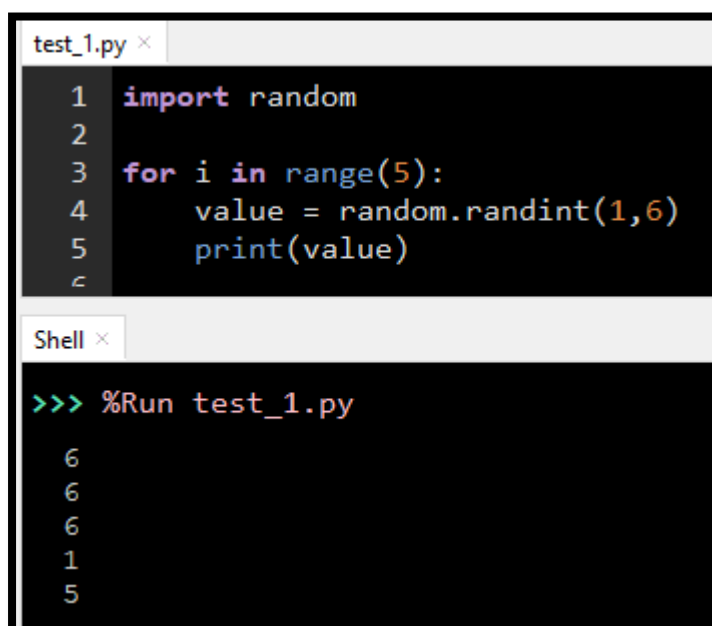


## Modules

**Modules** ဆိုတာက အတွေ့ရများတဲ့ (ခဏခဏအသုံးပြုရတဲ့) လုပ်ဆောင်ချက် (ဥပမာ - ကျပ်နန်းနံပါတ်တွေထုတ်ခြင်း၊ သင်္ချာဆိုင်ရာတွက်ချက်မှုတွေလုပ်ဆောင်ခြင်း) တွေကို ပြုလုပ်ဖို့အတွက် အခြားလူတွေရေးသားထားတဲ့ ကုဒ်အစုအဝေး၊ ကုဒ်ဖိုင် (code file) ကို ခေါ်ဝေါ်တာ ဖြစ်ပါတယ်။

**module** တစ်ခုကို အသုံးပြုတဲ့နည်းလမ်းက ကိုယ်ရေးတဲ့ပရိုဂရမ်ရဲ့ထိပ်ဆုံးမှာ **import** နဲ့ **module** အမည် တွဲရေးရမှာဖြစ်ပြီး module ထဲမှာပါတဲ့ function နဲ့ တန်ဖိုးတွေကို ရယူအသုံးပြုဖို့အတွက် module အမည်နောက်ကနေ module ထဲမှာပါတဲ့သက်ဆိုင်ရာ function အမည်ကို ထည့်ပြီးအသုံးပြုရမှာ ဖြစ်ပါတယ်။ အောက်ကပုံမှာ **random** module ထဲကနေ **randint()** function ကို ခေါ်ယူအသုံးပြုပြထားပါတယ်။ **randint()** ရဲ့လုပ်ဆောင်က ကျပ်နန်းနံပါတ်တွေ ထုတ်ပေးတာ ဖြစ်ပါတယ်။



```
test_1.py x
1 import random
2
3 for i in range(5):
4     value = random.randint(1,6)
5     print(value)
6

Shell x
>>> %Run test_1.py
6
6
6
1
5
```

နောက်ထပ် **import** ပြုလုပ်တဲ့ နည်းလမ်းတစ်ခုကတော့ **module** ထဲကနေ မိမိအသုံးပြုလိုတဲ့ function() တစ်ခုရဲ့အမည်ကို ခေါ်ပြီး **import** ပြုလုပ်တာဖြစ်ပါတယ်။ *module* ထဲက *function* တွေအားလုံးကို အသုံးပြုဖို့မလိုဘဲ *function* တစ်ခုထဲကို အသုံးပြုချင်တဲ့အခါမှာ ဒီနည်းလမ်းကို အသုံးပြုနိုင်ပါတယ်။ အောက်မှာပြထားတဲ့ပုံက **math** module ထဲကနေ **pi** ကိန်းသေတန်ဖိုးတစ်ခုထဲကို **import** ပြုလုပ်ထားတာဖြစ်ပါတယ်။

```
test_1.py ×
1  from math import pi
2
3  print(pi)
4

Shell ×
>>> %Run test_1.py
3.141592653589793
>>> |
```

အကယ်၍ တန်ဖိုး/function တွေတစ်ခုထက်မက ထည့်သွင်းချင်ရင် ကော်မာ (,) ခံပြီး import လုပ်နိုင်ပါတယ်။

```
test_1.py* ×
1  from math import pi, sqrt
2
```

module ထဲက ရှိသမျှအရာတွေအားလုံးကို import လုပ်ချင်ရင် \* ကို သုံးပါတယ်။ ဒါ ပေမဲ့ အခြေခံအားဖြင့် ဒီနည်းလမ်းကို အားမပေးပါဘူး။ ဘာကြောင့်လဲဆိုတော့ မိမိပရိုဂရမ်ထဲမှာ ပါတဲ့ variable အမည်နဲ့ ပြင်ပ module ထဲက variable အမည် (function/value အမည်တွေကို ဆိုလိုခြင်းဖြစ်) တွေ ရောထွေးနိုင်တဲ့အတွက် ဖြစ်ပါတယ်။

