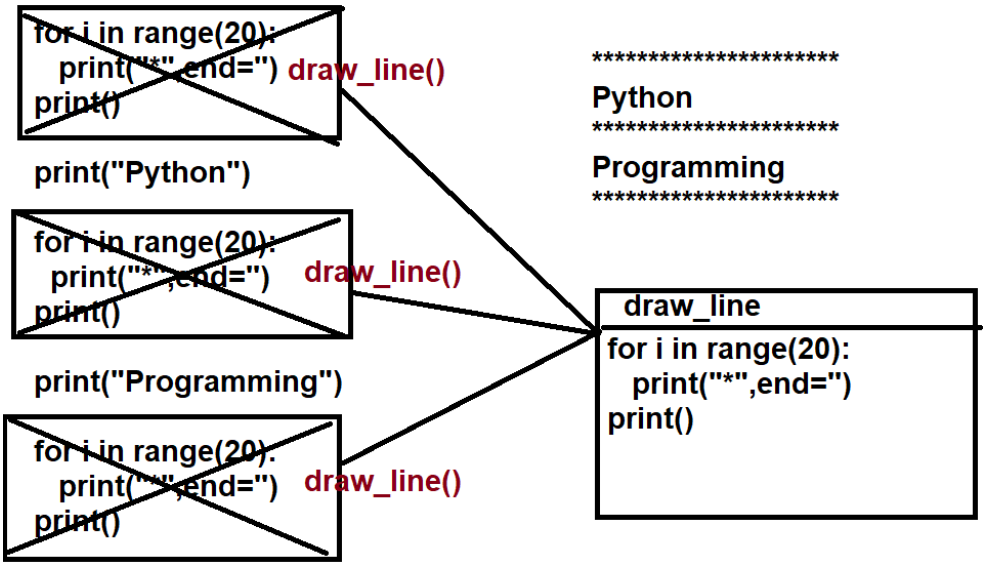
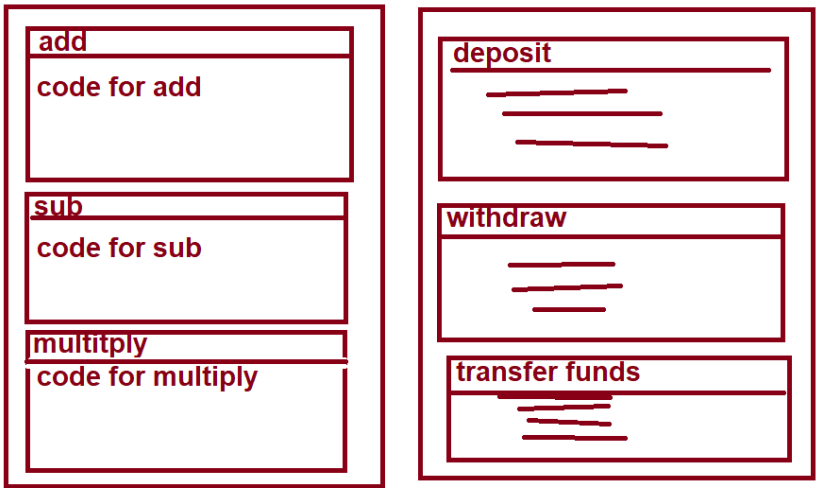
Functions

Python is a multi paradigm programming language. A programming paradigm defines set of rules and regulations for writing programs. In python programmer write programs using different programming paradigms.

1. Procedural Oriented Programming (POP)
2. Modular Oriented Programming (MOP)
3. Object Oriented Programming (OOP)

In procedural oriented programming, program is written by dividing programming instructions into small piece of code, which is called as sub routine.





What is function?

A function is small program within program.
A function is building block of procedural oriented programming.
A function is self contained block, which contain set of instructions to perform some operations.
A function is named block, contain set of instructions.

Advantage of functions

1. **Reusability:** Functions allows writing code once and using more than one time
2. **Modularity:** Dividing programming instructions according their operations into small pieces.
3. **Readability:** Easy to understand
4. **Efficiency:** functions increase performance of application.

Functions are two types

1. Predefined functions
2. User defined functions

Predefined functions

Existing functions are called predefined functions. These functions are also called library functions.
Example: print(),input(),oct(),hex(),bin(),int(),float(),….

User defined functions

A function written programmer is called user defined function. All application specific functions are called user defined functions.
Example: deposit(),login(),logout(),withdraw(),….

A function is divided into two parts.

1. Writing function or defining function
2. Invoking function or calling function or executing function

Writing function or defining function

In python function is defined or created using **“def”** keyword.

Syntax:

```
def <function-name>([parameters/arguments]):
        ‘’’doc string’’’
        Statement-1
```

Statement-2

A function is defined,

1. With parameters
2. Without parameters
3. With return value
4. Without return value

A function is defined with parameters, if function required input.
A function is defined without parameters, if function not required input.

Example of function without parameters:

```
def fun1():
    print("Inside fun1")
def fun2():
    print("Inside fun2")
def fun3():
    print("Inside fun3")

#main
fun1() # calling function or invoking function
fun2()
fun3()
```

Output

```
Inside fun1
Inside fun2
Inside fun3
```

Memory for function is allocated, when function is invoked or called.
Memory for function is de-allocated, after execution of function.
Whenever function is called execution control switched from calling place to called function and after execution of function returning to calling place.

Example:

```
def draw_line():
    for i in range(30):
        print("*",end='')
    print()
```

```
draw_line()
print("PYTHON")
draw_line()
print("PROGRAMMING")
draw_line()
```

Output:

```
******************************
PYTHON
******************************
PROGRAMMING
******************************
```

Local Variable

A variable declared/created inside function is called local variable. This variable is used within function but cannot access outside the function.

Example:

```
def fun1():
    x=100 # Local Variable
    print(f'Local variable x={x}')


fun1()
# print(x) Error
```

Output

Local variable x=100

Global Variable

A variable created outside the function is called global variable. Global variables share data between more than one function.

Example:

```
x=100 # Global Variable
def fun1():
    print(f'Global x={x}')
def fun2():
```

```
    print(f'Global x={x}')

fun1()
fun2()
```

Output:
Global x=100
Global x=100

Example:

```
def fun1():
    print(x)

def fun2():
    print(x)

def fun3():
    print(x)

x=100 # G.V
fun1()
fun2()
fun3()
```

Output:
100
100
100

Example:

```
def fun1():
    print(x)

def fun2():
    print(x)

def fun3():
    print(x)
```

```
fun1()
x=100 # G.V
fun2()
fun3()
```

Output:

```
File "C:\Users\nit\PycharmProjects\pythonProject1\funtest5.py", line 2, in
fun1
    print(x)
          ^
NameError: name 'x' is not defined
```