

- 학/습/목/표**
- 함수의 개념을 이해할 수 있다.
 - 함수를 활용한 프로그램을 작성할 수 있다.
 - 전역 변수와 지역 변수를 이해할 수 있다.

핵/심/용/어 모듈, 함수, 매개 변수, 전역 변수, 지역 변수

소프트웨어를 개발하다 보면, 예전에 만들었던 프로그램 코드의 일부를 다시 사용해야 하는 경우가 있다. 혹은 다른 사람들과의 협력을 위해 프로그램 코드의 복잡한 부분을 숨겨 누구나 쉽게 이해할 수 있도록 코드를 작성해야 하는 경우도 있다. 이러한 경우에는 프로그램 코드를 어떤 방법으로 작성해야 할까?



? 누구나 쉽게 이해할 수 있게 프로그램 코드를 작성하는 방법에는 어떤 것들이 있을까?

1 모듈과 모듈화

어떤 문제를 해결하기 위해서는 먼저 문제를 정확하게 이해하고 쉽게 해결할 수 있는 작은 문제들로 분해해야 한다. 소프트웨어를 개발할 때에도 마찬가지이다. 개발하려는 소프트웨어의 목적과 기능을 분명하게 정의한 다음, 목적을 이루는 데 필요한 작은 기능들로 분해하여 하나씩 해결함으로써 소프트웨어를 개발할 수 있다.

예를 들어 우리가 자주 사용하는 ‘문서 작성 소프트웨어(워드프로세서)’를 개발하는 경우를 생각해 보자. 문서 작성 소프트웨어의 목적과 필요한 기능은 다음과 같이 정리할 수 있다.

소프트웨어의 목적	• 문서의 작성과 편집을 쉽게 한다.
필요한 기능	• 문서를 작성하고 편집하는 기능 • 문서를 인쇄하는 기능 • 문서를 저장하고 불러오는 기능

•모듈

소프트웨어를 기능별로 분할한 논리적인 부분이다.

•모듈화의 장점

프로그램 코드가 단순해지고 소프트웨어 제작에 걸리는 시간이 단축된다. 또, 기능을 향상시킬 때에는 필요한 모듈만 성능이 더 좋은 모듈로 교체하면 된다.

이렇게 나누어진 각각의 논리적인 기능을 **모듈**이라고 하는데, 이 모듈들은 서로 독립적이다. 즉 ‘문서를 작성하고 편집하는 기능’과 ‘문서를 인쇄하는 기능’, ‘문서를 저장하고 불러오는 기능’은 서로 간에 아무런 영향을 주지 않는다. 이처럼 소프트웨어를 기능별로 나누어진 여러 개의 독립된 모듈로 만드는 것을 ‘모듈화’라고 한다.

소프트웨어를 모듈로 만들면, 모듈을 교체하는 것만으로 소프트웨어의 성능을 높이거나 새로운 기능을 추가할 수 있다. 또 오류가 발견될 경우 해당 모듈만 수정하면 되기 때문에 소프트웨어의 유지·관리가 쉬워진다.

2 함수

함수는 소프트웨어의 개발 과정에서 자주 사용되는 기능을 수행할 수 있도록 미리 만들어 둔 모듈로, 각각의 함수는 고유한 이름과 기능을 가진다. 따라서 함수의 이름을 호출하는 것만으로 그 함수가 가진 기능을 실행할 수 있다.

예를 들어 파이선 언어는 어떤 결과를 모니터에 출력할 때 `print()` 함수를 사용하는데, 이 함수는 결과를 모니터에 보여 주기까지의 복잡한 명령어들을 처리하는 역할을 한다. 따라서 프로그램을 만드는 사람은 복잡한 명령어들을 알지 못해도 `print()` 함수를 호출하는 것만으로 모니터에 결과를 출력할 수 있다. 밑변과 높이의 값을 입력하면 삼각형의 넓이를 계산해 주는 함수를 만들어 보자.

