# Session 9 - Modules

1. Types of modules
   * Built-in Modules
   * Installed Modules
   * user created modules
2. Getting started with the PyAutoGui Module
3. Creating simple Modules



## Modules are one of the biggest reasons why Python is so famous.

You can literally think of any task and you can find an existing module that will make your life a lot easier.

# What is MODULE

***A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.***

# syntax for module

module\_name.function\_name.

# 1. Built-in Modules

Few Modules come in handy in everyday programming. So these modules come preinstalled when you install Python. So you can directly install and start using these modules.

The below link shows a list of all the modules that come preinstalled with Python built-in [Modules](https://docs.python.org/3/py-modindex.html)

### Let's try using the Time module

To import a module we can use the following **syntax** (there are multiple ways to import a module this is just one of them)

import module\_name

import time

This makes all the functions defined inside the time module available to us. Once imported we can call any function we want.

print("hello")  
time.sleep(2)  
print("world")

hello  
world

If you run the above cell you will get hello printed on the screen followed by the world after a 2-second delay. well, that is what the sleep function does, it makes the Python code wait for some time.

### This brings us to the next important thing while using modules which is reading the documentation.

### !! Always read the documentation before you start working with a module.

[Time Module](https://docs.python.org/3/library/time.html#time.sleep)

random Module

# 2. Installed Modules

All Modules apart from the Built-in ones need to be installed to be able to use them. Almost all the modules created by the users can be found on the [PyPi](https://pypi.org/) website

### Installing modules is pretty straightforward

**For Windows**

* open the command prompt and type

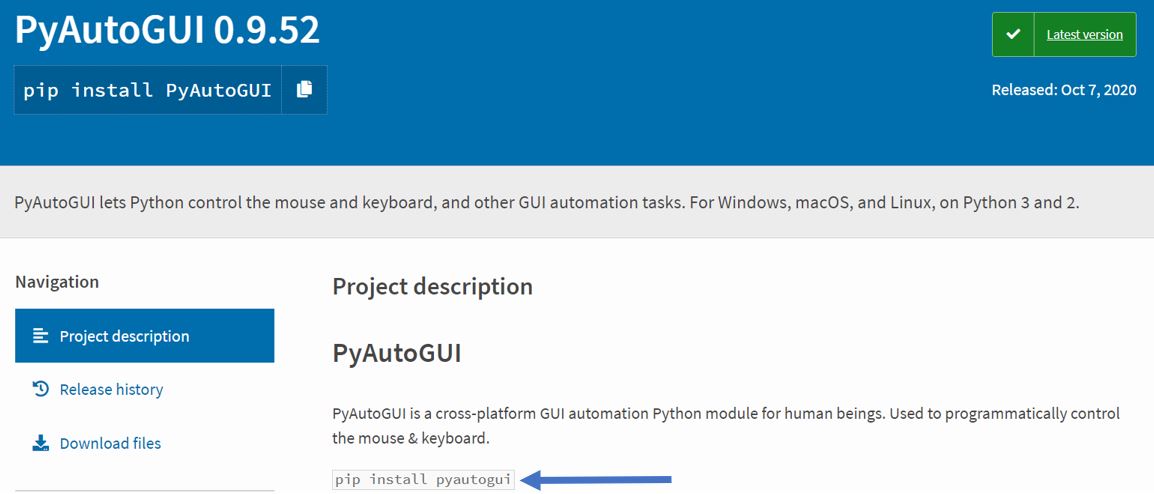
pip install module\_name

**For Mac**

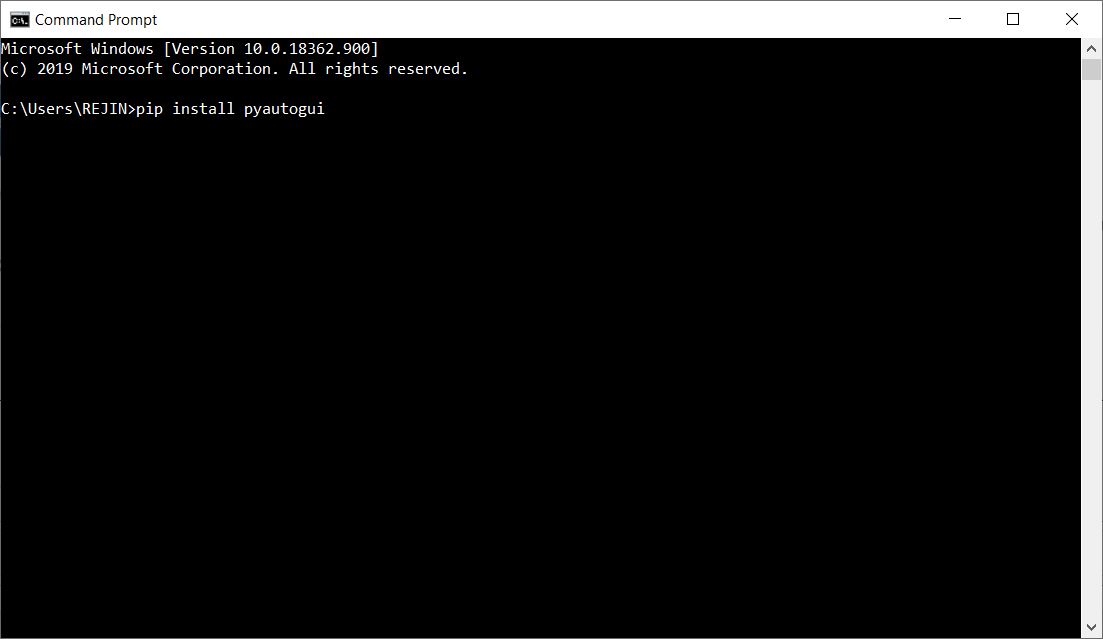
* open Terminal and type

pip3 install module\_name

**NOTE: The instructions to install a module can be found on the PyPi project page itself. See the below image**

****

We can open a command prompt or terminal on Windows to use the command shown to install the module



