These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

Script, Module Package and Library

A script is a Python file that's intended to be run directly. When you run it, it should do something.

This means that scripts will often contain code written outside the scope of any classes or functions.

A module is a Python file that's intended to be imported into scripts or other modules. It often defines members like classes,

functions, and variables intended to be used in other files that import it.

A package is a collection of related modules that work together to provide certain functionality. Eg math

These modules are contained within a folder and can be imported just like any other modules.

A library is an umbrella term that loosely means "a bundle of code."

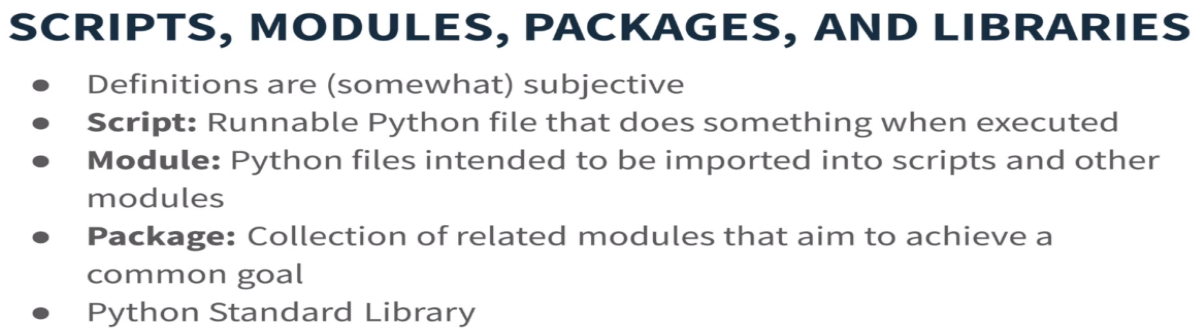
These can have tens or even hundreds of individual modules that can provide a wide range of functionality.

Matplotlib is a plotting library.

The Python Standard Library contains hundreds of modules for performing common tasks**, like sending emails or reading JSON data**.

What's special about the Standard Library is that it comes bundled with your installation of Python,

so you can use its modules without having to download them from anywhere.



#scripts are intended to be run directly, whereas modules are meant to be imported.

```
a=1

b=2

print(a+b)

#o/p

3
```

These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

add.py

```
#module

def add(a,b):

   return(a+b)
```

pgm.py

```
from add import add

output=add(7,8)

print(output)

#o/p

#15
```

A module contains different functions.

- Modules act as a pre-defined library in the script
- The python modules also store pre-defined functions from the library when the code is being executed.
- We have imported the required module and used the pow() function to calculate the powers of the given number as arguments. We have then printed the value of the pow for the user.

```
from math import pow

# using the pow() function

val=pow(2, 3)

# printing pow()

print(val)

#o/p

#8.0
```

A package is considered a collection of tools that allows the programmers to initiate the code.

- A Python package acts as a user-variable interface for any source code.
- This feature allows a Python package to work at a defined time for any functional script in the runtime.

```
# importing the package

import math

# printing a statement
```

These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

```
print("generates a print")

#o/p

#generates a print
```

Difference between Package and Library:

- A Package consists of the __init__.py file for each user-oriented script.
    - A package also modifies the user interpreted code in such a manner that it gets easily operated in the runtime.
- A module is a file that contains a Python script in runtime for the code specified to the users.
    - A python "module" contains a unit namespace, with the variables that are extracted locally along with some parsed functions like:
    - Constants and Variables
    - Any old or new value
    - Class definitions of properties
    - A module generally corresponds to a single file
    - A debugging tool in the user interface library.

Module time()

```
import time;  # This is required to include time module.

ticks = time.time()

print("Number of ticks since 12:00am, January 1, 1970:", ticks)
```

- #o/p
- #Number of ticks since 12:00am, January 1, 1970: 1667212743.631483

```
import time

localtime = time.localtime(time.time())

print("Local current time :", localtime)

#Local current time : time.struct_time(tm_year=2022, tm_mon=10, tm_mday=31,

#tm_hour=16, tm_min=10, tm_sec=53, tm_wday=0, tm_yday=304, tm_isdst=0)

#isdst –daylight saving
```

These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

Index	Attribute
0	**tm_year**
1	**tm_mon**
2	**tm_mday**
3	**tm_hour**
4	**tm_min**
5	**tm_sec**
6	**tm_wday**
7	**tm_yday**
8	**tm_isdst**

Module calendar

```
import calendar
cal = calendar.month(2008, 1)
print("Here is the calendar:")
print(cal)
```

- #Here is the calendar:
- #   January 2008
- #Mo Tu We Th Fr Sa Su
- #   1 2 3 4 5 6
- # 7 8 9 10 11 12 13
- #14 15 16 17 18 19 20
- #21 22 23 24 25 26 27
- #28 29 30 31

These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

Date time

```
from datetime import *
m = str(input(''))
#2022-11-15 11:00
base = datetime.strptime(m, "%Y-%m-%d %H:%M")
print(base)    #2022-11-15 11:00:00
n=int(input())
 #4
#Python timedelta() function is present under datetime library which is generally used
#for calculating differences in dates and also can be used for date manipulations in Python
for i in range(0, n):  #modify 1 to n
   #print(timedelta(days=i))
   #0:00:00
   #1 day, 0:00:00
   #2 day, 0:00:00
   #3 day, 0:00:00
   print(base + timedelta(days=i))
#2022-11-15 11:00:00
#2022-11-16 11:00:00
#2022-11-17 11:00:00
#2022-11-18 11:00:00
```

These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

format code	meaning	example
%a	Abbreviated weekday name	Sun, Mon
%A	Full weekday name	Sunday, Monday
%w	Weekday as decimal number	0…6
%d	Day of the month as a zero-padded decimal	01, 02
%-d	day of the month as decimal number	1, 2..
%b	Abbreviated month name	Jan, Feb
%m	month as a zero padded decimal number	01, 02
%-m	month as a decimal number	1, 2
%B	Full month name	January, February
%y	year without century as a zero padded decimal number	99, 00
%-y	year without century as a decimal number	0, 99
%Y	year with century as a decimal number	2000, 1999
%H	hour(24 hour clock) as a zero padded decimal number	01, 23
%-H	hour(24 hour clock) as a decimal number	1, 23
%I	hour(12 hour clock) as a zero padded decimal number	01, 12

These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

%-I	**hour(12 hour clock) as a decimal number**	**1, 12**
%p	**locale's AM or PM**	**AM, PM**
%M	**Minute as a zero padded decimal number**	**01, 59**
%-M	**Minute as a decimal number**	**1, 59**
%S	**Second as a zero padded decimal number**	**01, 59**
%-S	**Second as a decimal number**	**1, 59**
%f	**microsecond as a decimal number, zero padded on the left side**	**000000, 999999**
%z	**UTC offset in the form +HHMM or -HHMM**	
%Z	**Time zone name**	
%j	**day of the year as a zero padded decimal number**	**001, 365**
%-j	**day of the year as a decimal number**	**1, 365**
%U	**Week number of the year (Sunday being the first)**	**0, 6**
%W	**Week number of the year**	**00, 53**
%c	**locale's appropriate date and time representation**	**Mon Sep 30 07:06:05 2013**
%x	**locale's appropriate date representation**	**11/30/98**

These notes are only to get basic Knowledge, you'll have to study extensively separately, refer class notes

%X	**locale's appropriate time representation**	**10:03:43**
%%	**A literal '%' character**	**%**