 되었다. 불쌍한 승민이를 도와줄 수 있는 파이썬 문자열 메소드는?

> hELLO? IET ME INTRODUCE MYSELF. mY NAME IS sEUNGMIN jUNG. i AM 15 YEARS OLD. tHERE ARE ...

① title ② capitalize ③ swapcase ④ lower

12. 승민이는 실수로 CapsLock 버튼을 누른 채 영어 숙제를 해서 모든 문장의 대소문자가 반대로 입력이 되었다. 불쌍한 승민이를 도와줄 수 있는 파이썬 문자열 메소드는?

> hELLO? IET ME INTRODUCE MYSELF. mY NAME IS sEUNGMIN jUNG. i AM 15 YEARS OLD. tHERE ARE ...

① title ② capitalize ③ swapcase ④ lower

정답 ③

swapcase 는 대문자를 소문자로, 소문자를 대문자로 서로 바꿔줘요

13. 다음 중 문자열 메소드에 대한 설명으로 잘못된 것은?

① 특정 문자로 시작하는지 확인하려면 startsfrom() 을 사용한다.

② 문자열 앞뒤에 불필요한 부분이나 공백이 있는 경우 strip() 을 통해 제거할 수 있다.

③ 잘못된 단어가 있는 경우 replace() 로 바꿀 수 있다.

④ find() 를 사용하면 문자열 내에서 특정 단어가 어느 위치에 있는지 알 수 있다.

13. 다음 중 문자열 메소드에 대한 설명으로 잘못된 것은?

① 특정 문자로 시작하는지 확인하려면 startsfrom() 을 사용한다.

② 문자열 앞뒤에 불필요한 부분이나 공백이 있는 경우 strip() 을 통해 제거할 수 있다.

③ 잘못된 단어가 있는 경우 replace() 로 바꿀 수 있다.

④ find() 를 사용하면 문자열 내에서 특정 단어가 어느 위치에 있는지 알 수 있다.

정답 ①

~로 시작하는지 여부를 확인하기 위한 메소드는 startswith() 랍니다

14. 다음 보기 중 출력 결과가 다른 하나를 고르시오

```
# 변수 선언
apple = '사과'
banana = '바나나'
```

① print('빨가면 {} 맛있으면 {}'.format(apple,banana))
② print('빨가면', apple, '맛있으면', banana)
③ print('빨가면 {1} 맛있으면 {0}'.format(banana, apple))
④ print(f'빨가면 apple 맛있으면 banana')

14. 다음 보기 중 출력 결과가 다른 하나를 고르시오

```
# 변수 선언
apple = '사과'
banana = '바나나'
```

① print('빨가면 {} 맛있으면 {}'.format(apple,banana))
② print('빨가면', apple, '맛있으면', banana)
③ print('빨가면 {1} 맛있으면 {0}'.format(banana, apple))
④ print(f'빨가면 apple 맛있으면 banana')

정답 ④
f-string 에서도 출력하려는 값을 {apple} 과 같이 중괄호로 감싸야 해요

15. 다음 중 문자열 내에서 줄바꿈을 위해 (★) 부분에 사용할 수 있는 탈출 문자는?

```
print('사실( ★ )포테토칩도( ★ )맛있어요')
```

① \l　　② \i　　③ \n　　④ \e

15. 다음 중 문자열 내에서 줄바꿈을 위해 (★) 부분에 사용할 수 있는 탈출 문자는?

```
print('사실( ★ )포테토칩도( ★ )맛있어요')
```

① \l　② \i　③ \n　④ \e

정답 ③ 복습해볼까요?
큰 따옴표는 \'', 작은 따옴표는 \', 역슬래시는 \\, 줄바꿈은 \n

16. 다음 중 리스트의 특징으로 올바른 것은?

① 같은 값의 중복을 허용한다.
② 숫자면 숫자, 문자면 문자끼리만 넣을 수 있다.
③ 빈 리스트는 생성할 수 없다.
④ 리스트는 대괄호 속에 넣고 싶은 값들을 점(.)으로 구분한다.

16. 다음 중 리스트의 특징으로 올바른 것은?

① 같은 값의 중복을 허용한다.
② 숫자면 숫자, 문자면 문자끼리만 넣을 수 있다.
③ 빈 리스트는 생성할 수 없다.
④ 리스트는 대괄호 속에 넣고 싶은 값들을 점(.)으로 구분한다.

정답 ①

리스트에는 같은 값을 중복해서 여러 번 넣을 수 있어요

17. 다음 실행 결과를 위해 (★) 에 들어갈 메소드는?

```
langs = ['파이썬', '자바']
langs.(  ★  )('C#')
print(langs) # 실행 결과 : ['파이썬', '자바', 'C#']
```

① add　② append　③ extend　④ insert

17. 다음 실행 결과를 위해 (★) 에 들어갈 메소드는?

```
langs = ['파이썬', '자바']
langs.( ★ )('C#')
print(langs) # 실행 결과 : ['파이썬', '자바', 'C#']
```

① add ② append ③ extend ④ insert

정답 ②
값 하나를 추가하기 위한 메소드는 append( ) 랍니다

18. 다음 중 튜플에 대한 설명으로 옳은 것을 고르시오.

A. 대괄호 속에 값들을 넣고 콤마로 구분한다
B. 값을 추가할 때 사용하는 메소드는 append() 이다
C. 리스트와는 다르게 튜플에서는 슬라이싱이 불가능하다

① A ② B ③ C ④ 정답 없음

18. 다음 중 튜플에 대한 설명으로 옳은 것을 고르시오.

