

Chapter 10

함수 기초 다지기



목차

1. 이 장에서 만들 프로그램
2. 함수란?
3. 함수는 왜 사용할까?
4. 함수 정의하기
5. 함수 호출하기
6. 함수 내에서 또 다른 함수 호출하기

실전 예제 1 다국어 인사말 프로그램

실전 예제 2 계산기 프로그램

학습목표

- 함수의 개념을 이해합니다.
- 함수를 사용하는 이유를 알아봅니다.
- 함수를 정의하고 호출하는 방법을 알아봅니다.
- `pass`를 이용해서 실행부가 생략된 함수의 정의 방법을 알아봅니다.
- 함수 내에서 다른 함수를 호출하는 방법을 알아봅니다.

Section 01

이 장에서 만들
프로그램



1. 다국어 인사말 프로그램

- 출신 국가에 따른 인사말을 출력하는 프로그램
- 한국, USA, Japan 중 하나를 선택하면 해당 국가의 인사말이 출력됨

where are you from? 1.한국, 2.USA, 3.Japan 1
안녕.

where are you from? 1.한국, 2.USA, 3.Japan 2
Hello.



2. 계산기 프로그램

- 숫자 2개를 입력하고 연산자를 선택하면 연산 결과를 출력하는 프로그램

숫자를 입력하세요. 5

연산자를 선택하세요. 1. 덧셈, 2. 뺄셈, 3. 곱셈, 4. 나눗셈 3

숫자를 입력하세요. 3

곱셈 결과 : 15.0

숫자를 입력하세요. 3

연산자를 선택하세요. 1. 덧셈, 2. 뺄셈, 3. 곱셈, 4. 나눗셈 2

숫자를 입력하세요. 50

뺄셈 결과 : -47.0

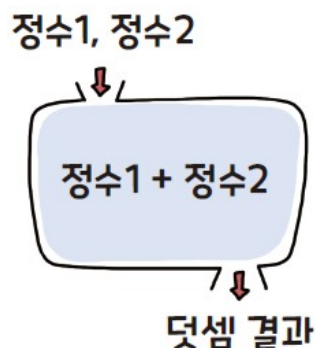
Section 02

함수란?

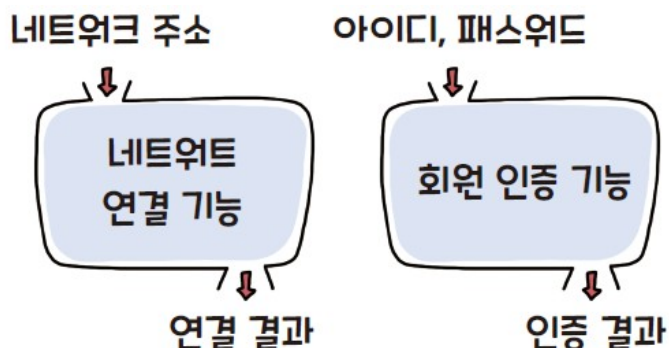


■ 프로그래밍의 함수

- 수학의 함수와 동일하게 값을 넣어주면 특정 기능을 수행한 연산 결과를 출력함
- 특정 기능이란 덧셈 같은 비교적 간단한 연산부터 네트워크 연결, 회원 인증, 메일 발송과 같이 복잡하고 어려운 작업까지 모두 포함함
- 즉 함수란 특정 기능을 하는 코드를 묶어 놓은 것이고 사용자는 함수에 값을 넣어 결과를 얻는 것을 의미함



(a) 비교적 간단한 함수



(b) 복잡한 함수

그림 10-3 프로그램에서 함수 사용의 예



■ 함수의 종류

- 내장 함수 : 파이썬에서 기본으로 제공하는 함수로 파이썬을 설치하면 바로 사용할 수 있음
- 사용자 함수 : 사용자가 필요할 때 직접 만들어 사용함



(a) 내장 함수



(b) 사용자 함수

그림 10-5 파이썬 함수의 종류



■ 대표적인 내장 함수

- print(), len() 등

코드 10-1

ch10_01.py

```
01  userName = 'Hong gil dong'
02
03  print('이름 : ', userName)
04  print('이름의 길이 : ', len(userName))
```

