Parul® University

Python Programming

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-4

Python Functions, Modules and Packages

What is Function?

- Group of statements to perform particular task

- Takes arguments as input
- Returns the result after manipulation

- Two kinds of function
  - Built – in functions : print(), input()
  - User defined functions

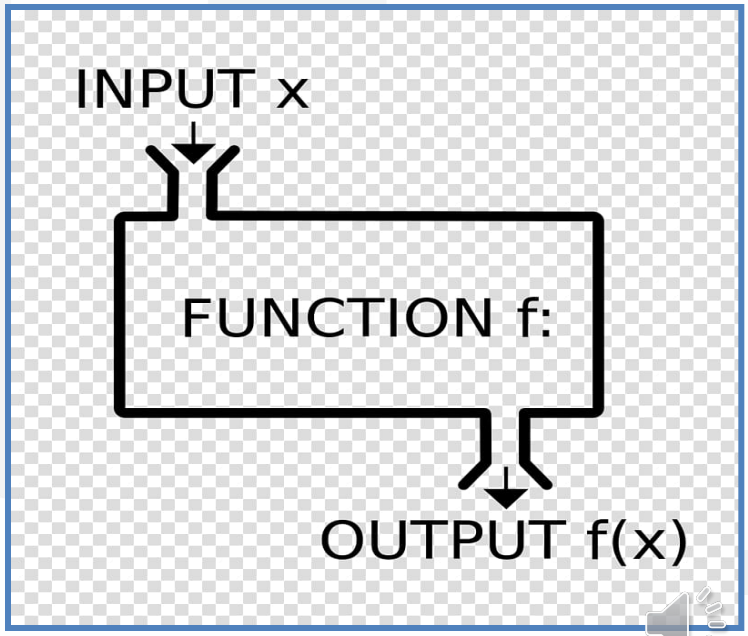


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

Function in Python

- Python uses ‘def’ keyword to create user defined function
- Define function once and call it many times to reuse
- Syntax

Function Definition

```
def <function_name> (<arguments>) :
    # function statement
```

```
def greetings():
    print('Hello...Good Morning!!')
```

Function Calling

```
<function_name>()
```

```
greetings()
```

Arguments

- Input values given to function
- Arguments are specified by parameters in function definition

```
def greetings(lang):

    if lang == 'fr':
        print('Bonjour')
    elif lang == 'es':
        print('Hola')
    else :
        print('Hello...Good Morning!!')

greetings('fr')
greetings('es')
```

Parameters

Arguments

```
Bonjour
Hola
```

Return Values

- Whatever function gives as outcome is defined as return values
- Uses ‘return’ keyword

```
def greetings(lang):

    if lang == 'fr':
        return 'Bonjour'
    elif lang == 'es':
        return 'Hola'
    else :
        return 'Hello...Good Morning!!'

x=greetings('fr')
print(x)
print(greetings('es'))
```

Return Values

```
Bonjour
Hola
```

Multiple Parameters

- Positional Parameter : Arguments values are identified based on their position

```
def printinfo(name,age):

    "This prints a passed info into this function"
    print("Name: ",name)
    print("Age ",age)
    return

printinfo('Nita', 23)
```

Argument at pos = 0

Argument at pos = 1

```
Name:  Nita
Age  23
```

Multiple Parameters

- Keyword Parameter : Arguments values are identified based on their names

```
def printinfo(name,age):

    "This prints a passed info into this function"
    print("Name: ",name)
    print("Age ",age)

printinfo(age=50,name="miki")
```

```
Name:  miki
Age  50
```

Multiple Parameters

- Default Parameter : Arguments values can be default

```
def printinfo(age , name = 'Mosam',):

    "This prints a passed info into this function"
    print("Name: ",name)
    print("Age ",age)

printinfo(50)
```

- Try calling 'printinfo(30 , 'Nita')'

```
Name:   Mosam
Age   50
```

No need to worry about number of arguments

```
def printinfo(*args):
    "This prints a passed info into this function"
    print(args)

printinfo('nita' , 50)
printinfo('nita' , 50 , 'mosam' , 30)
```

Accepts multiple arguments

You get tuple of arguments

```
('nita', 50)
('nita', 50, 'mosam', 30)
```

Lambda Function

- Anonymous functions at runtime
- Can take any number of arguments
- Can only have one expression
- Uses ‘lambda’ keyword to define

Arguments

```
>>> f = lambda f : f**2
>>> f(3)
9
```

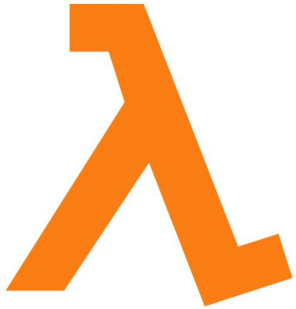


Image source : Google

Recursion : When function call itself

```
def fact(num):
    if num == 0 :
        return 1
    elif num ==1:
        return 1
    else:
        return (num * fact(num-1)) #fact call fact() itself

num = int(input("Enter number:"))
factorial = fact(num)
print (factorial)
```

```
Enter number:
6
```

Recursion : When function call itself

Factorial = fact (3)

return (3 * fact(2)) = 6

return (2 * fact(1)) = 2

1

Modules

- Logically organize your python code in module
- Module is just a python file
- Module can define functions, classes, variables

daily.py

```
def daily(str):
    if(str=="sunday"):
        print('rainy')
```

weekly.py

```
def weekly(str):
    if(str=="1"):
        print('rainy week')
```

Modules

- Module contents are made available to the caller with the import statement. The import statement takes many different forms, shown below.

