Computing Fundamentals using Python

SUBJECT CODE : UQ25CA151A

Samyukta D Kumta

Computer Applications

Introduction to functions

- **Functions** are modular blocks of code designed to perform specific tasks.
- They enhance code efficiency and clarity by reducing code repetition and enabling code reuse.

Introduction to functions

Types of Functions in Python

Python primarily categorizes functions into two types:

1. **Built-in Functions:** These are predefined functions in Python that are readily available for use. Examples include len(), range(), and abs().
2. **User-defined Functions:** As the name suggests, these are the functions defined by the users to perform specific tasks.

Introduction to functions

Syntax user defined function:

```
def function_name(parameters):
    # code block
    return value  # optional
```

A function is declared using the **keyword def**, succeeded by the name of the function and enclosed parentheses, which may enclose parameters.

Introduction to functions

- **Function Name:** This is the identifier for the function. It follows the same naming conventions as variables in Python. The function name should be descriptive enough to indicate what the function does.
- **Parameters (Optional):** These are variables that accept values passed into the function. Parameters are optional; a function may have none. Inside the function, parameters behave like local variables.
- **Function Body:** This block of code performs a specific task. It starts with a colon (:) and is indented. The function body typically contains at least one statement.
- **Return** : The return statement can be used to send back a result from the function to the caller. None is returned if no return statement is used.

Introduction to functions

Example:

Creating a Function in Python

```
def hello():
    print("Hello World")
```

Calling a Function in Python

```
def hello():
    print("Hello World")
hello()
```

Introduction to functions

Syntax: Python Function with Parameters

```
def function_name (parameters):
   statement 1
   statement 2
   …
   …
   return [expr]
```

Introduction to functions

Example: Python Function with Parameters

```
def add_numbers(a, b):
    result = a + b
    return result

print(add_numbers(2,3))
```

Introduction to functions

Example:

```
def make_greeting(title, name, greeting="Hello"):
    return f"{greeting}, {title} {name}!"

print(make_greeting("Dr", “SAM"))
```

THANK YOU

Samyukta D Kumta
Department of Computer Applications
samyuktad@pes.edu