To verify if the module is installed properly we can try importing the module

import pyautogui

If you don't get any output it means the module is installed properly. If you get a ModuleNotFound Error check the command prompt or terminal for the reason

import pyautogui

**Well, what is the package that we just installed??**

As the PyPi website states it's a cross-platform module that allows us to control the keyboard and mouse programmatically.

**As with any other Modules read the documentation before you start using it.** [PyAutoGui Documentation](https://pyautogui.readthedocs.io/en/latest/)

### Let's try to bring the search bar up on ---> windows.

we need to press the win+q key to make that happen

import pyautogui  
  
pyautogui.keyDown('win')  
pyautogui.press('q')  
pyautogui.keyUp('win')

If you run the above cell it should bring up the search bar

### Let's try to bring up the spotlight on ---> Mac.

import pyautogui  
  
pyautogui.keyDown('command')  
pyautogui.press('space')  
pyautogui.keyUp('command')

### Wait It didn’t work!!

Well in Mac you need to provide some extra access for the pyautogui module to be able to automate things.

You can follow these steps to provide it access.

Go to system preferences ---> unlock the lock ---> give access to the app you are using to run the code ---> save changes ---> lock

# 3. User defined module

Here you can create your module. Let's try to make our module for the calculator. We will use separate files to define functions used in the calculator And then will import the file into the main (execution file)

# 4. Creating a Banking App using Functions user-defined module

**The App should have the following features**

**Simple Tasks**

1. Create an account (During account creation username, password, and the initial deposit amount should be specified.)

2. The account holders should be able to log in and log out from the account using a password.

3. Account holders should be allowed to credit and debit Money.

**Difficult Tasks**

1. Remove Account Holder

2. The app should have checks which limit the withdrawal of amounts bigger than the current balance in the account.

3. Should have a way to check the current balance

1. **Make use of the function for each task**

def function\_name():

2. **Use a dictionary of dictionaries to store the account holder details**

acc\_hold = {"user1":{"pass":1234, "bal":50000}, "user2":{"pass":5678, "bal":20000}}

**Main dictionary to store all the account holders and their details**

# Account holder dictionary to store all the user details

acc\_hold = {}

**Global variables used for program state**

# variables used to check the state of things

login = False

running = True

**Login function**

def login():

global login

user\_name = input("Enter the user name : ")

password = input("Enter the password : ")

if user\_name in acc\_hold:

if acc\_hold[user\_name]["pass"] == password:

print("Logging in")

login = True

login\_menu(user\_name)

else:

print("wrong password")

else:

print("account not found")

**Logout function**

def logout():

global login

login = False

print("logging out")

**Login Menu**

def login\_menu(user):

global login

while login:

print("For deposit press 1: ")

print("For withdrawal press 2: ")

print("For logout press 3: ")

x = int(input("Enter your choice: "))

if x == 1:

dep = int(input("Enter deposit amount: "))

acc\_hold[user]["bal"] = acc\_hold[user]["bal"]+dep

print("current balance is:",acc\_hold[user]["bal"])

elif x == 2:

wit = int(input("Enter withdraw amount: "))

if wit > acc\_hold[user]["bal"]:

print("not sufficient balance.")

else:

acc\_hold[user]["bal"] = acc\_hold[user]["bal"]-wit

print("current balance is:",acc\_hold[user]["bal"])

elif x == 3:

logout()

**Function to create an account**

def create\_acc():

user\_name = input("Enter the user name: ")

password = input("Enter the password: ")

int\_dep = int(input("Enter a deposit amount: "))

if user\_name in acc\_hold:

print("user name already taken")

else:

acc\_hold[user\_name] = {"pass": password, "bal":int\_dep}

print("account created")

**Function for the main menu of the banking app**

def main\_menu():

print("Welcome to the banking app.")

print("For login press 1: ")

print("For creating new account press 2: ")

print("For exiting the app press 3: ")

x = int(input("Enter your choice: "))

if x == 1:

login()

elif x == 2:

create\_acc()

elif x == 3:

exit()

**Function exit the banking app**

def exit():

global running

print("closing the app")

running = False

**While true loop to keep the app running.**

while running:

main\_menu()

# Homework

### 1. Write a Python script to automatically send a mail to yourself using PyAutoGui

1. open google chrome
2. open Gmail
3. click on Compose mail
4. Fill in the email ID, Subject, and Body
5. Send mail using the shortcut

#Task1: Write a Python program to generate a random alphabetical character, alphabetical string, and alphabetical string of a fixed length.

#Task2: Write a Python program to construct a seeded random number generator, also generate a float between 0 and 1, excluding 1.

# Task 3: Write a Python program to select a random element from a list, set, dictionary (value), and a file from a directory.

#Task4: Write a Python program to generate a float between 0 and 1, inclusive, and generate a random float within a specific range.

# Task 5: Write a Python program to create a list of random integers and randomly select multiple items from the said list.

#Task6: Write a Python program to configure the rounding to round up and round down a given decimal value. Use decimal. Decimal