- import <module_name>

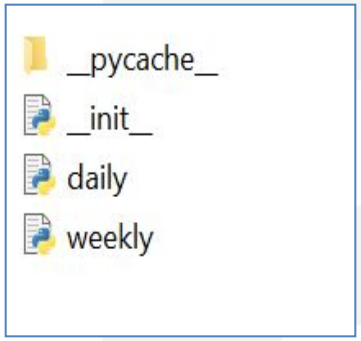
  OR

- from <module_name> import <name(s)>

Packages

- Hierarchical organization of modules
- Package is a directory consists of modules and file named as : __init__.py

weatherman



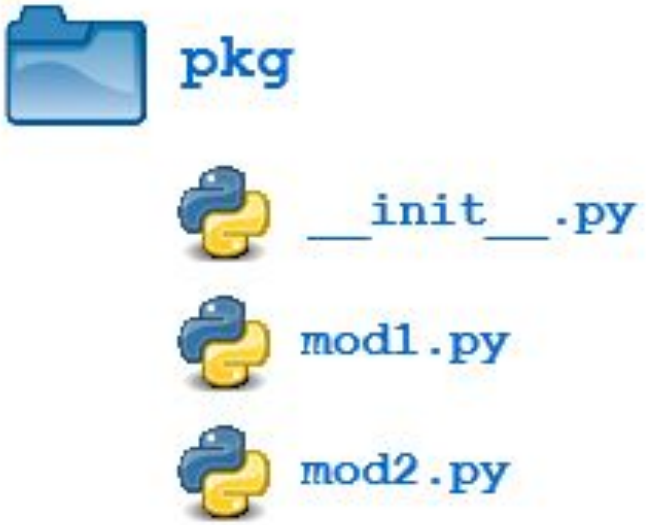
weatherman is a package consists of daily and weekly modules

- Use 'weatherman' package in another python file using 'from' and 'import' statement

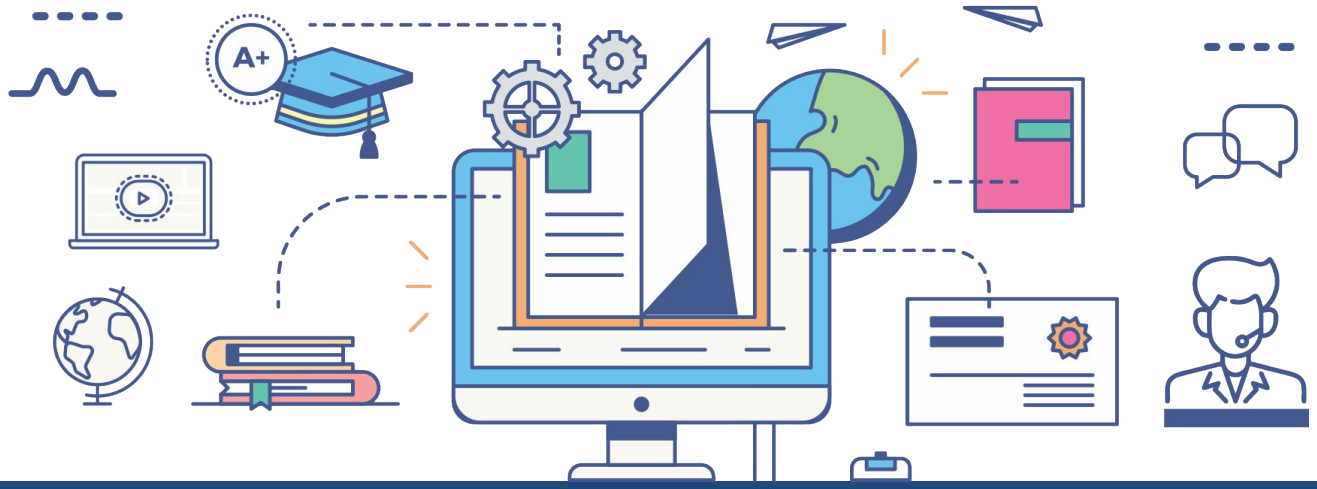
```
from weatherman import daily, weekly

print('Forecasting:',daily.daily())
print('Weekly Forecasting:',weekly.weekly())
```

Packages

- Hierarchical organization of modules
- Package is a directory consists of modules and file named as : __init__.py



DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in