이름 : Hong gil dong

이름의 길이 : 13



■ 사용자 함수

- 사용자가 직접 만들어 사용함
- 개발자가 특정 함수가 필요하다고 판단할 때 직접 함수를 만들어 프로그램에 삽입함

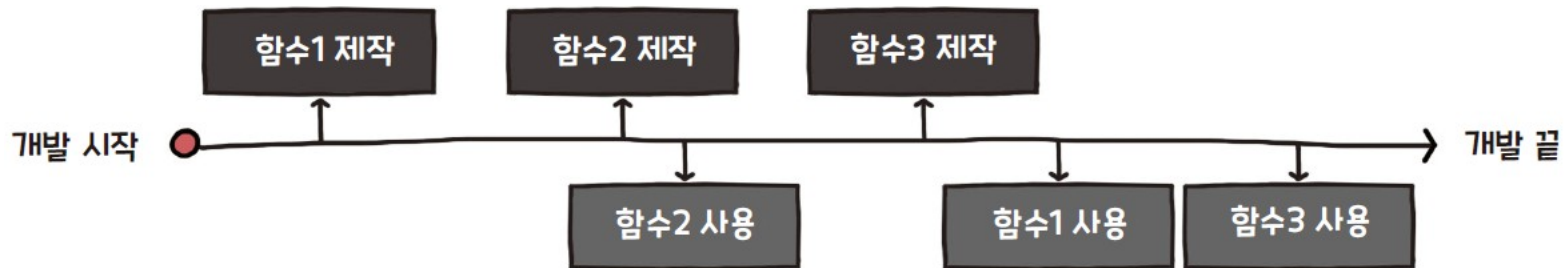


그림 10-6 필요할 때마다 만들어 사용하는 함수

Section 03

함수는 왜 사용할까?



■ 코드 재사용

- 10명의 스키점프 선수가 점프할 때마다 점프 거리를 입력 받아 1등부터 10등까지 순위를 결정하는 프로그램을 만든다고 가정함
- 이 프로그램에서 점프 거리를 등록하고 순위를 계산하는 코드가 10번 중복됨
- 이렇게 매번 중복되는 코드를 함수로 만들어 사용하면 프로그램의 코드 양이 줄어들고 간결해짐