A. 대괄호 속에 값들을 넣고 콤마로 구분한다
B. 값을 추가할 때 사용하는 메소드는 append() 이다
C. 리스트와는 다르게 튜플에서는 슬라이싱이 불가능하다

① A ② B ③ C ④ 정답 없음

정답 ④ 튜플은 소괄호를 통해 선언할 수 있습니다. 그리고 한 번 만들어지면 추가, 삭제, 수정이 불가능지요. 하지만 슬라이싱은 사용할 수 있답니다

19. 다음과 같이 튜플을 언패킹 했을 때 실행 결과로 올바른 것은?

```
fruits = ('apple', 'banana', 'orange', 'grape')
(mine, *yours) = fruits
print(yours)
```

① apple
② banana
③ banana orange grape
④ ['banana', 'orange', 'grape']

19. 다음과 같이 튜플을 언패킹 했을 때 실행 결과로 올바른 것은?

```
fruits = ('apple', 'banana', 'orange', 'grape')
(mine, *yours) = fruits
print(yours)
```

① apple
② banana
③ banana orange grape
④ ['banana', 'orange', 'grape']

정답 ④ mine 에는 apple 만, *yours 에는 나머지 값들이 모두 들어갑니다
이 때, 나머지 값들은 리스트 형태라는 점, 꼭 기억해주세요!

20. 다음 명령의 수행 결과로 올바른 것은?

```
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 4, 5, 6, 7}
print(set1.intersection(set2))
```

① {1, 2}
② {3, 4, 5}
③ {3, 4, 4, 5}
④ {1, 2, 3, 4, 5, 6, 7}

20. 다음 명령의 수행 결과로 올바른 것은?

```
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 4, 5, 6, 7}
print(set1.intersection(set2))
```

① {1, 2}
② {3, 4, 5}
③ {3, 4, 4, 5}
④ {1, 2, 3, 4, 5, 6, 7}

정답 ② 교집합을 구하는 문제네요! 공통으로 있는 값은 {3, 4, 4, 5} 이지만
세트는 값의 중복을 허용하지 않으므로 정답은 {3, 4, 5} 입니다.

21. 다음 중 세트에 대한 설명으로 옳은 것들로 이루어진 것은?

A. 값을 삭제하려면 remove() 를 쓸 수 있어
B. 모든 값을 한꺼번에 지우려면 clear() 를 쓰면 돼
C. discard() 를 통해 처음부터 순서대로 값을 지울 수 있어

① A, B ② B, C ③ A, C ④ A, B, C

21. 다음 중 세트에 대한 설명으로 옳은 것들로 이루어진 것은?

A. 값을 삭제하려면 remove() 를 쓸 수 있어
B. 모든 값을 한꺼번에 지우려면 clear() 를 쓰면 돼
C. discard() 를 통해 처음부터 순서대로 값을 지울 수 있어

① A, B ② B, C ③ A, C ④ A, B, C

정답 ① 세트는 순서를 보장하지 않아요. discard() 는 주어진 값을 삭제하는 메소드인데 remove() 와는 다르게 삭제하려는 값이 없어도 에러가 나지 않지요.

22. 다음 중 dictionary 를 선언하기 위한 방법으로 올바른 것은?

① dictionary = {key1:value1, key2:value2, ...}
② dictionary = [key1:value1, key2:value2, ...]
③ dictionary = (key1:value1, key2:value2, ...)
④ dictionary = key1:value1, key2:value2, ...

22. 다음 중 dictionary 를 선언하기 위한 방법으로 올바른 것은?

① dictionary = {key1:value1, key2:value2, ...}
② dictionary = [key1:value1, key2:value2, ...]
③ dictionary = (key1:value1, key2:value2, ...)
④ dictionary = key1:value1, key2:value2, ...

정답 ①
딕셔너리는 중괄호 안에 key:value 들을 콤마(,) 로 구분해서 선언합니다

23. 다음 코드를 실행한 뒤 dictionary 의 values 로 올바른 것은?

```
dictionary = {'k1':'v1', 'k2':'v2', 'k3':'v3'}
dictionary['k1'] = 'v0'
dictionary['k4'] = 'v4'
```

① v0, v2, v3
② v0, v1, v2, v3, v4
③ v0, v2, v3, v4
④ v1, v2, v3, v0, v4

23. 다음 코드를 실행한 뒤 dictionary 의 values 로 올바른 것은?

```
dictionary = {'k1':'v1', 'k2':'v2', 'k3':'v3'}
dictionary['k1'] = 'v0'
dictionary['k4'] = 'v4'
```

① v0, v2, v3
② v0, v1, v2, v3, v4
③ v0, v2, v3, v4
④ v1, v2, v3, v0, v4

정답 ③ k1 의 value 는 v1 에서 v0 으로 변경되고,
k4 라는 key 에 v4 라는 value 를 가지는 새로운 데이터가 추가되었네요

24. 기계 부품을 판매하는 A씨는 부품의 재고 관리를 효율적으로 하고 싶다. 서로 다른 고유번호를 가지는 부품들이 각각 몇 개씩 있는지 관리하기에 가장 적합한 자료형은?

고유번호	수량 (개)
NDCD_001	335
NDCD_002	223
…	…

① 리스트 ② 튜플 ③ 세트 ④ 딕셔너리

24. 기계 부품을 판매하는 A씨는 부품의 재고 관리를 효율적으로 하고 싶다. 서로 다른 고유번호를 가지는 부품들이 각각 몇 개씩 있는지 관리하기에 가장 적합한 자료형은?

