



연습 문제



연습 문제

변수

스칼라 자료형

컬렉션 자료형

Null과 키워드

제어문

함수

클래스와 모듈

연습 문제



간단한 연습 문제를 준비했습니다.

어렵게 느껴지더라도 걱정하지 마세요!

복습할 기회이자 **ChatGPT를 활용할 좋은 기회**입니다. ([🔗 ChatGPT 링크](#))

변수

1. 변수를 사용하는 이유를 설명하세요.



변수는 데이터를 저장하고 재사용할 수 있도록 하며, 코드의 가독성을 높이고 유지보수를 쉽게 해줍니다.

2. `x`에 5를 할당한 후, `x`의 값을 출력하세요.

```
x = 5
```

```
print(x) # 5
```

3. `x` 와 `y` 에 각각 5와 10을 저장한 후, 두 변수의 합을 출력하세요.

```
x = 5  
y = 10  
print(x + y) # 15
```

4. 다음 변수 선언 중 올바르지 않은 것을 찾으세요.

```
1var = 10  
my-var = 20  
_name2 = 'Python'  
True = 5
```



```
1var = 10 # 숫자로 시작 불가  
my-var = 20 # 하이픈(-) 사용 불가  
_name2 = 'Python'  
True = 5 # 키워드 사용 불가
```

5. 변수 `x` 에 10을 할당한 후, 값을 5 증가시켜 업데이트한 뒤 출력하세요.

```
x = 10  
x = x + 5  
print(x) # 15
```

스칼라 자료형

6. 정수 10의 자료형을 확인하는 코드를 작성하세요.

```
type(10) # 출력: int
```

7. `x = 3.99` 일 때, `x` 를 정수형으로 변환하는 코드를 작성하고 결과를 예측하세요.

```
x = 3.99
int(x) # 출력: 3 (소수점 이하 버림)
```

8. `7 // 2` 와 `7 / 2` 를 출력하고, 차이를 설명하세요.

```
print(7 // 2) # 3 (몫)
print(7 / 2)  # 3.5 (일반 나눗셈)
```

9. 변수 `s = '3.14'` 를 실수형으로 변환하는 코드를 작성하세요.

```
s = '3.14'
float(s) # 출력: 3.14
```

10. `float(10)` 의 출력 결과를 예측하세요.

```
print(float(10)) # 10.0
```

11. 변수 `s = 'Hello'` 일 때, `s[0]` 과 `s[-1]` 의 값을 출력하세요.

```
s = 'Hello'
print(s[0]) # 'H'
print(s[-1]) # 'o'
```

12. `a = '김판다'` 일 때, `a` 의 길이를 출력하세요.

```
a = '김판다'
print(len(a)) # 3
```

13. `'123' + '456'` 의 출력 결과를 예측하세요.

```
print('123' + '456') # '123456'
```

14. 문자열 `s = 'Hello, World!'` 에서 마지막 6글자만 출력하세요.

```
s = 'Hello, World!'
print(s[-6:]) # 'World!'
```

15. 문자열 `s = '12345'` 에서 홀수 번째 문자만 출력하세요.

```
s = '12345'
print(s[::2]) # '135'
```

16. `5 * 2` 의 결과가 `10.0` 과 같은지 비교하여 불(`bool`) 자료형을 출력하세요.

```
print(5 * 2 == 10.0) # True: 같다
```

17. 문자열 `'hello'` 와 `'Hello'` 를 다르다(`!=`) 연산으로 비교하여 불(`bool`) 자료형을 출력하세요.

```
print('hello' != 'Hello') # True: 다르다
```

18. 변수 `a` 에 `True` , `b` 에 `False` 를 할당한 후, `a + b` 의 결과를 출력하세요.

```
a = True
b = False
print(a + b) # 1 (True는 1, False는 0으로 간주)
```

컬렉션 자료형

19. 리스트 `[1, 2, 3]` 과 `[4, 5, 6]` 을 합쳐 하나의 리스트를 출력하세요.

```
print([1, 2, 3] + [4, 5, 6]) # [1, 2, 3, 4, 5, 6]
```

20. 리스트 `numbers = [10, 20, 30, 40]` 에서 첫 번째 원소를 출력하세요.

```
numbers = [10, 20, 30, 40]
print(numbers[0]) # 10
```

21. 리스트 `numbers = [10, 20, 30, 40]` 의 길이를 출력하세요.

```
numbers = [10, 20, 30, 40]
print(len(numbers)) # 4
```

22. 리스트 `numbers = [10, 20, 30, 40, 50]` 에서 앞의 3개 원소만 출력하세요.

```
numbers = [10, 20, 30, 40, 50]
print(numbers[:3]) # [10, 20, 30]
```

23. 튜플 `t = (10, 20, 30, 40, 50)` 에서 세 번째 원소를 출력하세요.

```
t = (10, 20, 30, 40, 50)
print(t[2]) # 30
```

24. 두 개의 튜플 `(1, 2, 3)` 과 `(4, 5, 6)` 을 하나의 튜플로 합치고 출력하세요.

```
print((1, 2, 3) + (4, 5, 6)) # (1, 2, 3, 4, 5, 6)
```

25. 튜플 언패킹을 사용하여 변수 `x` 에 5, 변수 `y` 에 10을 할당하세요.

```
x, y = 5, 10
```

26. 딕셔너리 `d = {'A': 10, 'B': 20, 'C': 30}` 에서 키 `'B'` 의 값을 출력하세요.

```
d = {'A': 10, 'B': 20, 'C': 30}
print(d['B']) # 20
```

27. 빈 딕셔너리 `d = {}` 에 키 `'X'` 와 값 `100` 을 추가하는 코드를 작성하세요.

```
d = {}
d['X'] = 100
print(d) # {'X': 100}
```

