1) Lambda Functions in Python :-

→ Lambda Functions in Python are anonymous Functions, implying they dont have a name.

→ As we know the 'def' keyword must be used to create a typical function in Python.

→ We can also use the Lambda keyword to define an unnamed function in Python.

Syntax :-

lambda arguments : expression

→ This Function accepts any counts of inputs but only evaluates and returns only one expression.

→ Simply means it takes many inputs but returns only one output.

→ Lambda Functions can be used whenever function arguments are necessary.

→ In addition to other forms of formulations in functions, it has a variety of applications in certain coding domains available.

→ It is very important to remember that according to Syntax, lambda functions are limited to a single statement.

Example :-

(I) Source code:

```
# Code to demonstrate how to use lambda Fn.
add = lambda num : num+4
Print (add(6))
```

- output:

  10.

II. Source code:

```
a = lambda x,y : (x*y)
Print (a(5,6))
```

- output:

  30.

III Source code:

```
def add (num):
    return num + 5
Print (add (6))
```

- output:

  11

2) Modules in Python :-

→ Module is a file that contains code to perform specific task. A module may contain variables, functions, classes and etc.

→ As our Program grows bigger, it may contain many lines of codes. Instead of putting everything into a single file, we can use modules to seperate codes in seperate files.

→ This makes our code organised and easier to maintain which is done by the Python Modules.

- Steps to Create a Module :-

→ Let us create a Module to add two numbers and save it as "example.py".

⇒ # Python Module

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

→ Here we have defined a function "add()" inside a Module named "example"

→ The Function takes in two variables (a & b) and returns their Sum.

- Import Python Modules :-

→ We can import the definitions inside a module to another module (or) the interactive interpreter.

→ We use the "import" keyword to do this.

for eg:- No we are going to import the already defined Module "example.py" using following prompt.

⇒ import example.

- Accessing Python Module :-

→ We use dot (.) operator to access the python Modul.

⇒ ie : addition.add (4,5)

output : 9.

- Eg code to Return "Pi" value!

⇒ Source code :

```
import math
Print(" Pi value is : ", math.pi)
```

⇒ output :

Pi value is : 3.141592653589793.

3) Python : Math Module.

→ Using Math module in Python we can easily calculate many mathematical calculations in Python using the math module.

→ Mathematical calculation may occasionally be required when dealing with specific fiscal (or) rigorous tasks.

→ The function in math module can perform simple mathematical calculations like addition(+) and subtraction(-) and advanced trigonometric and logarithmic operations

→ Modules in Python are generally imported to the code using the keyword "import".

→ Python has a built-in math module which is a standard module so we do not need to install them seperately.

→ Since the source code of "math" module is in 'C' language, it provides access to the functionalities of underlying 'c' library.

→ Below are some basic examples to understand about the math modules function in Python.

Python Module Examples :-

I Importing math module to find Square root :

```
import math
Print (math.Sqrt (9))
```

- Output :

  3.0

II Importing math module to find the factorial of a number :

```
import math
n = int (input())
Print (" Enter a Number :" , n )
Print (math.factorial (n))
```

- output :

  Enter a Number : 5

  120.

III Importing math module to find pi value

```
import math
Print (math.pi ())
```

- Output :

  3.141592653589793.

4.) Python : Time Module :-

→ The Time module in Python provides functions for handling or performing time related tasks.

→ The Time related tasks include.,

* Reading Current Time.
* Formatting Time.
* Sleeping for specified no. of Sec's and so on.

- Python time.time() Function:

→ The time() function returns the number of seconds passed since epoch.

eg:
```
import time
Seconds = time.time()
Print (" Sec's Since Epoch = ", Seconds)
```

* Output: Sec's Since Epoch = 1672215379.5045543

- Python time.ctime() Function :

→ The time.ctime() function takes seconds passed since epoch as an argument and represents local time.

Eg:
```
import time
Seconds = 1672215379.5045543
```

```
local_time = time.ctime(seconds)
Print("Local Time:", local_time)
```

* Output:

```
Local Time: Wed Dec 28 08:16:19 2022
```

- Python time.sleep() Function:

→ sleep() function delays the execution of the current thread for the given number of seconds.

eg:

```
import time
Print("Printed Immediately!")
time.sleep(2.4)
Print("Printed After 2.4 Seconds!")
```

* output:

```
Printed Immediately!
Printed After 2.4 Seconds!
```

→ These are some time() functions in the Time Python Module

→ So these module functions are usually used to handle and perform the time related tasks

→ There are other time() functions in the Time Module like time.localtime(), time.gmtime(), time.mktime() and etc available in Python.

5.) Python: ~~Help()~~ Function :-

→ The help() Method (or) Function in Python is used for interactive useses.

→ It is recommended to tay it in your interpreter when you need help to Python program.

→ The help() function in Python is also used to write Python Modules.

Syntax :-

The Syntax of help() is ;

help(object)

help() Parameters :

→ The help() function takes a maximum of only one parameter.

→ object is optional - you want to generate the help of the given 'object'.

* Note :-

The object is only passed in the help() method and it is not a String.

Try These in Python Shell:

```
>>> help(list)
>>> help(dict)
>>> help(print)
>>> help([1,2,3])
>>> help('random thing')
>>> help('def')
>>> help()
>>> from math import *
    help('math.pow')
```

We can also write (or) enter the name of the topic in shell to get help on writing programs and using python modules.

eg:

```
help > True
help > 'print'
help > print
```

→ To quit the help utility and return to the interpreter, you need to type 'quit' and Press enter

ie: help > quit.

→ These are some functionalities and uses of the help() function that is available in Python.