고유번호	수량 (개)
NDCD_001	335
NDCD_002	223
…	…

① 리스트 ② 튜플 ③ 세트 ④ 딕셔너리

정답 ④ dic = {'NDCD_001':335, 'NDCD_002':223, …} 와 같이 key 가 고유번호, value 가 수량인 딕셔너리 자료형으로 관리하면 쉽게 재고 확인 및 수정이 가능하겠네요 ☺

25. 다음 코드에 대한 설명으로 틀린 것은?

```
my_list = [1, 2, 3, 3, 3]
my_dic = dict.fromkeys(my_list)
my_list = list(my_dic)
```

① 실행 후 my_list 에는 [1, 2, 3] 만 남게 돼
② dict.fromkeys() 를 이용해서 새로운 딕셔너리를 만들 수 있어
③ list(my_dic) 는 딕셔너리의 value 들만 뽑아서 리스트로 변환해
④ 리스트의 순서는 보장하면서 중복값을 제거하는 코드야

25. 다음 코드에 대한 설명으로 틀린 것은?

```
my_list = [1, 2, 3, 3, 3]
my_dic = dict.fromkeys(my_list)
my_list = list(my_dic)
```

① 실행 후 my_list 에는 [1, 2, 3] 만 남게 돼
② dict.fromkeys() 를 이용해서 새로운 딕셔너리를 만들 수 있어
③ list(my_dic) 는 딕셔너리의 value 들만 뽑아서 리스트로 변환해
④ 리스트의 순서는 보장하면서 중복값을 제거하는 코드야

정답 ③
딕셔너리의 value 가 아닌 key 들만 뽑아서 리스트로 변환하지요

26. 나도리조트는 객실당 4인의 손님까지 무료로 입장이 가능하며 4인을 초과하는 경우 추가비용이 발생한다. 자동 체크인 프로그램이 다음과 같이 작성되어 있을 때 실행 결과로 올바른 것은?

```
total = 2 # 총 인원
if total <= 4:
    print('추가비용 없음')
else:
    print('1인당 만원')
print('감사합니다')
```

① 추가비용 없음
감사합니다

② 1인당 만원
감사합니다

③ 추가비용 없음

④ 감사합니다

26. 나도리조트는 객실당 4인의 손님까지 무료로 입장이 가능하며 4인을 초과하는 경우 추가비용이 발생한다. 자동 체크인 프로그램이 다음과 같이 작성되어 있을 때 실행 결과로 올바른 것은?

```
total = 2 # 총 인원
if total <= 4:
    print('추가비용 없음')
else:
    print('1인당 만원')
print('감사합니다')
```

① 추가비용 없음
감사합니다
② 1인당 만원
감사합니다
③ 추가비용 없음
④ 감사합니다

정답 ① total 은 4 이하이므로 if 조건을 만족합니다. 그래서 if 문 내부의 문장이 먼저 실행되고 나서 if 문 밖에 있는 다음 문장이 실행됩니다

27. 다음 코드의 실행 결과로 알맞은 것은?

```
temp = 40 # 체온
if temp >= 39:
    print('고열입니다')
elif temp >= 38:
    print('미열입니다')
else:
    print('정상입니다')
```

① 고열입니다
② 미열입니다
③ 정상입니다
④ 정답 없음

27. 다음 코드의 실행 결과로 알맞은 것은?

```
temp = 40 # 체온
if temp >= 39:
    print('고열입니다')
elif temp >= 38:
    print('미열입니다')
else:
    print('정상입니다')
```

① 고열입니다
② 미열입니다
③ 정상입니다
④ 정답 없음

정답 ① temp 는 39보다 크기 때문에 if 의 조건은 참입니다.
if 의 조건이 참인 경우 이후에 나오는 elif 와 else 는 실행될 필요가 없겠죠?

28. 다음 코드의 실행 결과를 고르시오

```
min = 35 # 게임 시간
if min > 20:
    print('게임 많이 했네?')
    if min > 40:
        print('당장 안 꺼?')
else:
    print('뭐 해?')
```

① 게임 많이 했네?

② 게임 많이 했네?
뭐 해?

③ 게임 많이 했네?
당장 안 꺼?

④ 게임 많이 했네?
당장 안 꺼?
뭐 해?

28. 다음 코드의 실행 결과를 고르시오

```
min = 35 # 게임 시간
if min > 20:
    print('게임 많이 했네?')
    if min > 40:
        print('당장 안 꺼?')
else:
    print('뭐 해?')
```

① 게임 많이 했네?
② 게임 많이 했네?
뭐 해?
③ 게임 많이 했네?
당장 안 꺼?
뭐 해?
④ 게임 많이 했네?
당장 안 꺼?
뭐 해?

정답 ① min 은 20 보다 크므로 첫 번째 if 의 조건은 참이지만 중첩 if 의 조건은 거짓이네요. else 는 중첩이 아닌 첫 번째 if 에 상응하는 것이랍니다.

29. 나도검진센터에서는 아침 일찍 오신 10명의 고객분들께 검진 후 죽과 음료 쿠폰을 서비스로 제공한다. 반복문을 사용하여 입장 번호에 따른 쿠폰을 출력하기 위해 ①, ② 에 들어갈 키워드로 올바른 것은?

```
(   ①   ) num (   ②   ) range(10):
    print(f'죽&음료 쿠폰 (입장 번호 : {num+1})')
```

① for, of　② for, in　③ for, on　④ foreach, in

29. 나도검진센터에서는 아침 일찍 오신 10명의 고객분들께 검진 후 죽과 음료 쿠폰을 서비스로 제공한다. 반복문을 사용하여 입장 번호에 따른 쿠폰을 출력하기 위해 ①, ② 에 들어갈 키워드로 올바른 것은?

```
(  ①  ) num (  ②  ) range(10):
    print(f'죽&음료 쿠폰 (입장 번호 : {num+1})')
```

① for, of　　② for, in　　③ for, on　　④ foreach, in

정답 ② for 반복문의 기본 사용법을 묻는 문제군요!
for x in range(10): 기억하시죠? 정답은 for, in 입니다.