프로그램 코드

```

1 def triangle(w, h) :                # triangle 함수 정의
2     t=(w*h)/2                       # 밑변과 높이의 값을 받아서 넓이 계산
3     return t                        # 계산 결과를 반환
4
5 width=input("삼각형의 밑변:")
6 width=int(width)
7 height=input("삼각형의 높이:")
8 height=int(height)
9 area=triangle(width, height)        # triangle() 함수를 호출하고, 매개 변수를 전달
10 print("삼각형의 넓이:", area)

```

def	define의 줄임말로, 함수를 만들기 위해 사용하는 파이선 명령어이다.
triangle	새로 만드는 함수의 이름이다.
w, h	함수가 계산에 사용할 수 있도록 넘겨주는 자료로, '매개 변수'라고 한다.
t=(w*h)/2	매개 변수로 받은 w와 h를 이용해서 삼각형의 넓이를 구하고, 그 결과를 변수 t에 저장한다.
return t	변수 t에 들어 있는 값을 원래의 위치에 되돌려 주는 것으로, '반환값'이라고 한다.
area=triangle (width, height)	triangle() 함수를 호출해서 기능을 실행한다. input()을 통해 사용자로부터 입력받은 width와 height의 값을 triangle() 함수에게 매개 변수로 넘겨주고, 계산된 결과를 반환값으로 받아서 변수 area에 저장한다.

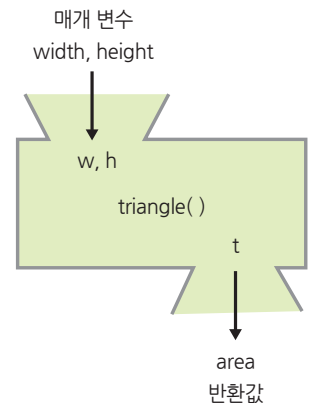
• 함수 이름

함수의 기능이 무엇인지 알 수 있도록 관련된 이름을 가져야 한다.



• 함수 호출

함수가 실행되도록 이름을 써 주는 것을 말한다.



• 매개 변수와 반환값

경우에 따라 매개 변수와 반환값이 없는 경우도 있다.

예) `print("안녕하세요!")`

매개 변수는 함수가 기능을 수행하는 데 필요한 자료를 함수 밖에서 안으로 전달해 주는 변수이다. triangle() 함수의 경우, 함수 밖에서 사용되는 두 변수 width와 height의 값이 함수 안의 매개 변수 w와 h에 전달되어 사용된다. 매개 변수는 값이 복사되는 것이기 때문에 함수 안에서 w와 h의 값이 변하더라도 함수 밖의 변수 width와 height의 값은 변하지 않는다.

3 함수의 종류와 특징

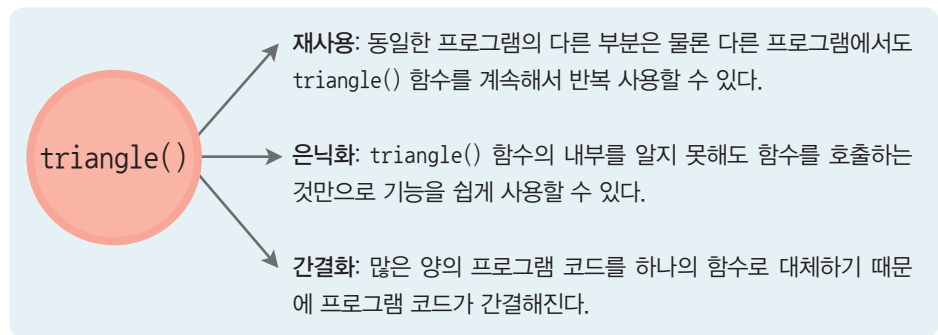
함수에는 프로그래밍 언어가 기본적으로 제공하는 ‘내장 함수’와 프로그래머가 직접 만들어 사용하는 ‘사용자 정의 함수’가 있다.

● 파이선 언어의 함수 이름을 정하는 규칙

- 한글, 영문, 숫자, 밑줄만 사용한다.
- 첫 문자에 숫자를 사용할 수 없다.
- 대소문자를 구분한다.
- if, for, while 등 파이선에서 미리 정의된 예약어는 사용할 수 없다.
- 공백은 사용할 수 없다.