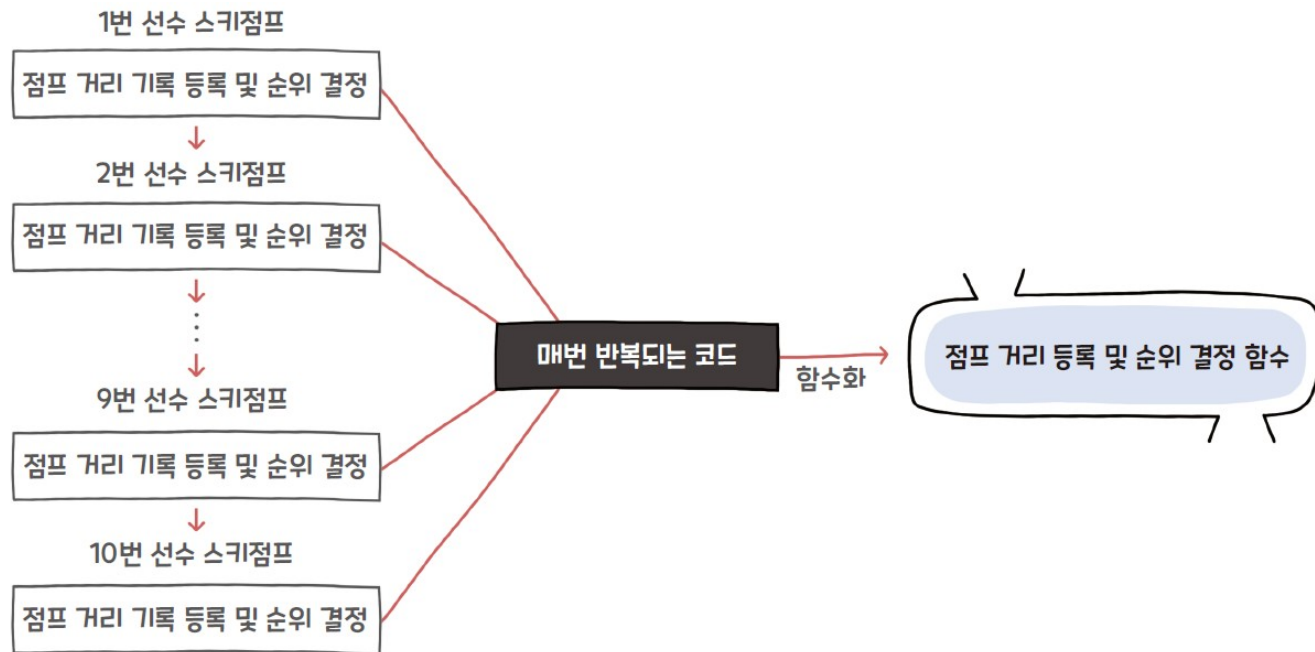


그림 10-7 매번 반복되는 코드를 함수로 정리



■ 모듈화

- 모듈은 특정 기능의 작은 프로그램을 뜻함
- 특정 기능이 함수로 모듈화되면 다른 프로그램에 쉽게 이식하여 사용할 수 있으며 그만큼 프로그램을 만드는 시간도 단축됨

