

Data structures in Python

Week 8A: user defined modules



○ User defined modules

The subprograms namely procedures and functions that are created by user can be used in other programs (**import**) as well. These are called user defined modules.

math – it is a module/library that contain predefined procedures and functions such as `sqrt()`, `pow()`, `pi()` and more.....

```
1 math.py/library file
2 def sqrt(n):
3     ...
4     ...
5     return n
6
7 def pi(n):
8     n = 3.141592653589793
9     return n
10
11 def pow(x, y)
12     p = 1
13     for i in range(y):
14         p = p*x
15     return p
16
17 .....
```

Importing module/library

Using `sqrt()` defined in math

Using `pow()` from imported math library

```
mathcheck.py x
1 #using math library subprograms
2 import math #importing math module here
3
4 x = 3
5 y = 2
6 print(math.sqrt(x))
7 print(math.pow(x,y))

Shell x
Python 3.7.9 (bundled)
>>> %Run mathcheck.py
1.7320508075688772
9.0
```



Attention: It is not a real Python's math library and contents- Just faked here to understand how to define a module with subprograms and reuse it. So, use different name for module instead of math and try this.



- Let us define our own modules and use it in our other programs
- Create a module called **bigSmall** that contain procedure and functions to get smaller or greater than two numbers via sub programs. Then we use those created subprograms in our other programs.

bigSmall.py ×

```
1 #find smaller one
2 def smallProcedure(x,y):
3     if x<y:
4         print(x)
5     else:
6         print(y)
7
8 def smallFunction(x,y)
9     if x<y:
10        return x
11    else:
12        return y
13
14 #find smaller one
15 def bigProcedure(x,y):
16     if x>y:
17         print(x)
18     else:
19         print(y)
20
21 def bigFunction(x,y)|
22     if x>y:
23         return x
24     else:
25         return y
```


findsmallest.py ×

```
1 import bigSmall #importing user defined module here
2
3 a = int(input("enter the a value:"))
4 b = int(input("enter the b value:"))
5 bigSmall.smallProcedure(a,b)
6
7 p = float(input("enter the p value:"))
8 q = float(input("enter the q value:"))
9 print(bigSmall.bigFunction(p,q))
10
```

Shell ×

```
Python 3.7.9 (bundled)
>>> %Run findsmallest.py

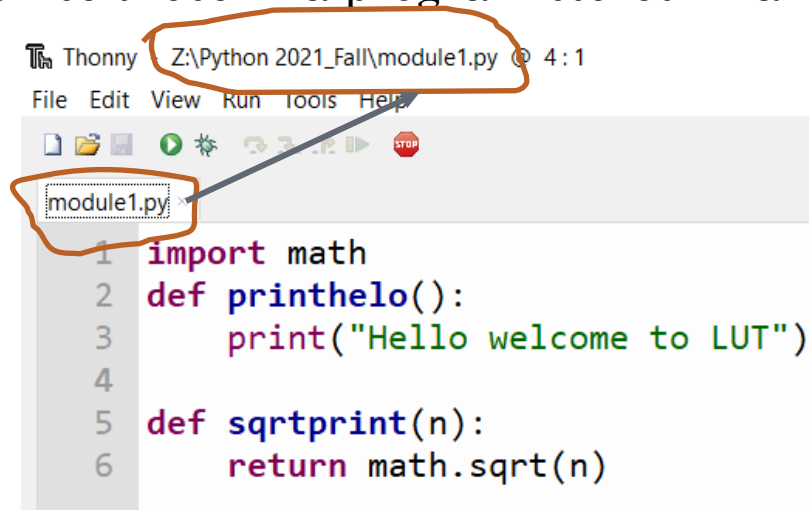
enter the a value:34
enter the b value:45
34
enter the p value:-23.56
enter the q value:-14.5
-14.5
```



Attention:

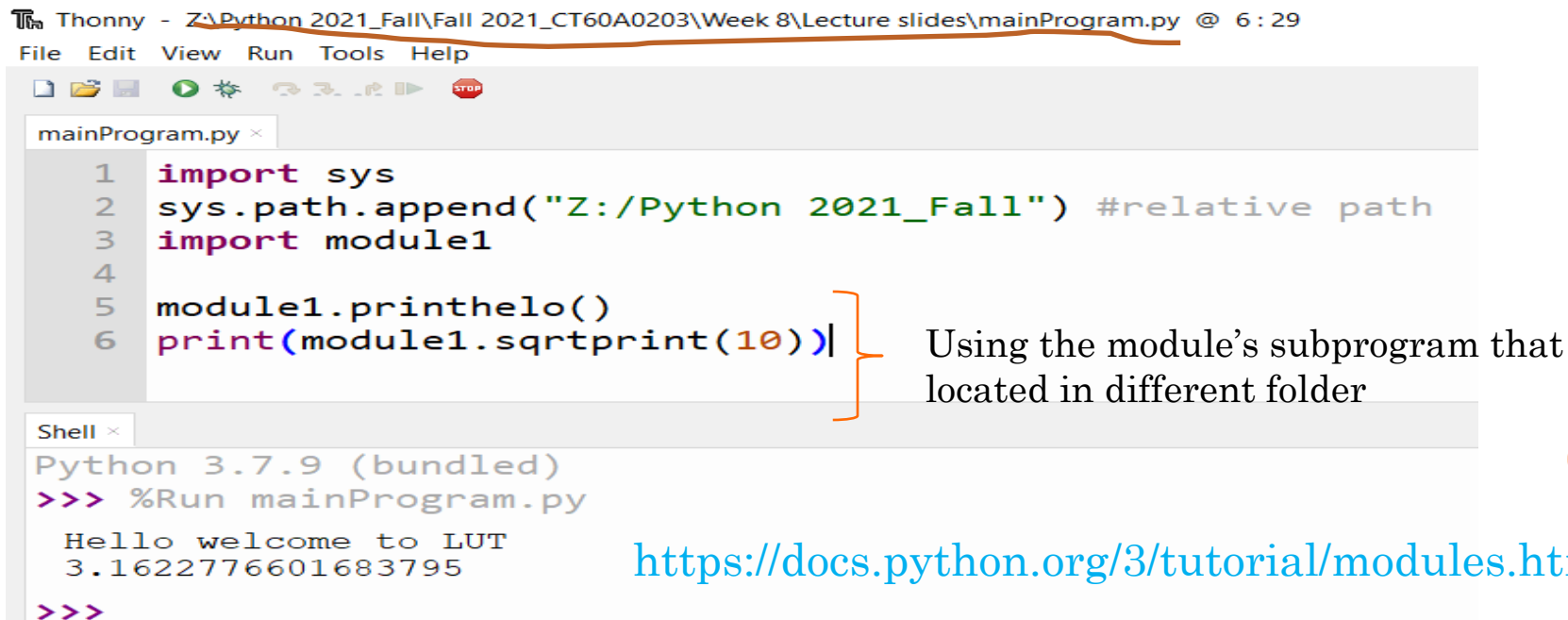
The results of called procedure and function totally depending on the way you defined in your module. Keep in mind you are a programmer not the computer. Computer is just a machine which execute the code you supplied.

- How about importing modules from different folder?
- Suppose you have kept a module that with some user-defined subprograms in a different folder and wants to use those in a program stored in another folder.



A screenshot of the Thonny IDE interface. The title bar shows 'Thonny - Z:\Python 2021_Fall\module1.py @ 4:1'. The menu bar includes 'File', 'Edit', 'View', 'Run', 'Tools', and 'Help'. The toolbar contains icons for file operations and execution. The file explorer on the left shows 'module1.py'. The main editor displays the following Python code:

```
1 import math
2 def printhelo():
3     print("Hello welcome to LUT")
4
5 def sqrtprint(n):
6     return math.sqrt(n)
```



A screenshot of the Thonny IDE interface showing a second file, 'mainProgram.py'. The title bar shows 'Thonny - Z:\Python 2021_Fall\Fall 2021_CT60A0203\Week 8\Lecture slides\mainProgram.py @ 6:29'. The menu bar and toolbar are the same. The file explorer on the left shows 'mainProgram.py'. The main editor displays the following Python code:

```
1 import sys
2 sys.path.append("Z:/Python 2021_Fall") #relative path
3 import module1
4
5 module1.printhelo()
6 print(module1.sqrtprint(10))
```

A bracket on the right side of the code, spanning lines 5 and 6, points to the text: 'Using the module's subprogram that located in different folder'.

Below the editor is a 'Shell' window showing the execution of the program:

```
Python 3.7.9 (bundled)
>>> %Run mainProgram.py

Hello welcome to LUT
3.1622776601683795
>>>
```

<https://docs.python.org/3/tutorial/modules.html>