28. 딕셔너리 `d = {'apple': 3, 'banana': 5}` 에서 키 `'banana'` 의 값을 10으로 변경하는 코드를 작성하세요.

```
d = {'apple': 3, 'banana': 5}
d['banana'] = 10
print(d) # {'apple': 3, 'banana': 10}
```

29. 딕셔너리 `d = {'a': 1, 'b': 2, 'a': 3}` 의 출력 결과를 예측하세요.

```
d = {'a': 1, 'b': 2, 'a': 3}
print(d) # {'a': 3, 'b': 2}
```

30. 집합 `set1 = {3, 2, 1, 3}` 의 출력 결과를 예측하세요.

```
set1 = {3, 2, 1, 3}
```

```
print(set1) # {1, 2, 3}
```

Null과 키워드

31. `x = None` 일 때, `x`의 자료형을 출력하세요.

```
x = None
print(type(x)) # <class 'NoneType'>
```

32. `None` 과 빈 문자열 `''` 이 같은지 비교해 출력하세요.

```
print(None == '') # False
```

33. `NaN` 을 생성하고 변수 `a` 에 할당한 뒤, 변수 `a` 의 자료형을 출력하세요.

```
a = float('nan')
print(type(a)) # <class 'float'>
```

34. 변수 `a` 에 `NaN` 을 할당한 후, `a + 2` 의 결과와 `a` 와 `NaN` 의 동등 비교 (`==`) 결과를 출력하세요.

```
a = float('nan')
print(a + 2) # nan
print(a == float('nan')) # False
```

35. `a = (1, 2)` , `b = (1, 2)` 일 때, `a is b` 의 결과를 출력하세요.

```
a = (1, 2)
b = (1, 2)
print(a is b) # False
```

36. 1 이 리스트 [1, 2, 3] 에 속하는지 확인하는 코드를 작성하세요.

```
1 in [1, 2, 3] # 출력: True
```

제어문

37. x = 5 일 때, x 가 양수이면 '양수입니다.' 를 출력하는 if 문을 작성하세요.

```
x = 5
if x > 0:
    print('양수입니다.') # '양수입니다.'
```

38. 문자열 s = 'python' 이 5글자 이상이면 '긴 문자열' 을 출력하세요.

```
s = 'python'
if len(s) >= 5:
    print('긴 문자열') # '긴 문자열'
```

39. 변수 num = 8 이 짝수이면 '짝수입니다.', 홀수이면 '홀수입니다.' 를 출력하는 if-else 문을 작성하세요.

```
num = 8
if num % 2 == 0:
    print('짝수입니다.') # '짝수입니다.'
else:
    print('홀수입니다.')
```

40. for 문의 반복 대상으로 사용될 수 있는 리스트, 튜플, 문자열, range 객체 등을 통칭하는 용어는?



이터러블(iterable)

41. `for` 문을 사용하여 0~5까지 정수를 하나씩 출력하세요.

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

42. 리스트 `[0, 2, 3, 5]`의 각 원소를 제공하여 하나씩 출력하세요.

```
for num in [0, 2, 3, 5]:  
    print(num ** 2)
```

43. 문자열 `'김판다'`를 `for` 문을 사용하여 거꾸로 한 글자씩 출력하세요.

```
word = '김판다'  
for i in word[::-1]:  
    print(i)
```

44. `for` 문을 사용해 리스트 `[1, 2, 3, 4]`의 모든 원소의 합을 구하고 출력하세요.

```
total = 0  
for i in [1, 2, 3, 4]:  
    total += i # total = total + i 와 동일  
print(total) # 출력: 10
```

함수

45. 함수의 입력값을 받기 위해 정의된 변수를 무엇이라고 하나요?



매개변수

46. 함수를 호출할 때, 실제 매개변수에 전달되는 값을 무엇이라고 하나요?

✓ 인수

47. 매개변수에 별도의 인수가 전달되지 않을 때 자동으로 전달되는 인수를 무엇이라고 하나요?

✓ 기본값(default)

48. 두 개의 숫자를 입력받아 합을 반환하는 함수 `add` 를 키워드 `def` 로 정의하고 **3, 4**의 합을 출력하세요.

```
def add(a, b):  
    return a + b  
  
print(add(3, 4)) # 7
```

49. 두 개의 숫자를 입력받아 합을 반환하는 함수를 `lambda` 함수로 정의하세요.

```
lambda a, b: a + b
```

50. 아래 함수가 주어졌을 때, `func(1, 3)` 의 출력 결과를 확인하세요.

```
def func(x=0, y=1, z=2):  
    return x + y + z
```

✓

```
def func(x=0, y=1, z=2):  
    return x + y + z  
  
print(func(1, 3)) # 6
```

51. 50번 문제에서 각 매개변수에 전달된 인수는 무엇인가요?



x=1, y=3, z=2

클래스와 모듈

52. 사용자 정의 자료형을 생성하는 틀 역할을 하는 것은 무엇인가요?



클래스

53. 클래스로부터 만들어진 실체이며, 클래스의 속성과 메서드를 사용할 수 있는 것은 무엇인가요?



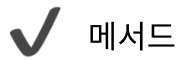
객체

54. 클래스 내부에서 선언되는 변수로, 객체에 데이터를 저장하는 것은 무엇인가요?



속성

55. 클래스 내부에서 정의된 함수로, 객체의 동작을 정의하는 것은 무엇인가요?



56. **random** 모듈을 불러오는 코드를 작성하세요.

```
import random
```

57. **pandas** 모듈을 **pd** 라는 이름으로 불러오는 코드를 작성하세요.

```
import pandas as pd
```