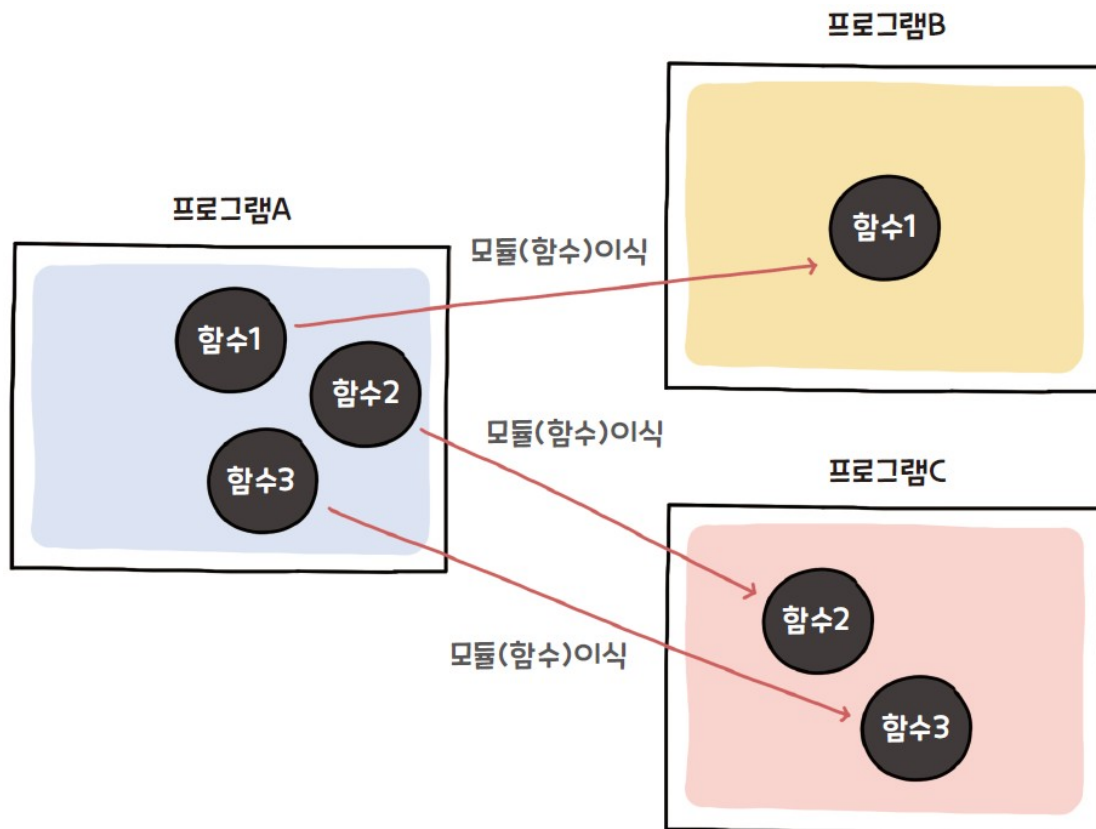


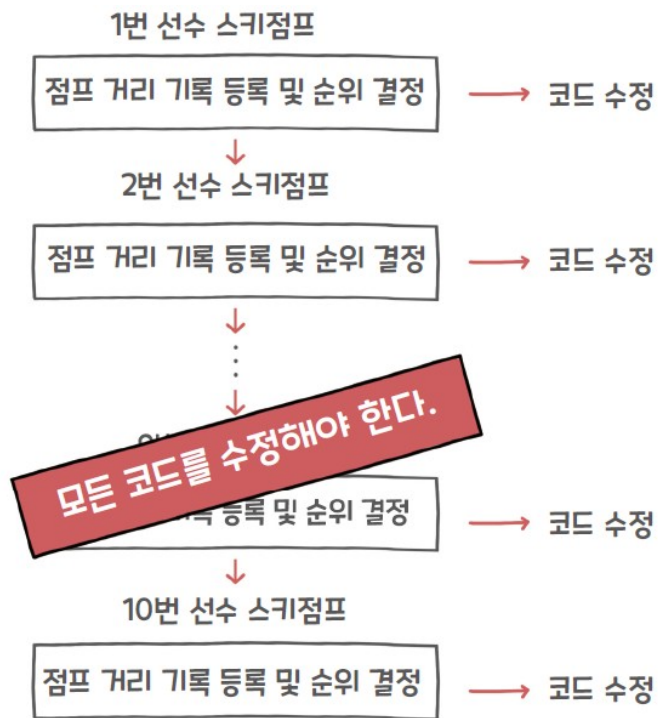
그림 10-8 함수를 여러 프로그램에 이식하여 사용

함수는 왜 사용할까?

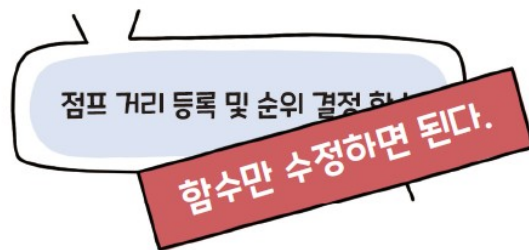


■ 코드 수정 용이

- 코드에 수정 사항이 발생했을 때 함수를 사용하면 함수만 수정하면 됨
- 만약 함수를 사용하지 않는다면 모든 코드를 수정해야 하는 불편함이 있고, 자칫 일부 코드가 수정되지 않아 프로그램에 심각한 오류가 발생할 수 있음



(a) 함수를 사용하지 않은 경우



(b) 함수를 사용한 경우



확인문제

1. 함수에 대한 설명으로 옳지 않은 것은 무엇인가?

- ① 함수는 비교적 간단한 연산을 할 때만 사용한다.
- ② 함수를 사용하면 코드를 재사용할 수 있다.
- ③ 함수는 내장 함수와 사용자 함수로 구분된다.
- ④ `len()` 함수는 문자열 또는 컨테이너 자료형의 길이를 반환한다.

2. 함수를 사용하는 이유로 옳지 않은 것은 무엇인가?

