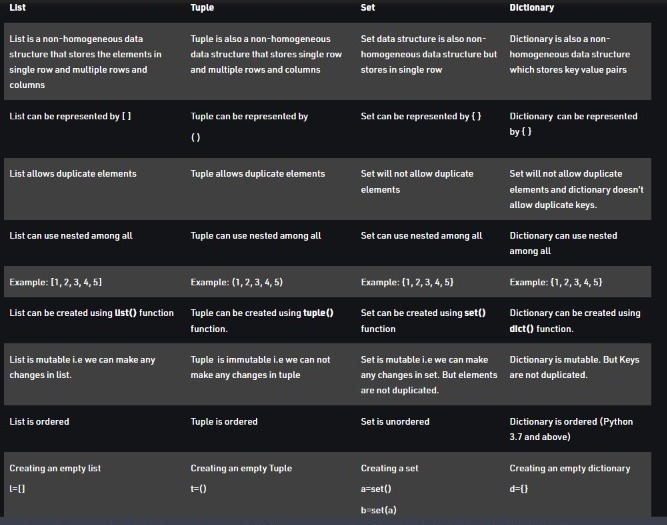
1. **Explain List, Tuple, Set, and Dictionary and provide at least one instance where each of these collection types can be used.**

**Ans.**

****

1. **Does Python allow you to program in a structured style?**

**Ans.**

**Structured Programming in Python**

Structured programming is a program written with only the three constructions **sequence**, **decision** (**if..elif** statements), and **repetition** (**while** or **for** statements). **Important:** the body of a Python **if**, **elif**, **while**, or **for** statement is indicated by indenting four spaces.  Python does not use end statements.

1. **Sequence.**   Lines or blocks of code are written and executed in sequential order.  
     
   ***Example:***
2. **x = 56**
3. **y = 11**
4. **z = x + y**
5. **print(z)**
6. **Decision.**   Execute a block of code (action) if a condition is true. The block of code is executed at most once.
7. **if condition:**
8. **action**

***Example:***

**if x % 2 == 0:**

**print("The number is even.")**

1. **Repetition.**   Repeat a block of code (action) while a condition is true. There is no limit to the number of times that the block can be executed.
2. **while condition:**
3. **action**

***Example:***

**n = 1**

**while n < 100:**

**print(n)**

**n = n \* n**

1. **What is PIP software in the Python world?**

**Ans.**

Package Installer for Python (pip) is **the de facto and recommended package-management system written in Python and is used to install and manage software packages**. It connects to an online repository of public packages, called the Python Package Index.

**Repository:**github.com/pypa/pip

**Operating system:**OS-independen

1. **What should be the typical build environment for Python-based application development?**

**Ans.**

**IDLE**. IDLE is an integrated development environment that is part of the Python standard distribution. It is completely written in Python and uses the Tkinter GUI toolkit.

A couple of common IDEs for Python development are **PyCharm and VSCode**, both of which runs on any major operating system.

**6.How does for Loop and while Loop differs in Python and when do you choose to use them?**

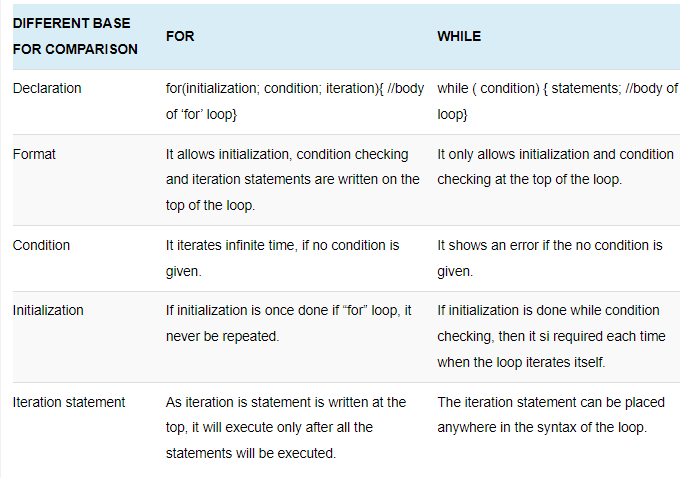
**Ans**

**For Loop**

For loop allows a programmer to execute a sequence of statements several times, it abbreviates the code which helps to manage loop variables.

**While loop**

While loop allows a programmer to repeat a single statement or a group of statements for the TRUE condition. It verifies the condition before executing the loop.

****

**7.How do you implement JSON given that Python is best suited for the server-side application?**

**Ans.**

: Python has built-in support to handle JSON objects. The user just has to import the JSON module and use the functions such as loads and dumps to convert the JSON string to a JSON object and vice versa.

**8.What is the purpose of \_init\_() function in Python?**

**Ans.** Constructor Concept:

1. Constructor is a special method in python.
2. The name of the constructor should be \_\_init\_\_(self)
3. Constructor will be executed automatically at the time of object creation.
4. The main purpose of constructor is to declare and initialize instance variables.
5. Per object constructor will be exeucted only once.
6. Constructor can take atleast one argument(atleast self)
7. Constructor is optional and if we are not providing any constructor then python will provide default constructor.

**9.What is the significance of the ‘self’ parameter in an object method? Should we always name this parameter as ‘self’?**

Self Variable: self is the default variable which is always pointing to current object (like this keyword in Java)

• By using self we can access instance variables and instance methods of object.