30. 우리반에는 30명의 학생들이 있다. 방과 후 코딩 교육과정을 진행하는데 각 1번들이 10명분의 신청서를 순서대로 모아서 반장에게 제출하기로 한다. 결과와 같이 출력하기 위해 빈 칸에 들어갈 수 있는 값은?

```
for n in range(    ★    ):
    print(f'{n}번은 {n}번부터 {n+9}번까지 모아줘')
```

결과 : 1번은 1번부터 10번까지 모아줘
11번은 11번부터 20번까지 모아줘
21번은 21번부터 30번까지 모아줘

① 1, 10, 30 ② 1, 10, 31 ③ 1, 21, 10 ④ 1, 31, 10

30. 우리반에는 30명의 학생들이 있다. 방과 후 코딩 교육과정을 진행하는데 각 1번들이 10명분의 신청서를 순서대로 모아서 반장에게 제출하기로 한다. 결과와 같이 출력하기 위해 빈 칸에 들어갈 수 있는 값은?

```
for n in range(    ★    ):
    print(f'{n}번은 {n}번부터 {n+9}번까지 모아줘')
```

결과 : 1번은 1번부터 10번까지 모아줘
11번은 11번부터 20번까지 모아줘
21번은 21번부터 30번까지 모아줘

① 1, 10, 30 ② 1, 10, 31 ③ 1, 21, 10 ④ 1, 31, 10

정답 ④ 출석번호는 1부터 시작하므로 start 는 1이고, 각 1번의 마지막 학생은 21번이니 stop 은 22~31 사이의 아무 값, 그리고 10번 만큼의 증가를 위해 step 은 10 입니다

31. 다음 코드의 빈 칸에 넣었을 때 실행 결과가 다른 하나는?

```
my_var = (   ★   )
for x in my_var:
    print(x)
```

① [1, 2, 3]
② (1, 2, 3)
③ '123'
④ 123

31. 다음 코드의 빈 칸에 넣었을 때 실행 결과가 다른 하나는?

```
my_var = (    ★    )
for x in my_var:
    print(x)
```

① [1, 2, 3]
② (1, 2, 3)
③ '123'
④ 123

정답 ④ 반복문을 통해 리스트, 튜플과 같은 자료형은 콤마로 구분된 값을,
문자열은 한 글자씩 출력합니다. 123 은 정수라서 반복 대상으로 쓸 수 없어요

32. 다음 코드의 수행 결과로 올바른 것은?

```
i = 3
while i <= 5:
    print(i)
    i += 1
```

① 3

② 3
4

③ 3
4
5

④ 3
4
5
6

32. 다음 코드의 수행 결과로 올바른 것은?

```
i = 3
while i <= 5:
    print(i)
    i += 1
```

① 3

② 3
4

③ 3
4
5

④ 3
4
5
6

정답 ③ i 는 3부터 시작하고 while 문 내에서 값을 출력 후 1씩 커집니다.
조건에 의해 i 는 5가 될 때까지만 값을 출력하고 6이 되면 반복문을 벗어나지요.

33. break 에 대한 설명으로 올바른 것은?

① 실행중인 프로그램을 즉시 종료하고자 할 때 사용한다
② 반복문 내에서 반복 수행중인 동작을 즉시 멈추고 반복문을 탈출하는 역할을 한다
③ for 반복문에서만 사용할 수 있다
④ 반복문 수행 중 의도치 않은 동작 등으로 문제가 발생한 경우 처음부터 다시 반복 수행이 필요할 때 사용한다

33. break 에 대한 설명으로 올바른 것은?

① 실행중인 프로그램을 즉시 종료하고자 할 때 사용한다
② 반복문 내에서 반복 수행중인 동작을 즉시 멈추고 반복문을 탈출하는 역할을 한다
③ for 반복문에서만 사용할 수 있다
④ 반복문 수행 중 의도치 않은 동작 등으로 문제가 발생한 경우 처음부터 다시 반복 수행이 필요할 때 사용한다

정답 ②

break 는 반복문을 즉시 탈출하기 위해 사용해요

34. 다음 코드의 수행 결과로 올바른 것은?

```
for x in range(10):
    if x % 2 == 1:
        continue
    print(x)
```

① 0
1

② 1
3
5
7
9

③ 0
2
4
6
8

④ 2
4
6
8
10

34. 다음 코드의 수행 결과로 올바른 것은?

```
for x in range(10):
    if x % 2 == 1:
        continue
    print(x)
```

① 0
1

② 1
3
5
7
9

③ 0
2
4
6
8

④ 2
4
6
8
10

정답 ③ 0 이상 10 미만의 숫자 중 2 로 나눴을 때의 나머지가 1인 홀수인 경우에는 다음 반복으로, 그렇지 않은 경우에는 출력을 하므로 0 2 4 6 8 이 출력됩니다

35. 다음 코드에서 처음으로 들여쓰기가 잘못된 곳은?

```
if 3 < 5:
    print('3은 5보다 작다') ---------- A
      if 3 > 1: ---------------------- B
        print('3은 1보다 크다') ------ C
        else:  ------------------------ D
          print('3은 1보다 작다')
```

① A ② B ③ C ④ D

35. 다음 코드에서 처음으로 들여쓰기가 잘못된 곳은?

```
if 3 < 5:
    print('3은 5보다 작다') ---------- A
      if 3 > 1: ---------------------- B
        print('3은 1보다 크다') ------ C
        else:  ------------------------ D
          print('3은 1보다 작다')
```

① A　② B　③ C　④ D

정답 ② A 의 print 문과 B 의 if 조건문은 같은 문단이므로
B 에도 A 와 동일한 간격만큼의 들여쓰기가 사용되어야 합니다

36. 다음 코드의 실행 결과로 올바른 것은?

```
my_list = ['korea', 'English', 'france']
new_list = [x.upper() for x in my_list if 'a' in x]
print(new_list)
```

① ['KOREA', 'English', 'FRANCE']
② ['KOREA', 'FRANCE']
③ ['korea', 'ENGLISH', 'france']
④ ['KOREA', 'ENGLISH', 'FRANCE']

36. 다음 코드의 실행 결과로 올바른 것은?

```
my_list = ['korea', 'English', 'france']
new_list = [x.upper() for x in my_list if 'a' in x]
print(new_list)
```