- ① 코드 재사용
- ② 모듈화
- ③ 코드 수정 용이
- ④ 프로그램 개발 수익 극대화

정답

1. ① 2. ④

Section 04

함수 정의하기



■ 함수 정의(function definition)

- 함수를 정의할 때는 def 키워드, 함수명, 콜론(:), 실행문으로 나누어 작성함

def 키워드 함수명 콜론

def greet() :

코드 블록을 위한
들여쓰기 { print('안녕하세요.')
 print('반갑습니다~.') } 실행문

그림 10-10 함수 정의 구문



■ def

- def는 definition의 약자로 ‘무언가를 정의한다.’는 뜻임
- 여기서는 함수를 정의한다는 의미로 사용함



■ 함수명

- def 키워드 다음에는 함수명을 명시함
 - 함수명을 지을 때의 유의 사항
- ① 내장 함수명과 동일하면 안 됨
 - print(), len() 등과 같이 기존의 내장 함수와 같은 이름은 사용이 불가능함
 - 이 경우 myPrint(), myLen()처럼 내장 함수 이름과 중복되지 않도록 함
 - ② 첫 글자는 주로 소문자로 시작함
 - 예를 들어 MyCalculator()는 첫 글자가 대문자이므로 권장하지 않음
 - 이 경우 myCalculator()처럼 첫 글자를 소문자로 수정하여 사용함
 - ③ 첫 글자로 숫자를 사용할 수 없음
 - 첫 글자 외에는 숫자를 사용해도 되지만 2myCalculator()처럼 숫자로 시작하는 함수명은 사용이 불가능함
 - ④ 특수문자는 사용할 수 없지만 언더바(_)는 사용 가능함
 - 특수 문자 하이픈(-)이 포함된 my-Calculator()는 사용 불가능함



■ 콜론(:)과 실행문

- 콜론(:)은 실행문(코드 블록)의 시작을 나타내는 것으로 앞에서 살펴본 if문, for문과 사용 방법이 같음
- 실행문 역시 들여쓰기로 구분함

```
def greet() :  
    print('안녕하세요.')  
    print('반갑습니다~.')  
    print('저는 홍길동입니다.')  
print('함수의 바깥 영역입니다.')
```

코드 블록 시작 ←

코드 블록 끝 ←

그림 10-11 실행문은 들여쓰기로 구분



■ 간단한 함수 3개를 정의하기

- 실행문은 코드를 완성하지 않고 어떤 기능을 하는지만 작성함
- 단위 변환 함수 : cm를 m로, g을 kg으로와 같이 단위를 변환하는 함수

```
def convertUnit():  
    # 이곳에 단위 변환 내용을 코딩
```

- 문자 메시지 발송 함수 : 문자 메시지를 보낼 때 사용하는 함수

```
def sendMessage():  
    # 이곳에 문자 메시지 내용을 코딩
```

- 거리 계산 함수 : 두 물건 사이의 거리가 얼마나 떨어져 있는지 계산하는 함수

```
def calculateDistance():  
    # 이곳에 거리 계산 내용을 코딩
```



문제 해결 10-1

온도센서 작동 시스템 만들기

ch10_sol_01.py

온도센서를 작동을 시작하고 멈추는 함수를 정의해봅시다. 함수명은 함수의 기능을 이해하기 좋도록 짓습니다.

```
01 def startTemperatureSensor():
02     print('온도센서 작동을 시작한다.')
03
04 def stopTemperatureSensor():
05     print('온도센서 작동을 중지한다.')
06
07 startTemperatureSensor()
08 stopTemperatureSensor()
```

온도센서 작동을 시작한다.
온도센서 작동을 중지한다.



확인문제

다음 중 함수가 올바르게 정의된 것은 무엇인가?

①

```
definition calculator():  
    print(3 + 4)
```

②

```
def calculator():  
    print(3 + 4)
```

③

```
def 3calculator():  
    print(3 + 4)
```

④

```
def calculator()  
    print(3 + 4)
```

정답

②

Section 05

함수 호출하기



■ 함수 호출

- 함수를 사용하는 것을 의미함
- 함수 호출은 함수 실행이 필요한 곳에서 함수 이름을 적으면 됨
- 인사 문구를 출력하는 greet() 함수를 정의하고 호출하기

코드 10-2

ch10_02.py

```
01 def greet():  
02     print('Hello.')  
03     print('Nice to meet you.')  
04  
05 greet()
```

함수 정의

함수 호출

```
Hello.  
Nice to meet you.
```



■ 함수 호출

- [코드 10-2]의 인사 문구를 세 번 출력하려면 함수를 세 번 호출하면 됨

코드 10-3

ch10_03.py

```
01 def greet():  
02     print('Hello.')  
03     print('Nice to meet you.')  
04  
05 greet()           # 첫 번째 함수 호출  
06 greet()           # 두 번째 함수 호출  
07 greet()           # 세 번째 함수 호출
```

함수 정의

Hello. — 첫 번째 함수 호출 결과

Nice to meet you.

