

Tolentino, Preetam D.
CS3C

Defining a Function

To define a function in Python, you use the “def” keyword followed by the function name and parentheses `def my_function(): print("Hello, world!")`.

Reasons for Using Functions

Functions allow you to break down your program into smaller, manageable parts. They make your code reusable, easier to read, and easier to maintain.

Types of Functions in Python:

There are several types of functions in Python, including:

1. Built-in functions like `Print()` and `len()`.
2. User-defined functions, which you define yourself.
3. Anonymous functions, also known as lambda functions.

Advantages of User-Defined Functions:

User-defined functions allow you to encapsulate code for reuse, improve readability, and simplify complex tasks by breaking them into smaller parts.

Rules in Declaring a Function in Python:

Some key rules for declaring functions in Python include:

1. Function blocks begin with the `def` keyword followed by the function name and parentheses ().
2. Any input parameters or arguments should be placed within these parentheses.
3. The first statement of a function can be an optional statement - the documentation string of the function or docstring.
4. The code block within every function starts with a colon (:) and is indented.
5. The statement `return [expression]` exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as `return None`.

Python Function Syntax:

```
def function_name(parameters):  
    """docstring"""  
    statement(s)  
    return [expression]
```

The Return Statement:

The return statement is used to exit a function and optionally return a value to the caller. If no expression is provided, `None` is returned.