

## **Finals - Assignment 1**

### **Defining a Function**

In Python, a function is a reusable block of code that performs a specific task. You define a function using the `def` keyword followed by the function name and parentheses. The code you want the function to execute is indented within the function body.

```
def greet(name):  
    """Prints a greeting message."""  
    print("Hello, " + name + "!")  
  
greet("Alice") # Output: Hello, Alice!
```

### **Reasons to Use Functions**

**Code Reusability:** You can call the same function multiple times with different arguments, avoiding repetitive code.

**Modularity:** Functions break down complex programs into smaller, manageable parts, making code easier to understand, maintain, and test.

**Organization:** Functions help in structuring your code logically.

**Readability:** Using meaningful function names improves code readability.

### **Types of Functions**

**Built-in Functions:** Python comes with many built-in functions like `print()`, `len()`, `str()`, etc.

**User-Defined Functions:** You create these functions to meet your specific needs.

### **Advantages of User-Defined Functions**

**Improved Code Structure:** Breaking down code into functions enhances organization and clarity.

**Reduced Redundancy:** Reusing functions avoids writing the same code multiple times.

**Increased Maintainability:** It's easier to modify or debug a specific function rather than rewriting scattered code.

**Modular Testing:** You can test functions independently, making testing more efficient.

### **Rules for Declaring a Function**

The function definition starts with `def`.

The function name should follow coding conventions (lowercase with underscores for separation).

Parentheses after the name can hold parameters (variables that receive data when the function is called).

A colon (:) follows the parentheses.

The indented code block within the colon defines the function's behavior.

## Python Function Syntax

```
def function_name(parameter1, parameter2, ...):  
    """Function docstring (optional)"""  
  
    # Function body (statements indented)  
  
    return value # Optional return statement
```

## Function Arguments and Parameters

Parameters: These are placeholders within the function definition that hold values passed during the function call.

Arguments: These are the actual values provided when calling the function, assigned to the corresponding parameters.

```
def greet(name): # "name" is the parameter  
    print("Hello, " + name + "!")  
  
greet("Bob")    # "Bob" is the argument passed to "name"
```

## The return Statement

The return statement (optional) specifies a value to send back from the function to the caller. If not used, the function returns None by default.

```
def add(x, y):  
    return x + y  
  
result = add(5, 3) # result will be 8 (returned value from the function)
```