



Data Analytics Based Python

SECT8. 함수 (Function)

BigData Science Practical Training Course
Kim Jin Soo



- ◆ 함수 정의하기
- ◆ 기본 인자 값 활용하기, Default Argument Value
- ◆ 여러 개의 인자 값 및 키워드 인자 활용하기, Keyword Arguments
- ◆ 가변 인자 리스트 활용하기, Arbitrary Argument Lists
- ◆ 언패킹 인자 리스트 활용하기, Unpacking Argument Lists
- ◆ 변수의 유효 범위, Scope
- ◆ 문서화를 위한 문자열 활용하기, Documentation Strings (docstring)



❖ 함수란 ?

- 소스코드를 사용하여 일련의 행동을 기술한 것
- 함수명, 인자값, 반환값 세가지 기본 개념 이해 cf. Review Sect03

❖ 함수의 정의

- 인자값, 반환값 없는 함수
- 인자값은 있으나, 반환값은 없는 함수
- 인자값, 반환값 모두 있는 함수
- 반환값을 표기하지 않는 함수의 반환값, None



❖ Default Argument Value

- range() 함수
 - 인자 값에 따라 다양한 형태의 숫자로 이루어진 리스트를 반환하는 함수
 - 1개 이상의 인자값을 반드시 가져야 한다.
 - 인자값이 1개인 경우, 시작값은 0으로 기본 설정
 - 인자값이 2개인 경우, 범위를 설정
 - 인자값이 3개인 경우,
 - 첫 번째 인자값은 시작값
 - 두 번째 인자값은 마지막값보다 하나 큰 값
 - 세 번째 인자값은 스텝



❖ 함수의 인자값이 여러 개 있는 경우

- 함수의 인자값에는 함수 호출 시 반드시 대입해야 한다.
 - 대입하지 않은 경우에 에러가 발생하는 '필수 required 인자값'
 - 값을 대입하지 않은 경우에는 기본값을 할당하는 '옵션 optional 인자값'

❖ 파이썬에는 두 타입의 인자값이 공존하는 형태의 함수 가능

- positional or keyword 라고 부른다.
- positional, 위치 인자 값
 - 인자값을 집어 넣는 순서가 중요한 필수 인자 값
- keyword, 키워드 인자 값
 - 인자값을 대입할 때 매개 변수명이 키워드로 들어간다.
 - 기본값을 가지고 있는 인자 값
- 파이썬에서는
 - 위치 인자 값을 키워드 인자값처럼 호출하여 사용할 수도 있고,
 - 키워드 인자값을 위치 인자 값처럼 호출하여 사용할 수도 있다.



❖ 하나의 변수에 값이 여러 개가 들어가는 경우

- 1개의 변수에 여러 개의 값이 들어가 있는 튜플형 데이터나 사전형 데이터를 인자값으로 집어 넣는 경우는 흔하게 사용된다.
- 인자값 여러 개를 하나의 변수에 담아서 전달하고자 한다면 별표기호 (*)를 활용하면 된다.
 - 별표기호 한 개 (*)는 튜플형 데이터를 대입한다는 의미
 - 별표기호 두 개 (**)는 키와 값의 쌍으로 이루어진 사전형 데이터를 대입한다는 의미
- 참고로, C나 C++에서 변수 앞에 붙는 별표기호는 포인터를 의미하는 것과 조금 다른 의미이다.



❖ 열거형에는 리스트형도 있는데, 튜플형을 쓰는 이유가 뭐죠?

