ICS 214 IT Workshop III (Python)
IIIT Kottayam

Session 3 - Writing Effective Functions

Parameters, Arguments, Recursion, and Decorators

Instructor: Anmol Krishan Sachdeva

Wednesday | December 7, 2022

Anmol Krishan Sachdeva
Hybrid Cloud Architect, Google
MSc Advanced Computing
University of Bristol, United Kingdom
LinkedIn: greatdevaks
Twitter: @greatdevaks

- International Tech Speaker
- Distinguished Guest Lecturer
- Represented India at reputed International Hackathons
- Deep Learning Researcher
- 8+ International Publications
- Google, Microsoft, IBM, and HP Certified Professional
- ALL STACK DEVELOPER
- Mentor

Agenda

- Functions
- Arguments
- Recursion
- Passing Functions to Functions

Indian Institute of Information Technology Kottayam

Functions are like mini programs within our programs that allow us to break code into smaller units. This spares us from having to write duplicate code, which can introduce bugs. But writing effective functions requires making many decisions about naming, size, parameters, and complexity.

Al Sweigart

Beyond The Basic Stuff with Python

Built-in Functions

```
a = ['foo', 'bar', 'baz', 'qux']
len(a)
```

Indian Institute of Information Technology Kottayam

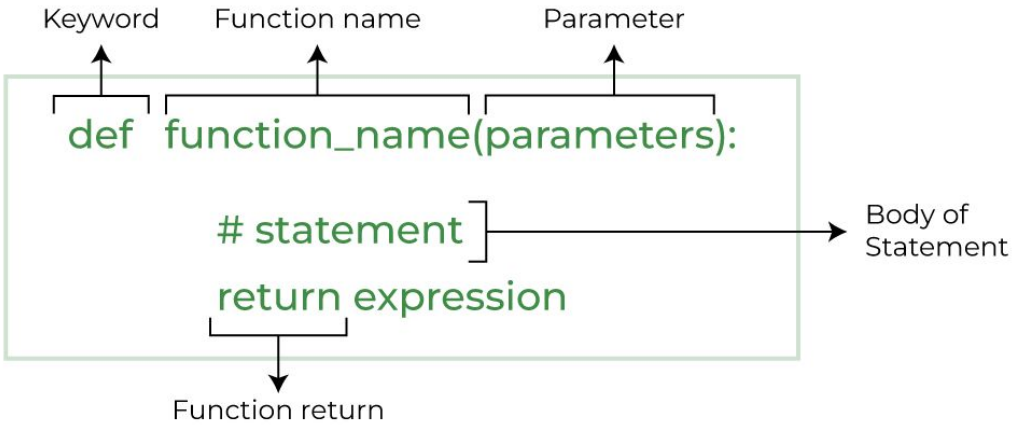
Subroutines?
Functions?
Procedures?
Methods?
Subprograms?

Almost all are user-defined functions but are called differently in different Programming Languages.

Understanding the Need for User-defined Functions

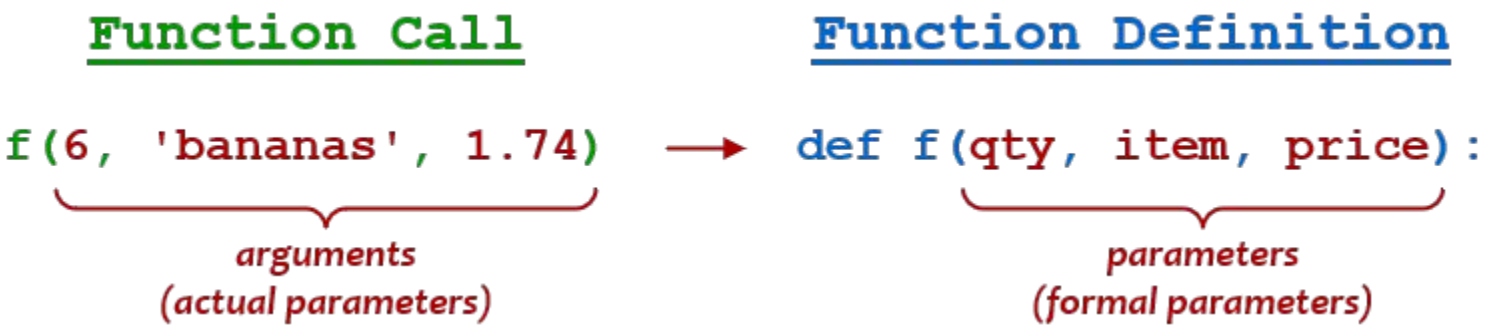
- Reusability - Don’t Repeat Yourself (DRY) principle
- Avoid Maintenance Nightmares
- Abstraction
- Modularity - break large chunks into smaller coherent units
- Namespace segregation

Structure of a Function



Source: GeeksForGeeks

Example



Source: Real Python

Arguments

- **Positional Arguments:** Required; order is important
- **Keyword Arguments:** Required; order is not important
- **Default/Optional Parameters:** Defaults set in the function definition
  - But with a Caveat for Mutable Objects like Lists
    - Use ***None***?
- ***Pass-by-Value/Pass-by-Reference?***
  - Object References - ***Pass-by-Assignment***
  - ***Changes in the Calling Environment - Side Effect?***
- **Variable-length Arguments**
  - Tuple Packing ***(*args)***
  - Dictionary Packing ***(**kwargs)***
  - Tuple Unpacking
  - Dictionary Unpacking
  - Multiple Unpacks

References

- [**Real Python**] Functions in Python
- [**Real Python**] Python Dictionaries
- [**Real Python**] Python Recursion