구분	내장 함수	사용자 정의 함수
설명	<ul style="list-style-type: none"> • 프로그래밍 언어가 기본적으로 제공하는 함수이다. • 준비 없이 바로 사용할 수 있다. • 함수의 이름이 고정되어서 변경할 수 없다. 	<ul style="list-style-type: none"> • 프로그래머가 필요할 때 직접 만들어 사용하는 함수이다. • 만들기 전에는 사용할 수 없다. • 이름을 마음대로 정해서 사용할 수 있다.
예	print(), input()	triangle()

프로그램을 개발하다 보면 내장 함수만으로는 한계가 있기 때문에, 프로그래머가 직접 필요한 기능을 정의하고 그 기능에 맞는 함수를 설계·제작해서 사용해야 한다. 이렇게 만들어지는 함수를 ‘사용자 정의 함수’라 하며, triangle() 함수가 여기에 해당한다. 사용자 정의 함수를 사용하면 다음과 같은 장점이 있다.



이처럼 함수를 사용하면, 프로그램 코드가 짧아져 이해하기가 쉽기 때문에 여러 명이 협력 작업을 하는 데에도 유리하다. 또 프로그램을 개발할 때 처음부터 끝까지 모든 것을 직접 만들지 않고 다른 사람이 만들어 둔 함수를 사용해서 쉽고 빠르게 프로그램을 만들 수 있다. 따라서 프로그램을 만들 때에는 함수를 만들어 사용하는 것이 좋다.

● 함수 라이브러리

프로그램을 쉽고 빠르게 개발할 수 있도록 다양한 함수를 만들어 모아 놓은 것을 말한다.

▶ 시(hour) 단위로 입력된 시간을 초(second) 단위로 변환하는 기능의 함수를 만들어 보자.

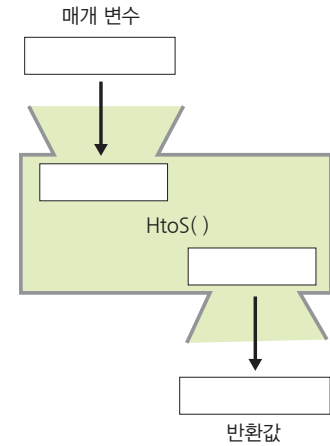
1 계산에 필요한 매개 변수와 계산된 반환값에 각각 무엇이 필요한지 오른쪽 빈칸을 채워 보자.

2 다음 빈 곳을 채워 프로그램 코드를 완성해 보자.

프로그램 코드

```

1 def _____ :
2     s=_____
3     return _____
4
5 hour=input("변환하려는 시간(h):")
6 hour=int(hour)
7 second= _____
8 print(hour, "시간은", second, "초입니다.")
    
```



4 전역 변수와 지역 변수

함수는 변수를 이용한 여러 가지 계산을 수행하게 되는데, 이때 사용되는 변수는 전역 변수와 지역 변수로 구분할 수 있다. **전역 변수**는 프로그램 전체에 걸쳐 사용할 수 있는 변수이고, **지역 변수**는 함수와 같은 특정 영역 안에서만 사용할 수 있는 변수이다. 다음 코드를 통해 전역 변수와 지역 변수에 대해 알아보자.

•매개 변수와 지역 변수

함수에 전달된 매개 변수는 지역 변수에 해당하기 때문에 함수 안에서만 사용할 수 있다.

프로그램 코드

```

1 def function1() :
2     result1=num1+num2
3     print("함수 안에서 결과 출력", result1)
4
5 def function2() :
6     result2=num1-num2
7     print("함수 안에서 결과 출력", result2)
8
9 num1=50                                # 전역 변수 num1
10 num2=20                                # 전역 변수 num2
11
12 function1()                            # function1 실행, 70 정상 출력
13 function2()                            # function2 실행, 30 정상 출력
14
15 print("함수 밖에서 결과 출력", result1) # 실행 오류
16 print("함수 밖에서 결과 출력", result2) # 실행 오류
    
```

앞의 프로그램 코드를 보면, 함수 밖에서 만들어진 변수 num1과 num2는 전역 변수이기 때문에 두 함수 function1()과 function2() 안에서도 사용할 수 있다. 하지만 function1()의 result1 변수와 function2()의 result2 변수는 함수 안에서 만들어진 지역 변수이기 때문에 함수 안에서만 사용할 수 있고, 함수 밖의 프로그램에서는 사용할 수 없다.

