

## 변수 예제

1. 다음 코드의 실행 결과를 쓰시오.

```
>>> a = 777
```

```
>>> b = 777
```

```
>>> print(a == b, a is b)
```

True True

2. 다음 중 변수를 메모리에서 삭제하기 위해 사용하는 명령어는? 1

☒ del

☐ delete

☐ remove

☐ pop

☐ clear

3. 빈칸에 들어갈 각각의 코드 실행 결과를 쓰시오.

```
>>> a = 3.5
```

```
>>> b = int(3.5)
```

```
>>> print(a*((a // b) * 2))
```

(가)

12.25

```
>>> print(((a - b) * a) // b)
```

(나)

0 0

```
>>> b = (((a - b) * a) % b)
```

```
>>> print(b)
```

(다)

1.75

```
>>> print((a * 4) % (b * 4))
```

(라)

0.0

4. 입력받은 섭씨온도를 화씨온도로 변환하는 프로그램을 코딩하려고 한다. 코드 순서를 바르게 나열한 것은? 4

(1) fahrenheit = (( 9 / 5) \* celsius) + 32

(2) celsius = input("섭씨온도를 입력하세요:")

(3) print("섭씨온도:", celsius, "화씨온도:", fahrenheit)

(4) celsius = float(input("섭씨온도를 입력하세요:"))

☐ (2) - (1) - (3)

☐ (2) - (4) - (3) - (1)

☐ (3) - (1) - (2) - (3)

☒ (4) - (1) - (3)

☐ (5) - (4) - (2) - (3)

5. 다음 변수 a의 자료형은? 2

```
a = "True"
```

☐ 소수형

☒ 문자형

☐ 실수형

☐ 불린형

☐ 정수형

6. 다음과 같은 코드 작성 시, 실행 결과로 알맞게 찍지어진 것은? 3

```
>>> a = 10.6
```

```
>>> b = 10.5
```

```
>>> print(a * b)
```

```
>>> print(type(a + b))
```

☐ 111.3, <class 'int'>

☐ 111.3, <class 'str'>

☒ 111.3, <class 'float'>

☐ 105.0, <class 'int'>

☐ 105.0, <class 'float'>

7. box = "apple"의 뜻은? 2

- ① box 변수의 이름은 apple이다.      ☒ box 변수에 apple 값을 넣어라.
- ③ box 변수와 apple은 같다.      ④ box 변수는 apple이다.
- ⑤ apple 변수는 box이다.

8. 다음과 같이 코드를 작성했을 때, 실행 결과로 알맞은 것은? 2

```
>>> a = "3.5"  
>>> b = 4  
>>> print(a * b)
```

- ① error      ☒ 3.53.53.53.5      ③ 14.0      ④ 14      ⑤ "14"

9. a = "3.5", b = "1.5"일 때, print(a + b)의 실행 결과는? 2

- ① 5      ☒ 3.51.5      ③ a + b      ④ ab      ⑤ 2

10. 다음과 같이 코드를 작성했을 때, 실행 결과로 알맞은 것은? 3

```
>>> a = '3'  
>>> b = float(a)  
>>> print(b ** int(a))
```

- ① TypeError      ② '27.0'      ☒ 27.0      ④ 27      ⑤ '27'

11. 변수(variable)에 대한 설명으로 틀린 것은? 3

- ① 프로그램에서 사용하기 위한 특정한 값을 저장하는 공간이다.
- ② 선언되는 순간 메모리의 특정 영역에 공간이 할당된다.
- ☒ 변수에 할당된 값은 하드디스크에 저장된다.
- ④ A = 8은 "A는 8이다"라는 뜻이 아니다.
- ⑤ '2x + 7y'는 14라고 하면, 이 식에서 x와 y가 변수이다.

12. 다음과 같이 코드를 작성했을 때, 실행 결과로 알맞은 것은? 5

```
>>> a = '20'  
>>> b = '4'  
>>> print(type(float(a / b)))
```

- ① <class 'int'>      ② <class 'str'>      ③ <class 'float'>      ④ 4. 3.3333333333      ☒ TypeError

13. 다음 코드의 실행 결과가 나오도록 빈 칸에 코드를 추가하시오.(단, 변수를 삭제하면 안 됨.)

c = "50" 또는 c = str(50)

```
>>> a = "Gachon"  
>>> b = "CS"  
>>> c = 200
```

C = C // 4

```
>>> print(a, b, c)  
Gachon CS 50
```

14. 다음 코드의 예상되는 실행 결과를 쓰시오.