Hello. — 두 번째 함수 호출 결과

Nice to meet you.

Hello. — 세 번째 함수 호출 결과

Nice to meet you.

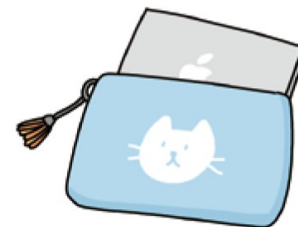


문제 해결 10-2

내 노트북은 몇 인치일까?

ch10_sol_02.py

고등학교 졸업 기념으로 노트북을 하나 장만했습니다. 노트북 사이즈에 꼭 맞는 파우치를 하나 구매하려고 하는데 사이즈 표에 인치로만 표시되어 있습니다. cm를 인치로 바꿔주는 함수를 만들어봅시다.



```
01 def convertUnit():
02     print(lengthCm, 'cm = ', lengthCm * 0.393701, 'inch')
03
04     lengthCm = float(input('길이를 입력하세요.(cm) '))
05     convertUnit()
```

길이를 입력하세요.(cm) 38

38.0 cm = 14.9606 inch

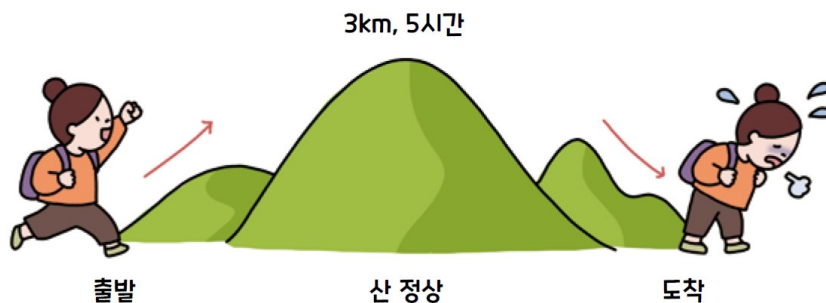


문제 해결 10-3

이동 거리를 계산하는 함수

ch10_sol_03.py

길동이는 5시간 동안 3km의 속도로 등산을 했습니다. 길동이가 등산한 시간과 속도를 입력하면 이동한 거리를 계산해주는 프로그램을 함수를 이용하여 만들어봅시다.



```
01 def calculateDistance():
02     print('이동 거리는', hourData * speedData, 'km 입니다.')
03
04     hourData = float(input('이동 시간을 입력하세요. '))
05     speedData = float(input('이동 속도를 입력하세요. '))
06     calculateDistance()
```

이동 시간을 입력하세요. 5
이동 속도를 입력하세요. 3
이동 거리는 15.0 km 입니다.



■ pass 키워드

- 함수에서 pass를 사용하여 아직 실행부가 정해지지 않은 함수를 정의함

코드 10-4

ch10_04.py

```
01 def getMemberInDatabase():
02     pass ● ————— 실행할 내용을 아직 정하지 못 함
03
04 def sendMemberIdByEmail():
05     pass ● ————— 실행할 내용을 아직 정하지 못 함
06
07 m_name = input('이름을 입력하세요. ')
08 m_mail = input('메일 주소를 입력하세요. ')
09
10 getMemberInDatabase()
11 sendMemberIdByEmail()
```

이름을 입력하세요. Hong gil dong

메일 주소를 입력하세요. honggildong@gmail.com



확인문제

1. 다음 프로그램의 실행 결과를 보고 빈칸을 채우시오.

```
01 def addCalculator():
02     print('첫 번째 숫자와 두 번째 숫자의 합은 ', ( ), '입니다.')
03
04     num1 = int(input('첫 번째 숫자를 입력하세요.'))
05     num2 = int(input('두 번째 숫자를 입력하세요.'))
06
07     
```

첫 번째 숫자를 입력하세요.5
두 번째 숫자를 입력하세요.9
첫 번째 숫자와 두 번째 숫자의 합은 14 입니다.