① ['KOREA', 'English', 'FRANCE']
② ['KOREA', 'FRANCE']
③ ['korea', 'ENGLISH', 'france']
④ ['KOREA', 'ENGLISH', 'FRANCE']

정답 ② 퀴즈에서 사용된 리스트 컴프리헨션을 문장으로 풀어보면 다음과 같아요
(1) my_list 에서 (2) 'a' 가 포함된 값들만 (3) 대문자로 바꿔서 (4) 새 리스트로 만들어줘

37. 다음 코드의 수행 결과로 올바른 것은?

```
def my_function():
    print('새로운')
    print('함수를')
    print('만들었어요')
```

① 아무것도 표시되지 않음

② 새로운

③ 새로운
함수를

④ 새로운
함수를
만들었어요

37. 다음 코드의 수행 결과로 올바른 것은?

```
def my_function():
    print('새로운')
    print('함수를')
    print('만들었어요')
```

① 아무것도 표시되지 않음

② 새로운

③ 새로운
함수를

④ 새로운
함수를
만들었어요

정답 ① my_function 이라는 함수를 정의하기는 했지만
호출하는 곳이 없으므로 함수가 실행되지 않아요

38. 다음 중 옳은 설명을 하는 친구를 모두 고르시오

수연 : 함수에는 전달값을 넣지 않아도 돼
지안 : 함수에는 전달값을 마음껏 넣을 수 있어
승민 : 전달값은 함수 밖에서도 얼마든지 사용할 수 있어

① 수연, 지안
② 지안, 승민
③ 수연, 승민
④ 모두 옳음

38. 다음 중 옳은 설명을 하는 친구를 모두 고르시오

수연 : 함수에는 전달값을 넣지 않아도 돼
지안 : 함수에는 전달값을 마음껏 넣을 수 있어
승민 : 전달값은 함수 밖에서도 얼마든지 사용할 수 있어

① 수연, 지안
② 지안, 승민
③ 수연, 승민
④ 모두 옳음

정답 ①
전달값은 함수 내에서만 사용 가능해요

39. 다음 요구사항에 해당하는 함수를 올바르게 구현한 코드는?

(1) 함수의 이름은 add 로 한다
(2) 2 개의 수를 num1, num2 라는 이름으로 전달받는다
(3) num1 과 num2 를 서로 더한 값을 반환한다

①
```
def add():
    return num1 + num2
```

②
```
def add():
    throw num1 + num2
```

③
```
def add(num1, num2):
    return num1 + num2
```

④
```
def add(num1, num2):
    throw num1 + num2
```

39. 다음 요구사항에 해당하는 함수를 올바르게 구현한 코드는?

(1) 함수의 이름은 add 로 한다
(2) 2 개의 수를 num1, num2 라는 이름으로 전달받는다
(3) num1 과 num2 를 서로 더한 값을 반환한다

①
```
def add():
    return num1 + num2
```

②
```
def add():
    throw num1 + num2
```

③
```
def add(num1, num2):
    return num1 + num2
```

④
```
def add(num1, num2):
    throw num1 + num2
```

정답 ③
함수 내에서 처리된 결과를 반환하는 키워드는 return 입니다

40. 다음 함수에 대한 설명으로 잘못된 것은?

```
def order(shipping='선불'):
    print(f'주문이 완료되었습니다. 배송료는 {shipping}입니다')
```

① 전달값은 1개이다
② 함수를 호출할 때 전달값은 따로 명시하지 않아도 된다
③ 함수 호출을 order() 로 할 경우 shipping 은 '선불' 이 된다
④ 기본값은 참고용이므로 함수 호출 시 전달값은 생략할 수 없다

40. 다음 함수에 대한 설명으로 잘못된 것은?

```
def order(shipping='선불'):
    print(f'주문이 완료되었습니다. 배송료는 {shipping}입니다')
```

① 전달값은 1개이다
② 함수를 호출할 때 전달값은 따로 명시하지 않아도 된다
③ 함수 호출을 order() 로 할 경우 shipping 은 '선불' 이 된다
④ 기본값은 참고용이므로 함수 호출 시 전달값은 생략할 수 없다

정답 ④ 기본값은 전달값을 따로 명시하지 않을 때 기본으로 설정되는 값이므로 기본값이 제공되는 함수 호출 시 전달값은 생략 가능합니다

41. 아래 함수를 다양한 방법으로 호출할 때 실행 결과가 다른 하나는?

```
def order(shot=2, size='Regular', takeout=True): # 커피 주문
    print(f'아메리카노 {size} 사이즈 {shot}샷')
    if takeout:
        print('포장 주문이 완료되었습니다')
    else:
        print('주문이 완료 되었습니다')
```

① order()
② order(2, takeout=True)
③ order(size='Regular')
④ order('Regular', takeout=True)

41. 아래 함수를 다양한 방법으로 호출할 때 실행 결과가 다른 하나는?

```
def order(shot=2, size='Regular', takeout=True): # 커피 주문
    print(f'아메리카노 {size} 사이즈 {shot}샷')
    if takeout:
        print('포장 주문이 완료되었습니다')
    else:
        print('주문이 완료 되었습니다')
```

① order()
② order(2, takeout=True)
③ order(size='Regular')
④ order('Regular', takeout=True)

정답 ④ 'Regular' 는 키워드를 명시하지 않았기 때문에 의도치 않게 size 가 아닌 shot 의 전달값으로 들어가게 됩니다

42. 다음 중 함수의 전달값이 몇 개인지 모를 때 가변 인자로 사용하기 위해 전달값 앞에 붙이는 기호는?

① $　② &　③ ?　④ *

42. 다음 중 함수의 전달값이 몇 개인지 모를 때 가변 인자로 사용하기 위해 전달값 앞에 붙이는 기호는?

① $　② &　③ ?　④ *

정답 ④

별 (*) 을 붙이면 되지요?! ^^

43. 다음 중 지역 변수에 대한 설명으로 잘못된 것은?