- 튜플은 값을 변경할 수 없는(immutable) 데이터형이기 때문
- 함수 호출 시 사용한 인자값을 추후에 변경하지 못하게 제약을 거는 용도로 사용한다 → 함수 안에서는 변경이 불가
- 실제로 이런 제약 사항은 개발 시에 발생할 수 있는 혼돈을 줄여준다.
- 만약, 리스트형으로 값을 입력 받아서 함수 블록문에서 일부 가공을 한 상태로 다른 함수의 인자값으로 넘겼다고 가정
 - 인자값을 넘겨 받은 함수는 이 리스트형 데이터가 최초의 값으로부터 변경 여부를 확인하는 절차가 필요



❖ 열거형 데이터를 함수 내에서 각 항목이 분리되어 사용하는 사례

- 두 개의 인자값이 하나의 리스트형 변수에 들어가 있는 경우에는 별표 기호(*)를 활용할 수 있다.
- 파이썬에서의 별표기호(*)는 데이터를 패킹할 지, 언패킹 할지를 결정하는 역할을 하는 것을 확인할 수 있다.
- 사전형을 위한 언패킹은 별표 기호 2개(**)를 사용하면 된다.



❖ 함수 내 변수들의 유효 범위에 대해 알아보자.

- 유효범위, Scope란?
 - 대부분의 프로그래밍 언어에서는 변수를 사용할 때 해당 변수가 영향을 미치는 영역이 존재한다.
 - 가령, 함수에 선언한 변수는 일반적으로 함수 내에서만 사용이 가능하며, 함수 밖에서는 사용이 제약이 있다.
- 전역변수, Global Variable이란?
 - 어떤 함수 이든지 함수 밖에서 선언하여 사용할 수 있는 함수

Tip. 전역변수 vs 지역변수



❖ 전역변수를 사용하는 것이 좋나요? 아니면 지역변수를 사용하는 것이 좋나요?

- 정확한 답은 둘 다 적당한 목적에 의해서 명확하게 구분해서 사용해야 한다는 것이다.
- 만약 변수의 값이 변경되는 경우라면 무조건 유효범위를 최소화 해야 하며 지역변수를 사용하는 것이 옳다.
- 전역변수는 프로그램이 실행하는 내내 절대로 변경되지 않는 값들이 들어가는 것이 좋다.
 - 예를 들면,
 - 실행 시에 이용된 운영체제 관련 설정 값
 - 학사정보시스템의 경우라면 시스템상으로 변하지 않는 대학교상세정보

✓소스코드의 변수는 가급적이면 미리 생성도 하지 말고, 사용되는 최소한의 영역에서 선언하여 최소한의 유효범위 내에서만 사용하는 습관을 들이는 것이 좋다.

✓전역변수와 지역변수의 이름을 동일하게 가져가는 것은 좋지 않다.



❖ docstring (= documentation string)

- 개발자들이 사용하거나 추후에 소스 코드를 확인하기 위해 필요한 상세 설명은 대개 함수 내에 기술하게 되어 있다.
- 특히, 고품질의 소스 코드에는 소스 코드만으로는 설명하기 어려운 내용들이 잘 기술되어 있어야만 한다.
- 이런 내용을 함수 내에 기술하기 위해서 여러 줄로 이루어진 문자열을 활용하는데, 이를 영문으로 'documentation string' 이라고 한다.
- 여러 줄을 기술할 경우, 세 개의 쌍따옴표 기호(""" """)로 감싸져 있는 부분이 함수에 대한 설명
 - 첫 번째 줄에는 함수의 간단한 설명을 집어 넣는다.
 - 공백 줄 이후에 각 매개변수에 대한 설명을 한다.
 - `__doc__` 라는 변수를 통해 함수를 선언한 docstring 부분을 확인 가능
- 보통 여러 사람이 함께 사용하는 함수에는 함수에 대한 상세한 설명이 반드시 작성해야 한다.



- ❖ 함수 정의 시에 필수 혹은 옵션으로 대입하는 인자값을 표현하기 위한 다양한 매개 변수 표현 방법
- ❖ 함수에서 기본값을 설정하는 방법
- ❖ 정의된 순서대로 인자값이 매개변수와 매핑되는 위치인자값
- ❖ 매개변수명을 키워드로 하여 순서와는 상관없이 대입할 수 있는 키워드 인자값
- ❖ 변수의 유효범위를 통해 전역변수와 지역변수의 차이점
- ❖ 함수에 대한 상세설명을 docstring으로 기술하고 `__doc__`이라는 함수 내장변수를 통해 출력