2. 다음 중 startMachine() 함수를 호출하는 코드로 옳은 것은 무엇인가?

```
def startMachine():
    print('기계를 작동한다.')
```

- | | |
|-------------------|----------------------|
| ① startMachine() | ② startMachine |
| ③ startMachine(): | ④ print('기계를 작동한다.') |

정답

1. num1+num2, addCalculator() 2. ①

Section 06

함수 내에서 또 다른 함수 호출하기



■ 함수 내부에서 또 다른 함수를 호출하기

코드 10-5

ch10_05.py

```
01 def fun1():
02     print('fun1 함수를 호출합니다!')
03
04 def fun2():
05     print('fun2 함수를 호출합니다!')
06
07 def fun3():
08     fun1()
09     fun2()
10     print('fun3 함수를 호출합니다!')
11
12 fun3()
```

fun1 함수를 호출합니다!
fun2 함수를 호출합니다!
fun3 함수를 호출합니다!



■ 함수 내부에서 또 다른 함수를 호출하기

- [코드 10-15]의 12행에서 fun3()을 호출하면 fun3() 내부에서는 fun1()과 fun2()를 호출함
- 따라서 fun3()을 호출하면 fun1()과 fun2()가 모두 호출됨

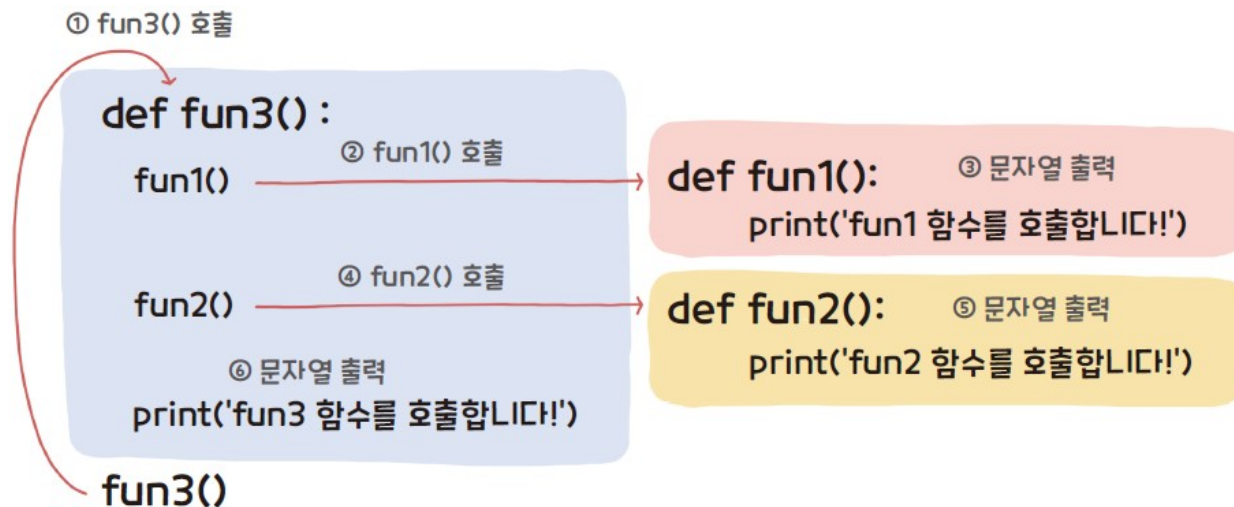


그림 10-12 함수 호출 및 실행 순서



확인문제

다음 프로그램 실행 결과로 옳은 것은 무엇인가?

```
def printMemberInfo():
    getMemberInDB()
    print('회원정보를 출력한다.')

def getMemberInDB():
    connectDB()
    selectMemberInDB()
    print('데이터베이스에서 회원정보를 가져온다.')

def connectDB():
    print('데이터베이스에 접속한다.')

def selectMemberInDB():
    print('데이터베이스에서 회원정보를 검색한다.')

printMemberInfo()
```

