

파이썬 과목평가 연습문제

1. 산술연산자 +, - 보다 연산자의 우선순위가 낮은 연산자를 고르시오.

- 1) 산술연산자 *, /
- 2) 제곱연산자 **
- 3) 단항연산자 +, -
- 4) 비교연산자 >, <

2. 지문의 코드를 실행하였을 때, 출력되는 결과로 옳은 것을 고르시오.

```
for i in range(5):  
    if i == 3:  
        continue  
    print(i, end=' ')
```

- 1) 0 1 2 3
- 2) 0 1 2 4
- 3) 1 2 4 5
- 4) 1 2 3 4

3. 함수에 대한 설명으로 옳지 않은 것을 고르시오.

- 1) 함수의 선언을 할 때는 def 키워드를 활용한다.
- 2) 함수는 입력값으로 인자(argument)를 넘겨줄 수 있다.
- 3) 함수의 return으로 반드시 하나의 객체가 반환된다.
- 4) 함수의 return 값이 없으면 오류가 발생한다.

4. 지문의 코드를 실행하였을 때, 출력되는 결과로 옳은 것을 고르시오.

```
cities = {'광주', '나주', '화순'}  
cities.update({'광주', '대구', '부산', '울산'}, {'부산', '울산', '포항'})  
print(cities)
```

- 1) 광주, 나주, 화순, 광주, 대구, 부산, 울산, 부산, 울산, 포항
- 2) 광주, 나주, 화순, 광주, 대구, 부산, 울산, 포항
- 3) 광주, 나주, 화순, 대구, 부산, 울산, 포항
- 4) 광주, 나주, 화순

5. 다음 중 옳지 않은 내용을 고르시오.

- 1) 조건문은 다른 조건문에 중첩할 수 있습니다.
- 2) in, not in 연산자를 통해 동일한 object인지 확인할 수 있다.
- 3) enumerate 함수는 열거 객체를 돌려줍니다.
- 4) 사용자가 의도하지 않아도 파이썬 내부적으로 자동으로 형변환할 수 있다.

6. 지문의 코드를 실행하였을 때, 출력되는 결과를 작성하시오.

```
kids = {'name': 'Kim', 'birthday': 0112, 'city': '광주'}  
print(list(kids))
```

7. 지문의 코드를 실행하였을 때 결과가 다음과 같을 때, (A)와 (B)에 들어갈 코드나 값을 작성하시오.

<코드>

```
(A) my_package.math.tools import (B)
print(my_max(2, e-1)
print(pi)
```

<결과값>

3.141592653589793
1.718281828459045

8. 원의 중심과 둘레, 넓이를 구하기 위해 만든 지문의 코드를 실행하였을 때, (A)와 (B)에 들어갈 코드나 값, 출력되는 결과(C)를 작성하시오.

```
class Circle:
```

```
    pi = 3.14
```

```
    def __(A)__(self, r, x, y):
```

```
        self.r = r
```

```
        self.x = x
```

```
        self.y = y
```

```
    def center(self):
```

```
        return (self.x, self.y)
```

```
    def circumference(self):
```

```
        return 2 * (B) * self.r
```

```
    def area(self):
```

```
        return (B) * self.r * self.r
```

```
circle = Circle(5, 0, 5)
print(int(circle.area() - circle.circumstance()))
```

9. 어떠한 동작을 반복적으로 수행하도록 하는 명령문인 반복문에는 크게 while을 활용하는 반복문과 for을 활용하는 반복문이 있다. while을 활용하는 반복문의 특징과 주의해야 할 점을 for을 활용하는 반복문과의 차이점을 위주로 설명하시오.

10. 지문의 코드를 실행하였을 때, 출력되는 결과와 그 이유를 서술하시오. 만약, 오류가 발생한다면, 어떤 오류가 발생하는지 왜 발생하는지도 작성하시오.

```
favorite_foods = ['caffe latte', 'pizza', 'sushi']
favorite_foods.extend('pasta')
favorite_foods.append('chocolate')
favorite_foods.remove('pasta')
print(favorite_foods)
```

1. 4) 비교연산자

해설 : 비교연산자와 in, is는 산술연산자 +, - 보다 우선순위가 낮다. not, and, or 등이 그 아래에 위치한다.

2. 2) 0 1 2 4

해설 : 0부터 4까지의 범위를 순회하며 출력하는 반복문 중 3이 나오는 경우 continue 하게 되는 조건문이다.

3. 4) 함수의 return 값이 없으면 오류가 발생한다.

해설 : 함수의 return 값이 없으면 None을 반환한다.

4. 3) 광주, 나주, 화순, 대구, 부산, 울산, 포항

해설 : set은 중복된 값이 없는 자료구조이며, '.update()'을 활용하여 여러 값을 추가할 수 있다.

5. 2) in, not in 연산자를 통해 동일한 object인지 확인할 수 있다.

해설 : in, not in 연산자는 멤버십 연산자로 요소가 시퀀스에 속해 있는지 확인할 수 있다. is 연산자를 통해 동일한 object인지 확인할 수 있다.

6. ['name', 'birthday', 'city']

해설 : 딕셔너리를 str, list, tuple, set으로 형변환할 경우 딕셔너리의 key값만 출력된다.

7. (A) : from (B) : *

해설 : 특정한 함수, 혹은 attribute만 활용하고 싶을 때는 'from () import ()'과 같이 작성할 수 있다. 만약 import 뒤에 '*'을 활용할 경우 해당하는 모듈 내의 모든 변수, 함수, 클래스를 가져올 수 있다.

8. (A) : init (B) : Circle.pi (C) : 47

해설 : 생성자 메서드는 인스턴스 객체가 생성될 때 자동으로 호출되는 함수로, 반드시 '__init__(self)'라는 이름으로 정의한다. 또한 원의 둘레와 넓이를 구하기 위해서는 class Circle에서 정의된 'pi = 3.14'를 활용하기 위해 Circle.pi를 'self.pi'와 곱

하는 연산이 이루어져야 한다. print 내의 값을 연산하면 47.1이 나오고 이를 int()를 통해 정수로 변환하면, 출력값은 47이 된다.

9. 해설 : while을 활용하는 반복문은 조건식이 참인 경우 반복적으로 코드를 실행한다. 따라서 시퀀스를 포함한 순회가능한 객체의 요소들을 순회하며 시퀀스의 마지막 값에 접근할 경우 끝나게 되는 for을 활용하는 반복문과 달리 반드시 종료조건을 설정해야 한다는 주의점을 가진다.

10. 해설 : list에서는 append()를 통해서 값을 추가할 수 있고, extend()를 통해서 list, range, tuple, string 값을 붙일 수 있다. 또한 remove(n)를 통해서 값이 n인 첫 번째 항목을 제거할 수 있고, 그런 항목이 없으면 ValueError가 발생한다. 문제에서는 extend()를 통해 'pasta' 값을 붙였는데, 이 경우 p, a, s, t, a로 나뉘어 favorite_foods에 추가된다. 따라서 remove('pasta')를 실시했을 때 해당 항목은 favorite_foods에 존재하지 않고, ValueError가 발생하게 된다.