따라서 앞의 프로그램을 실행하면, function1()과 function2() 안에서는 정상적으로 70과 30이 출력되지만, 함수 밖에서는 result1과 result2의 출력에서 오류가 발생한다.

만약 함수 안에서 계산한 result1과 result2를 함수 밖에서도 사용하고 싶다면, 두 변수를 전역 변수로 선언해야 한다. 앞의 프로그램 코드를 다음과 같이 고치면 result1과 result2가 전역 변수가 되어 프로그램이 오류 없이 실행된다.

프로그램 코드

```

1 def function1() :
2     global result1                # result1을 전역 변수로 선언
3     result1=num1+num2
4     print("함수 안에서 결과 출력", result1)
5
6 def function2() :
7     global result2                # result2를 전역 변수로 선언
8     result2=num1-num2
9     print("함수 안에서 결과 출력", result2)
10
11 num1=50                          # 전역 변수 num1
12 num2=20                          # 전역 변수 num2
13
14 function1()                      # function1 실행, 70 정상 출력
15 function2()                      # function2 실행, 30 정상 출력
16
17 print("함수 밖에서 결과 출력", result1)    # 70 정상 출력
18 print("함수 밖에서 결과 출력", result2)    # 30 정상 출력

```

전역 변수는 일반적으로 프로그램 전반에 걸쳐 사용되는 값을 저장할 때 사용한다. 예를 들어 비행기 슈팅 게임을 개발할 때 플레이어의 남은 비행기 대수나 플레이어가 얻은 점수 등은 게임 전체에 걸쳐 사용되고, 게임이 끝날 때까지 유지되어야 하기 때문에 전역 변수로 설정하는 것이 좋다.

전역 변수를 이용하면 매개 변수를 사용하는 번거로움이 없어지기 때문에 편리하지만, 프로그램의 규모가 커지면 변수 관리에 어려움이 생겨 오류가 발생할 가능성이 커진다. 따라서 전역 변수는 꼭 필요한 곳에서만 사용하는 것이 좋다.

파이선 명령어

global

함수 안에서 만들어진 변수가 전역 변수로 사용될 수 있도록 선언하는 파이선 명령어이다.

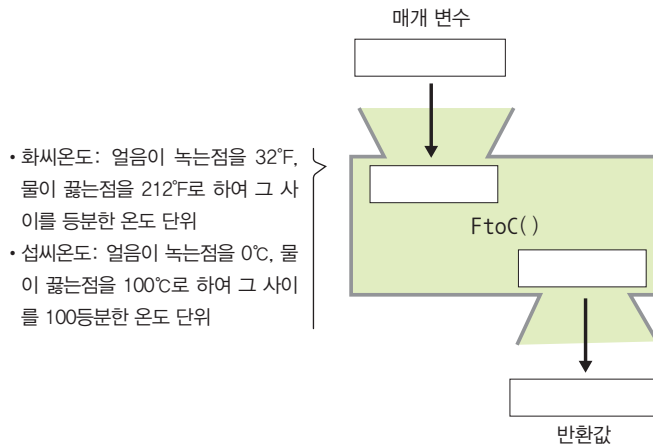
화씨온도를 섭씨온도로 변환하기

분석하기 + 설계하기 + 프로그래밍하기

- 민수는 미국 여행을 갔는데, 텔레비전 일기 예보에서 오늘의 기온을 화씨온도로 알려 주고 있다. 화씨온도를 섭씨온도로 변환하는 프로그램을 만들어 보자.



- 매개 변수와 반환값에는 각각 무엇이 필요한지 다음 빈칸을 채워 보자.



- 화씨온도를 섭씨온도로 변환하는 계산 공식을 찾아 적어 보자.

- 프로그램 코드를 작성해 보자.

프로그램 코드

- 위 프로그램 코드에 섭씨온도를 화씨온도로 변경하는 함수(CtoF)를 추가해 보자.