module တစ်ခုကို import လုပ်လို့မရရင် (တကယ်မရှိတဲ့ module တစ်ခုကို import လုပ်လို့မရရင်) ModuleNotFoundError ပြပါတယ်။

```
test_1.py ×
1 import some_module
2

Shell ×
>>> %Run test_1.py
Traceback (most recent call last):
  File "D:\Job\Python Myanmar Syllabus\test 1.py", line 1, in <module>
    import some_module
ModuleNotFoundError: No module named 'some_module'
>>>
```

module သို့မဟုတ် module ထဲက function/value တစ်ခုခုကို import လုပ်တဲ့အခါမှာ နာမည်က ရှည်လွန်းရင် သို့မဟုတ် ဖတ်ရနားလည်ရခက်ရင် as keyword ကိုသုံးပြီး မိမိနှစ်သက်ရာ နာမည်အမည်ကို ပြောင်းလဲအသုံးပြုနိုင်ပါတယ်။

```
test_1.py ×
1 from math import sqrt as square_root
2 print(square_root(100))
3

Shell ×
>>> %Run test_1.py
10.0
>>> |
```

## The Standard Library & pip

Python မှာ အဓိက **module** သုံးမျိုးရှိပါတယ်။

- မိမိဘာသာရေးသည့် **module** ၊
- အခြားသူများ ရေးထားသည့် **module** ၊
- Python တွင် မူလကတည်းက built in ပါဝင်သည့် **module** တို့ဖြစ်ပါတယ်။

**မူလကတည်းကပါတဲ့ module** တွေကို **standard library** လို့ခေါ်ပါတယ်။ အသုံးတည့်တဲ့ modules တွေ အများကြီးပါဝင်ပြီး အသုံးများတဲ့ module တွေကတော့ **string, re, datetime, math, random, os, multiprocessing, subprocess, socket, email, json, doctest, unittest, pdb, argparse** နဲ့ **sys** တို့ ဖြစ်ပါတယ်။ ဒီ module တွေရဲ့ လုပ်ဆောင်ချက်တွေအကြောင်းကို Python ရဲ့ documentation မှာ ဖတ်နိုင်ပါတယ်။ လုပ်ဆောင်ချက်တွေအများကြီးပါဝင်တဲ့ Python ရဲ့ standard library ကြောင့် Python က အစွမ်းထက်တဲ့ programming language တစ်ခု ဖြစ်နေတာ ဖြစ်ပါတယ်။

standard library ထဲမှာရှိတဲ့ module အချို့ကို Python နဲ့ရေးထားပြီး အချို့ကို C နဲ့ ရေးထားပါတယ်။ အများစုက Platform အမျိုးမျိုးမှာသုံးနိုင်ပေမဲ့ အချို့ကိုတော့ Windows သို့မဟုတ် Unix platform မှာပဲ အသုံးပြုနိုင်ပါတယ်။ module တွေအကြောင်း အသေးစိတ်ကိုတော့ [www.python.org](http://www.python.org) မှာ လေ့လာဖတ်ရှုနိုင်ပါတယ်။

အခြားသူတွေဖန်တီးထားတဲ့ third-party Python modules တွေကို Python Package Index (PyPI) မှာ သိမ်းဆည်းထားပါတယ်။ ဒီ modules တွေကို Install ပြုလုပ်ဖို့အကောင်းဆုံးနည်းလမ်းက **pip** လို့ခေါ်တဲ့ ပရိုဂရမ်တစ်ခုကို အသုံးပြုတာဖြစ်ပါတယ်။ နောက်ပိုင်း Python ဗားရှင်းတွေမှာ **pip** ကို ထည့်သွင်းပေးထားပါတယ်။ မိမိမှာမရှိခဲ့ရင်လည်း Online ကနေ အလွယ်တကူ Install လုပ်နိုင်ပါတယ်။ **pip** ရှိတယ်ဆိုရင် **PyPI** ကနေ **Library** တွေကို Install ပြုလုပ်ရတာ အရမ်းလွယ်ကူသွားပါပြီ။ မိမိထည့်သွင်းချင်တဲ့ library အမည်ကိုရှာ command line (Windows တွင် Command Prompt လို့ခေါ်ပါတယ်) ကို သွားပြီး **pip install library\_အမည်** လို့ ရေးသားရုံပဲ ဖြစ်ပါတယ်။ မိမိလိုချင်တဲ့ library ကို သွင်းပြီးပြီဆိုတာနဲ့ ကိုယ့်ရဲ့ program ထဲမှာ import လုပ်ပြီး ထည့်သုံးနိုင်ပြီ ဖြစ်ပါတယ်။

**pip** ကိုအသုံးပြုတာက Operating System အများစုအတွက် Library ထည့်သွင်းနည်းစံသတ်မှတ်ချက်ပဲ ဖြစ်ပါတယ်။ သို့ပေမဲ့ အချို့ library တွေကတော့ ကွန်ပြူတာမှာ အခြားပုံမှန်ဆော့ဖ်ဝဲတွေထည့်သွင်းသလို GUI နဲ့ထည့်သွင်းရတာတွေလည်းရှိပါတယ်။

Thonny Software မှာ library နဲ့ module တွေကို ထည့်သွင်းလိုရတဲ့ pip ကို တပါတည်း ထည့်ပေးထားပါတယ်။ Tools ထဲကနေ Manage Packages ကို သွားပြီး မိမိထည့်သွင်းလိုတဲ့ module ကို **install** လုပ်နိုင်ပါတယ်။ ပိုပြီးလွယ်ကူပါတယ်။

**pip** ကို **command line** ကနေ ထည့်သွင်းနည်းကို အောက်မှာလေ့လာနိုင်ပါတယ်။

<https://phoenixnap.com/kb/install-pip-windows>

(**Library** ဆိုတာက **Modules** တွေကို စုစည်းထားတာ ဖြစ်ပါတယ်။)