- ① 회원정보를 출력한다.
데이터베이스에서 회원정보를 가져온다.
데이터베이스에 접속한다.
데이터베이스에서 회원정보를 검색한다.
- ② 데이터베이스에 접속한다.
데이터베이스에서 회원정보를 검색한다.
데이터베이스에서 회원정보를 가져온다.
회원정보를 출력한다.
- ③ 회원 정보를 출력한다.
데이터베이스에서 회원정보를 가져온다.
데이터베이스에서 회원정보를 검색한다.
데이터베이스에 접속한다.
- ④ 회원정보를 출력한다.

정답

②

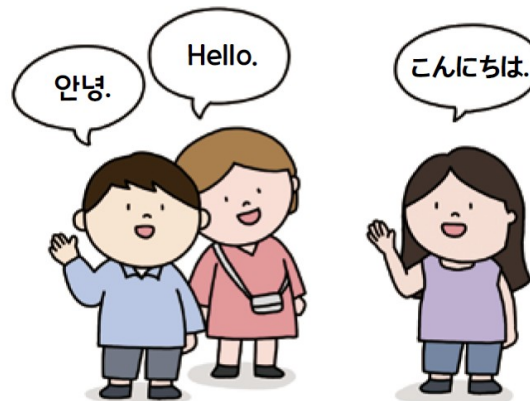
실전 예제



문제

출신 국가를 선택하면 해당하는 국가의 인사말이 출력되는 프로그램을 함수를 이용해서 만들어봅시다.

국가 선택	1. 한국	2. USA	3. Japan
인사말	안녕.	Hello.	こんにちは.



where are you from? 1.한국, 2.USA, 3.Japan 1
안녕.

where are you from? 1.한국, 2.USA, 3.Japan 2
Hello.



해결

ch10_appEx_01.py

```
01 def introKor():
02     print('안녕.')
03
04 def introEng():
05     print('Hello.')
06
07 def introJap():
08     print('こんにちは.')
09
10 selectNum = int(input('where are you from? 1.한국, 2.USA, 3.Japan '))
11
12 if(selectNum == 1):
13     introKor()
14 elif(selectNum == 2):
15     introEng()
16 elif(selectNum == 3):
17     introJap()
18 else:
19     introEng()
```



문제

사용자가 숫자 2개를 입력하고 연산자를 선택하면 연산결과가 출력되는 프로그램을 함수를 이용해서 만들어봅시다.



숫자를 입력하세요. 5

연산자를 선택하세요. 1.덧셈, 2.뺄셈, 3.곱셈, 4.나눗셈 3

숫자를 입력하세요. 3

곱셈 결과 : 15.0

숫자를 입력하세요. 3

연산자를 선택하세요. 1.덧셈, 2.뺄셈, 3.곱셈, 4.나눗셈 2

숫자를 입력하세요. 50

뺄셈 결과 : -47.0



해결

ch10_appEx_02.py

```
01 def add():          # 덧셈
02     print('덧셈 결과 : ', inputNumber1 + inputNumber2)
03
04 def sub():          # 뺄셈
05     print('뺄셈 결과 : ', inputNumber1 - inputNumber2)
06
07 def mul():          # 곱셈
08     print('곱셈 결과 : ', inputNumber1 * inputNumber2)
09
10 def div():          # 나눗셈
11     print('나눗셈 결과 : ', inputNumber1 / inputNumber2)
12
13 def calculator():   # 계산기
14     if(selectOperator == 1):
15         add()
```



```
16     elif(selectOperator == 2):
17         sub()
18     elif(selectOperator == 3):
19         mul()
20     elif(selectOperator == 4):
21         div()
22
23     inputNumber1 = float(input('숫자를 입력하세요. '))
24     selectOperator = int(input('연산자를 선택하세요. 1.덧셈, 2.뺄셈, 3.곱셈, 4.나눗셈 '))
25     inputNumber2 = float(input('숫자를 입력하세요. '))
26
27     calculator()
```

Thank you!