① 함수 내에서 선언된 변수는 지역 변수이다.
② 지역 변수는 외부에서는 접근이 불가능하다.
③ 함수가 호출된 이후에는 그 함수 내에 선언된 지역 변수는 외부에서도 접근 가능하다.
④ 서로 다른 함수에서 같은 이름의 지역 변수를 선언할 수 있으며, 이 때 두 변수는 서로 관련이 없다.

43. 다음 중 지역 변수에 대한 설명으로 잘못된 것은?

① 함수 내에서 선언된 변수는 지역 변수이다.
② 지역 변수는 외부에서는 접근이 불가능하다.
③ 함수가 호출된 이후에는 그 함수 내에 선언된 지역 변수는 외부에서도 접근 가능하다.
④ 서로 다른 함수에서 같은 이름의 지역 변수를 선언할 수 있으며, 이 때 두 변수는 서로 관련이 없다.

정답 ③ 지역 변수는 호출 여부와 상관 없이
외부 또는 다른 함수에서 접근이 불가능합니다

44. 다음 코드의 실행 결과로 올바른 것은?

```
x = 3
def add():
    x = 6
    x += 3
add()
print(x)
```

① 3

② 6

③ 9

④ 에러 발생

44. 다음 코드의 실행 결과로 올바른 것은?

```
x = 3
def add():
    x = 6
    x += 3
add()
print(x)
```

① 3

② 6

③ 9

④ 에러 발생

정답 ① add() 함수 내에서 x = 6 과 같이 global 선언 없이 값을 설정하면
x 라는 지역 변수가 새로 생기게 됩니다. 그래서 전역 변수인 x 는 변함 없이 3이랍니다

45. 프로그램 실행 중 사용자로부터 입력을 받기 위해 (★) 에 들어갈 함수의 이름으로 올바른 것은?

```
dream = ( ★ ) ('당신의 꿈은 무엇인가요?')
print(f'제 꿈은 {dream} 입니다')
```

① input ② in ③ ask ④ enter

45. 프로그램 실행 중 사용자로부터 입력을 받기 위해 (★) 에 들어갈 함수의 이름으로 올바른 것은?

```
dream = ( ★ ) ('당신의 꿈은 무엇인가요?')
print(f'제 꿈은 {dream} 입니다')
```

① input ② in ③ ask ④ enter

정답 ①
사용자로부터 입력을 받기 위한 내장 함수는 input() 입니다

46. open() 함수를 통해 파일을 열고 필요한 작업들을 한 뒤에 파일을 닫기 위해 호출해야 하는 함수는?

① dispose()　② clear()　③ exit()　④ close()

46. open() 함수를 통해 파일을 열고 필요한 작업들을 한 뒤에 파일을 닫기 위해 호출해야 하는 함수는?

① dispose()　② clear()　③ exit()　④ close()

정답 ④
파일을 닫는 함수는 close() 입니다

47. with 를 사용하여 파일을 열기 위한 문법으로 올바른 것은?
단, open 함수에 들어가는 값들은 편의상 ... 로 대체한다.

① f = with open(...)
② with f = open(...)
③ open(...) as f with:
④ with open(...) as f:

47. with 를 사용하여 파일을 열기 위한 문법으로 올바른 것은?
단, open 함수에 들어가는 값들은 편의상 ... 로 대체한다.

① f = with open(...)
② with f = open(...)
③ open(...) as f with:
④ with open(...) as f:

정답 ④
간단하면서 편리한 with 의 올바른 사용법, 잘 익혀두세요!

48. 다음 빈 칸에 들어갈 용어로 알맞은 것은?

A 클래스로부터 만들어진 객체 B 가 있다고 할 때,
B 는 A 의 ( ) 라고 표현한다.

① 인스턴트　② 인스턴스　③ 오브젝트　④ 블루프린트

48. 다음 빈 칸에 들어갈 용어로 알맞은 것은?

A 클래스로부터 만들어진 객체 B 가 있다고 할 때,
B 는 A 의 ( ) 라고 표현한다.

① 인스턴트 ② 인스턴스 ③ 오브젝트 ④ 블루프린트

정답 ②
어떤 클래스로부터 만들어진 객체는 그 클래스의 인스턴스(instance) 랍니다

49. 다음 클래스 객체의 name 변수에 값을 저장하기 위해 빈 칸에 들어갈 기호로 알맞은 것은?

```
class Student:
    pass

student = Student()
student(  )name = '나학생'
```

① .　　② ,　　③ :　　④ ->

49. 다음 클래스 객체의 name 변수에 값을 저장하기 위해 빈 칸에 들어갈 기호로 알맞은 것은?

```
class Student:
    pass

student = Student()
student(  )name = '나학생'
```

① .　　② ,　　③ :　　④ ->

정답 ①

클래스 객체의 변수에 접근하기 위해서는 점(.) 을 사용해요!

50. 다음 중 name 과 price 를 변수로 가지는 클래스의 __init__ 함수를 정의하는 방법으로 올바른 것은?

① def __init__(self, name, price):
② class __init__(self, name, price):
③ method __init__(self, name, price):
④ def __init__(name, price):

50. 다음 중 name 과 price 를 변수로 가지는 클래스의 __init__ 함수를 정의하는 방법으로 올바른 것은?

① def __init__(self, name, price):
② class __init__(self, name, price):
③ method __init__(self, name, price):
④ def __init__(name, price):

정답 ① __init__ 함수는 일반 함수와 동일하게 def 를 통해 정의하지만 괄호 속 다른 변수들 앞에 self 가 들어간다는 점 주의해주세요!

51. 다음 중 클래스에 관한 설명으로 잘못된 것은?

