



## 07장 함수

# 함수 1/5



항목	설명	예시
함수의 구조	def 키워드를 사용하여 함수 정의 후 실행 가능	<pre>def 함수명(입력 인수):     &lt;수행할 문장1&gt;     &lt;수행할 문장2&gt;     ...</pre>
일반적인 함수	입력값을 받아 처리하고 결과값을 반환하는 함수	<pre>def sum(a, b):     result = a + b     return result &gt;&gt;&gt; a = sum(3, 4) &gt;&gt;&gt; print(a) 7</pre>
입력값이 없는 함수	매개변수가 없이 실행되는 함수	<pre>&gt;&gt;&gt; def say(): ...     return 'Hi' ... &gt;&gt;&gt; &gt;&gt;&gt; a = say() &gt;&gt;&gt; print(a) Hi</pre>

# 함수 2/5



항목	설명	예시
결과값이 없는 함수	return 없이 실행만 하는 함수	<pre>&gt;&gt;&gt; def sum(a, b): ...     print("%d, %d의 합은 %d입니다." % (a, b, a+b)) ... &gt;&gt;&gt; &gt;&gt;&gt; sum(3, 4) 3, 4의 합은 7입니다.</pre>
입력값도 결과값도 없는 함수	매개변수도 없고 return도 없는 함수	<pre>&gt;&gt;&gt; def say(): ...     print('Hi') ... &gt;&gt;&gt; &gt;&gt;&gt; say() Hi</pre>

# 함수 3/5



항목	설명	예시
여러 개의 입력값	*args를 사용하여 여러 개의 입력을 받을 수 있음	<pre>&gt;&gt;&gt; def sum_many(*args): ...     sum = 0 ...     for i in args: ...         sum = sum + i ...     return sum ... &gt;&gt;&gt;</pre>
함수의 결과값은 언제나 하나이다	return을 사용하면 여러 값도 튜플로 하나의 값으로 반환됨	<pre>&gt;&gt;&gt; def sum_and_mul(a,b): ...     return a+b, a*b &gt;&gt;&gt; result = sum_and_mul(3,4) result = (7, 12)</pre>

# 함수 4/5



항목	설명	예시
매개 변수에 초깃값 미리 설정하기	기본값을 지정해 특정 인자 없이도 호출 가능	<pre>def say_myself(name, old, man=True):     print("나의 이름은 %s 입니다." % name)     print("나이는 %d살입니다." % old)     if man:         print("남자입니다.")     else:         print("여자입니다.") say_myself("박응용", 27) say_myself("박응용", 27, True)</pre>
함수 매개 변수에 초깃값을 설정할 때 주의할 사항	기본값 매개변수는 반드시 일반 매개변수 뒤에 위치해야 함	<pre>def example(a, b=10): return a + b (○) def example(a=10, b): return a + b (✗)</pre>
함수 안에서 선언된 변수의 효력 범위	함수 내부에서 선언된 변수는 함수 내에서만 사용 가능 (지역 변수)	<pre>a = 1 def vartest(a):     a = a + 1  vartest(a) print(a)</pre>

# 함수 5/5



항목	설명	예시
함수 안에서 함수 밖의 변수를 변경하는 방법	global 키워드를 사용하거나 return으로 값을 반환해야 함	<pre>a = 1 def vartest(a):     a = a + 1     return a  a = vartest(a) print(a) &gt;&gt;&gt; &gt;&gt;&gt; &gt;&gt;&gt; a = 1 def vartest():     global a     a = a + 1  vartest() print(a)</pre>

# 형성평가 1/3



## 문제 1

함수의 구조를 올바르게 설명한 것은?

1. 함수는 ``def`` 키워드 없이 선언해야 한다.
2. 함수는 반드시 입력값이 있어야 한다.
3. 함수는 ``def`` 키워드를 사용하여 정의하며, ``return``을 통해 값을 반환할 수 있다.
4. 함수 내부에서 선언한 변수는 항상 전역 변수이다.

## 문제 2

결과값이 없는 함수의 특징은?

1. ``return`` 없이 실행만 하고 종료된다.
2. 반드시 ``return None``을 포함해야 한다.
3. 입력값도 항상 없어야 한다.
4. ``print()``를 사용하면 결과값이 반환된다.

# 형성평가 2/3



## 문제 3

여러 개의 입력값을 받을 수 있는 방법은?

1. 함수의 매개변수를 여러 개 선언해야 한다.
2. `*args`를 사용하여 여러 개의 입력을 받을 수 있다.
3. 리스트만 매개변수로 사용해야 한다.
4. 여러 개의 `return`을 사용해야 한다.

## 문제 4

함수의 결과값은 언제나 하나인 이유는?

1. `return`을 여러 개 사용할 수 없기 때문이다.
2. `return`을 여러 개 사용하면 에러가 발생한다.
3. 여러 개의 값을 반환해도 튜플 형태로 하나의 값으로 처리된다.
4. `print()`를 사용하면 여러 개의 값을 반환할 수 있다.



# 형성평가 3/3



## 문제 5

함수의 매개변수에 초깃값을 설정할 때 주의할 사항은?

1. 기본값을 가진 매개변수는 반드시 일반 매개변수 뒤에 와야 한다.
2. 기본값을 가진 매개변수는 함수 정의에서 필수로 작성해야 한다.
3. 기본값을 설정하면 해당 매개변수는 항상 값을 가져야 한다.
4. 초깃값을 설정하면 ``return``이 필요 없다.

## 문제 6

함수 안에서 선언된 변수의 효력 범위에 대한 설명으로 옳은 것은?

1. 함수 내부에서 선언된 변수는 함수 밖에서도 사용 가능하다.
2. 함수 내부에서 선언된 변수는 지역 변수이며, 함수 외부에서는 접근할 수 없다.
3. ``return``을 사용하면 지역 변수를 전역 변수로 변경할 수 있다.
4. 함수 내부에서 선언된 변수는 ``print()``를 사용하면 전역 변수가 된다.