(가) `print("1.0" * 5)` 1.01.01.01.01.0 (나) `print("1.0" + 2)` TYPE Error  
(다) `print("Hanbit" + "Python")` HanbitPython (라) `print("3.5" + "0.5")` 3.50.5

15. 변수명을 지을 때 권장하는 규칙 중 틀린 것은? 2

- ① 변수명은 알파벳, 숫자, 언더스코어( ) 등을 사용하여 표현할 수 있다. O
- ✓ ② 변수명은 의미 있는 단어로 쓰는 것을 권장하며, 한글도 사용할 수 있다. X
- ③ 변수명은 대소문자가 구분된다. O
- ④ 문법으로 사용되는 특별한 예약어는 변수명으로 쓰지 않는다. O
- ⑤ 변수명은 "a", "b" 등으로 사용하는 것은 권장하지 않는다. O

## 리스트 예제

1. 다음 코드의 실행 결과를 쓰시오. [0, 1, 2] [0, 1]

```
a = [0, 1, 2, 3, 4]
print(a[:3], a[:-3])
```

2. 다음 코드의 실행 결과를 쓰시오. [4, 3, 2, 1, 0]

```
a = [0, 1, 2, 3, 4]
print(a[::-1])
```

3. 다음 코드의 실행 결과를 쓰시오. [['egg', 'salad', 'bread'], ['lamb', 'chicken'], ['apple']]

```
first = ["egg", "salad", "bread", "soup", "canafe"]
second = ["fish", "lamb", "pork", "beef", "chicken"]
third = ["apple", "banana", "orange", "grape", "mango"]

order = [first, second, third]
john = [order[0][:2], second[1:3], third[0]]
del john[2]
john.extend([order[2][0:1]])
print(john)
```

4. 다음 코드의 실행 결과를 쓰시오. [1, 2, 3, 4] None

```
list_a = [3, 2, 1, 4]
list_b = list_a.sort()
print(list_a, list_b)
```

5. 다음 코드의 실행 결과를 쓰시오. | None

```
a = [5, 7, 3]
b = [3, 9, 1]
c = a + b
c = c.sort()
print(c)
```

6. GUI와 CLI의 설명으로 틀린 것은? 5

- ① GUI는 Graphical User Interface의 약자이다.
- ② GUI는 마우스로 아이콘을 클릭하며 프로그램을 작동시키는 컴퓨팅 환경을 말한다.
- ③ CLI는 Command Line Interface의 약자이다.
- ④ CLI는 텍스트를 사용하여 컴퓨터에 명령을 입력하는 인터페이스 체계를 말한다.
- ⑤ CLI는 윈도우에서는 불가능하고, 맥 또는 리눅스에서만 터미널로 작동한다.

7. 다음 코드의 실행 결과를 쓰시오.

```
fruits = ['apple', 'banana', 'cherry', 'grape', 'orange', 'strawberry', 'melon']
print(fruits[-3:], fruits[1:3])
```

['orange', 'strawberry', 'melon'] ['banana', 'orange']



8. 다음 코드의 실행 결과를 쓰시오. *[1, 2, 3, 4, 1, 2, 3, 4]*

```
num = [1, 2, 3, 4]
print(num * 2)
```

9. 다음 코드의 실행 결과를 쓰시오. *false 6*

```
a = [1, 2, 3, 5]
b = ['a', 'b', 'c', 'd', 'e']
a.append('g')
b.append(6)
print('g' in b, len(b))
```

10. 다음과 같이 코드를 작성했을 때, 실행 결과로 알맞은 것은? *3*

```
list_a = ['Hankook', 'University', 'is', 'an', 'academic', 'institute', 'located', 'in', 'South Korea']
list_b = [ ]
for i in range(len(list_a)):
    if i % 2 != 1:
        list_b.append(list_a[i])
print(list_b)
```

① None

② Error

③ ☒ ['Hankook', 'is', 'academic', 'located', 'South Korea']

④ ['University', 'an', 'institute', 'in']

⑤ ['Hankook', 'University', 'is', 'an', 'academic', 'institute', 'located', 'in', 'South Korea']

11. 다음 코드를 실행한 후, 2018과 "2018"을 각각 입력했을 경우 알맞은 실행 결과끼리 묶인 것은? *3*

```
admission_year = input("입학 연도를 입력하세요: ")
print(type(admission_year))
```

① <class 'str'>, <class 'float'>

② <class 'int'>, <class 'str'>

③ ☒ <class 'str'>, <class 'str'>

④ <class 'int'>, <class 'int'>

⑤ <class 'float'>, <class 'int'>

12. 다음 코드의 실행 결과를 쓰시오. ['Korea', 'Japan', 'China', ['Seoul', [2, 3],

```
country = ["Korea", "Japan", "China"]
capital = ["Seoul", "Tokyo", "Beijing"]
index = [1, 2, 3]
country.append(capital)
country[3][1] = index[1:]
print(country)
```

'Beijing']]

13. 다음 코드의 실행 결과가 나오도록 빈칸에 알맞은 코드를 쓰시오. [week1 + week2]

```
>>> week1 = ["Mon", "Tue", "Wed"]
>>> week2 = ["Thu", "Fri", "Sat", "Sun"]
>>> week3 = 
>>> print(week2[:len(week3) + 1])
['Thu', 'Fri']
```

14. 다음과 같이 코드를 작성했을 때 예측되는 실행 결과를 쓰고, 이러한 결과가 나오는 이유에 대해 서술하시오.

```
>>> a = [5, 4, 3, 2, 1]
>>> b = a
>>> c = [5, 4, 3, 2, 1]
>>> a is b
```

True

```
>>> a is c
```

False