① 특정 객체에 별도의 변수를 따로 추가하게 되면 해당 클래스로 만든 모든 객체에 동일하게 변수가 추가된다
② __init__ 함수 내에서 self.name = name 과 같이 정의되는 변수를 멤버변수라고 한다
③ 서로 다른 두 객체의 멤버 변수는 서로 다른 값을 가질 수 있다
④ 멤버 변수는 없을 수도 있고 여러 개가 있을 수도 있다

51. 다음 중 클래스에 관한 설명으로 잘못된 것은?

① 특정 객체에 별도의 변수를 따로 추가하게 되면 해당 클래스로 만든 모든 객체에 동일하게 변수가 추가된다
② __init__ 함수 내에서 self.name = name 과 같이 정의되는 변수를 멤버변수라고 한다
③ 서로 다른 두 객체의 멤버 변수는 서로 다른 값을 가질 수 있다
④ 멤버 변수는 없을 수도 있고 여러 개가 있을 수도 있다

정답 ①
특정 객체에 별도의 변수를 추가해도 다른 객체에는 영향을 미치지 않아요

52. 다음 빈 칸에 들어갈 수 있는 코드로 올바른 것은?

```
class Student:
    def introduce(self, name, age):
        print(f'제 이름은 {name} 이고 {age} 살입니다')
student = Student()
student.introduce(      ★      )
```

① 'Ⅰ나도코딩'
② 'Ⅰ나도코딩', 29
③ self, 'Ⅰ나도코딩', 29
④ 정답 없음

52. 다음 빈 칸에 들어갈 수 있는 코드로 올바른 것은?

```
class Student:
    def introduce(self, name, age):
        print(f'제 이름은 {name} 이고 {age} 살입니다')
student = Student()
student.introduce(      ★      )
```

① '나도코딩'
② '나도코딩', 29
③ self, '나도코딩', 29
④ 정답 없음

정답 ②
메소드는 self 를 제외한 전달값들을 일반 함수와 같은 방식으로 호출하면 돼요

53. 다음 중 self 에 대한 설명으로 올바른 것들로 짝지어진 것은?

ㄱ. self 는 객체 자기 자신을 의미한다
ㄴ. 메소드를 정의할 때 처음 전달값은 반드시 self 를 넣는다
ㄷ. 객체를 통해 메소드를 호출할 때 self 부분에 해당하는 전달값은 따로 명시하지 않는다
ㄹ. 메소드 내에서 self.name 과 같은 형태로 멤버변수를 사용한다

① ㄱ, ㄴ　② ㄱ, ㄴ, ㄷ　③ ㄴ, ㄷ, ㄹ　④ ㄱ, ㄴ, ㄷ, ㄹ

53. 다음 중 self 에 대한 설명으로 올바른 것들로 짝지어진 것은?

ㄱ. self 는 객체 자기 자신을 의미한다
ㄴ. 메소드를 정의할 때 처음 전달값은 반드시 self 를 넣는다
ㄷ. 객체를 통해 메소드를 호출할 때 self 부분에 해당하는 전달값은 따로 명시하지 않는다
ㄹ. 메소드 내에서 self.name 과 같은 형태로 멤버변수를 사용한다

① ㄱ, ㄴ ② ㄱ, ㄴ, ㄷ ③ ㄴ, ㄷ, ㄹ ④ ㄱ, ㄴ, ㄷ, ㄹ

정답 ④
모두 옳은 설명입니다. self 에 대해 다시 한 번 정리하고 넘어가요 우리 ^^

54. 다음 Burger 클래스를 상속받아서 EggBurger 를 만들기 위한 방법으로 옳은 것은? (단, class 키워드는 생략)

```
class Burger:
    def __init__(self):
        self.add('패티')
        self.add('양상추')

    def add(self, item):
        print(f'{item} 추가')
```

① EggBurger in Burger:

② EggBurger(Burger):

③ EggBurger extends Burger:

④ EggBurger[Burger]:

54. 다음 Burger 클래스를 상속받아서 EggBurger 를 만들기 위한 방법으로 옳은 것은? (단, class 키워드는 생략)

```
class Burger:
    def __init__(self):
        self.add('패티')
        self.add('양상추')

    def add(self, item):
        print(f'{item} 추가')
```

① EggBurger in Burger:

② EggBurger(Burger):

③ EggBurger extends Burger:

④ EggBurger[Burger]:

정답 ② class EggBurger(Burger): 와 같이 클래스명 뒤에 소괄호를 이용하여 상속 받으려는 부모 클래스명을 적으면 돼요

55. 다음 코드의 자식 클래스에서 부모 클래스의 메소드를 호출하기 위해 빈 칸에 들어갈 값으로 알맞은 것은?

```
class Parent:
    def method_a(self):
        pass

class Children(Parent):
    def method_b(self):
        (   ★   ).method_a()
```

① Parent

② base

③ super

④ super()

55. 다음 코드의 자식 클래스에서 부모 클래스의 메소드를 호출하기 위해 빈 칸에 들어갈 값으로 알맞은 것은?

```
class Parent:
    def method_a(self):
        pass

class Children(Parent):
    def method_b(self):
        (   ★   ).method_a()
```

① Parent

② base

③ super

④ super()

정답 ④ super 를 이용해서 부모 클래스의 메소드를 호출하기 위해서는 괄호를 포함한 super() 로 사용해야 합니다. 이 때 self 는 필요 없지요

56. 다음 중 상속에 대한 설명으로 잘못된 것은?

① 상속을 이용하면 코드의 중복 입력 없이 부모 클래스의 기능을 그대로 이용할 수 있다
② 여러 개의 클래스로부터 상속받는 것을 다중상속이라 한다
③ 메소드 내에서 super() 를 통해 부모 클래스의 메소드에 접근할 수 있다
④ 다중상속은 클래스명 뒤의 괄호 속에 여러 개의 클래스를 콜론(:)으로 구분하여 적는다

56. 다음 중 상속에 대한 설명으로 잘못된 것은?

① 상속을 이용하면 코드의 중복 입력 없이 부모 클래스의 기능을 그대로 이용할 수 있다
② 여러 개의 클래스로부터 상속받는 것을 다중상속이라 한다
③ 메소드 내에서 super() 를 통해 부모 클래스의 메소드에 접근할 수 있다
④ 다중상속은 클래스명 뒤의 괄호 속에 여러 개의 클래스를 콜론(:)으로 구분하여 적는다

정답 ④ 다중 상속은 콜론(:) 이 아닌 콤마(,) 로 구분하여
여러 개의 클래스를 적으면 돼요

57. 부모 클래스의 메소드를 자식 클래스에서 새롭게 정의하여
기존 동작을 개선하거나 새로운 동작을 수행하도록 하는 것은?

① 메소드 오버라이딩 (overriding)
② 메소드 오버로딩 (overloading)
③ 메소드 오버라이팅 (overwriting)
④ 메소드 리라이팅 (rewriting)

57. 부모 클래스의 메소드를 자식 클래스에서 새롭게 정의하여
기존 동작을 개선하거나 새로운 동작을 수행하도록 하는 것은?

① 메소드 오버라이딩 (overriding)
② 메소드 오버로딩 (overloading)
③ 메소드 오버라이팅 (overwriting)
④ 메소드 리라이팅 (rewriting)

정답 ① 부모 클래스의 메소드를 자식 클래스에서 새롭게 정의 하는 것은
메소드 오버라이딩 (overriding) 입니다.

58. 다음 중 pass 의 사용 예시로 올바르지 않은 것은?

① while True:
    pass

② def add():
    pass

③ if pass:
    print('to be defined')

④ class Book:
    pass

58. 다음 중 pass 의 사용 예시로 올바르지 않은 것은?

① while True:
    pass

② def add():
    pass

③ if pass:
    print('to be defined')

④ class Book:
    pass

정답 ③

pass 는 if 문의 '조건'으로 사용될 수는 없답니다

59. 다음 중 예외처리에 대한 설명으로 잘못된 것은?

① 수행 문장에서 에러가 발생했을 때 프로그램이 멈추지 않고 계속해서 수행될 수 있도록 에러를 처리하는 것을 말한다
② except: 구문은 try: 구문에서 에러가 발생했을 때 수행할 문장들을 적어주면 된다
③ else: 구문은 try: 구문 내의 문장 수행에 아무런 문제가 없을 때 수행할 문장들을 적어주면 된다
④ 에러가 발생해도 별다른 예외처리를 하지 않고 계속 진행하고자 하는 경우에는 try: 만 단독으로 정의해주면 된다

59. 다음 중 예외처리에 대한 설명으로 잘못된 것은?

① 수행 문장에서 에러가 발생했을 때 프로그램이 멈추지 않고 계속해서 수행될 수 있도록 에러를 처리하는 것을 말한다

② except: 구문은 try: 구문에서 에러가 발생했을 때 수행할 문장들을 적어주면 된다

③ else: 구문은 try: 구문 내의 문장 수행에 아무런 문제가 없을 때 수행할 문장들을 적어주면 된다

④ 에러가 발생해도 별다른 예외처리를 하지 않고 계속 진행하고자 하는 경우에는 try: 만 단독으로 정의해주면 된다

정답 ④

try: 는 항상 except: 또는 finally: 와 함께 쌍을 이뤄야 해요

60. 다음 코드의 출력 결과로 올바른 것은?

```
try:
    int('일이삼') # ValueError 발생
except ValueError:
    print('값이 이상해요')
except Exception as err:
    print('에러가 발생했어요')
finally:
    print('수행 종료')
```

① 값이 이상해요

② 값이 이상해요
수행 종료

③ 수행 종료

④ 값이 이상해요
에러가 발생했어요
수행 종료

60. 다음 코드의 출력 결과로 올바른 것은?

```
try:
    int('일이삼') # ValueError 발생
except ValueError:
    print('값이 이상해요')
except Exception as err:
    print('에러가 발생했어요')
finally:
    print('수행 종료')
```

① 값이 이상해요

② 값이 이상해요
수행 종료

③ 수행 종료

④ 값이 이상해요
에러가 발생했어요
수행 종료

정답 ② ValueError 가 발생하므로 '값이 이상해요' 가 먼저 출력되고 나서 마지막으로 항상 수행되는 finally: 구문의 '수행 종료' 가 출력됩니다

61. 파이썬에서 기본으로 제공되는 random 모듈에서 choice 함수만 가져다가 쓰기 위한 방법으로 올바른 것은?

① import random as choice

② import random and choice

③ import random from choice

④ from random import choice

61. 파이썬에서 기본으로 제공되는 random 모듈에서 choice 함수만 가져다가 쓰기 위한 방법으로 올바른 것은?

① import random as choice

② import random and choice

③ import random from choice

④ from random import choice

정답 ④ 모듈에서 하나의 함수만 사용하기 위해서는
from 모듈 import 함수 <= 이렇게 하면 돼요

62. 다음 중 패키지에 대한 설명으로 잘못된 것은?

① 패키지는 여러 모듈들의 모임이다

② 패키지는 하나의 폴더 안에 여러 개의 파이썬 파일들로 구성된다

③ 패키지에는 반드시 __init__.py 라는 파일이 있어야 한다

④ 패키지는 import 또는 from ~ import 구문을 통해 가져다 쓸 수 있다

62. 다음 중 패키지에 대한 설명으로 잘못된 것은?

① 패키지는 여러 모듈들의 모임이다

② 패키지는 하나의 폴더 안에 여러 개의 파이썬 파일들로 구성된다

③ 패키지에는 반드시 __init__.py 라는 파일이 있어야 한다

④ 패키지는 import 또는 from ~ import 구문을 통해 가져다 쓸 수 있다

정답 ③ 과거에는 패키지로 인식되기 위해 __init__.py 파일이 필요했지만 파이썬 3.3 버전 이후부터는 이 파일이 없어도 돼요