• Note:

1) self should be first parameter inside constructor def \_\_init\_\_(self):

2) self should be first parameter inside instance methods def talk(self):

**Should we always name this parameter as ‘self’?**

The reason why we use self is that Python does not use the '@' syntax to refer to instance attributes.

**10.How does Lambda function differ from a normal function in Python?**

**Ans.** Lambda Function, also referred to as 'Anonymous function' is same as a regular python function but can be defined without a name. While normal functions are defined using the def keyword, anonymous functions are defined using the lambda keyword. However ,they are restricted to single line of expression.

Lambda functions can only have one expression in their body. Regular functions can have multiple expressions and statements in their body. Lambdas do not have a name associated with them. That's why they are also known as anonymous functions.

lambda Function:

We can define by using lambda keyword lambda n:n\*n

Syntax of lambda Function: lambda argument\_list : expression

Note: By using Lambda Functions we can write very concise code so that readability of the pro

Q. Write a program to create a lambda function to find square of given number?

1) s=lambda n:n\*n

2) print("The Square of 4 is :",s(4))

3) print("The Square of 5 is :",s(5))

4) Output

6) The Square of 4 is : 16

7) The Square of 5 is : 25 gram will be improved

**11. How is Exception Handling done in Python?**

What is Exception:

An unwanted and unexpected event that disturbs normal flow of program is called exception.

Eg: ZeroDivisionError

TypeError

ValueError

FileNotFoundError

EOFError

SleepingError

TyrePuncturedError

It is highly recommended to handle exceptions. The main objective of exception handling is Graceful Termination of the program(i.e we should not block our resources and we should not miss anything)

Exception handling does not mean repairing exception. We have to define alternative way to continue rest of the program normally.

Eg: For example our programming requirement is reading data from remote file locating at London. At runtime if london file is not available then the program should not be terminated abnormally. We have to provide local file to continue rest of the program normally. This way of defining alternative is nothing but exception handling.

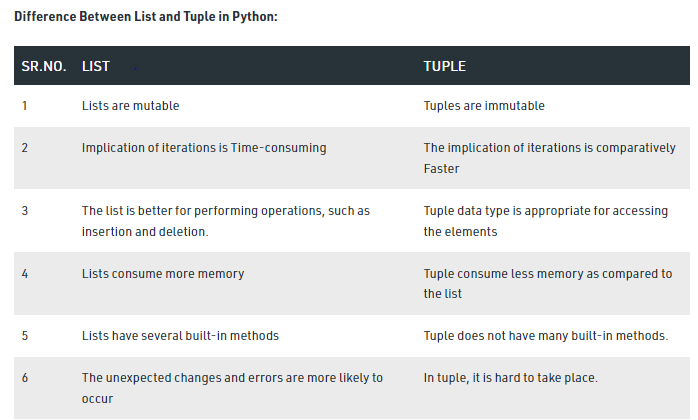
try:

read data from remote file locating at london

except FileNotFoundError:

use local file and continue rest of the program normally

1. **What are Python Tuples and how is it different from Lists?**

****

**13.What are membership operators in Python? Write an example to explain both.**

**Ans.**

We can use Membership operators to check whether the given object present in the given collection.(It may be String,List,Set,Tuple or Dict)

1. In- Returns True if the given object present in the specified Collection
2. not in -Retruns True if the given object not present in the specified Collection

Eg: 1) x="hello learning Python is very easy!!!"

2) print('h' in x) True

3) print('d' in x) False

4) print('d' not in x) True

5) print('Python' in x) True

**14.Write a code to display the current time.**

**Ans.**

from datetime import datetime

# Current date time in local system

print(datetime.now())

print(datetime.date(datetime.now()))

print(datetime.time(datetime.now()))

**15.What are help () and dir() in python?**

**Ans.**

**help ()**

We can find help for any module by using help() function

Eg: import math

help(math)

**dir()**

**Finding members of module by using dir() function:**

Python provides inbuilt function dir() to list out all members of current module or a specified module.

dir() ===>To list out all members of current module

dir(moduleName)==>To list out all members of specified module

Eg 1: test.py

1) x=10

2) y=20

3) def f1():

4) print("Hello")

5) print(dir()) # To print all members of current module

6)

7) Output

8) ['\_\_annotations\_\_', '\_\_builtins\_\_', '\_\_cached\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_nam e\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'f1', 'x', 'y']

**16.What does the term ‘Monkey Patching’ refer to in Python?**

**ANS.**

* + Monkey Patching is an exciting topic of Python. Monkey-patching is a term that refers to modifying a class or module at a run time. In simple words, a class or module's work can be changed at the runtime
  + In Python, the term monkey patch refers to dynamic (or run-time) modifications of a class or module. In Python, we can actually change the behavior of code at run-time.

**# monk.py**

**class A:**

**def func(self):**

**print ("func() is being called")**

We use above module (monk) in below code and change behavior of func() at run-time by assigning different value.

**import monk**

**def monkey\_f(self):**

**print ("monkey\_f() is being called")**

**# replacing address of "func" with "monkey\_f"**

**monk.A.func = monkey\_f**

**obj = monk.A()**

**# calling function "func" whose address got replaced**

**# with function "monkey\_f()"**

**obj.func()**

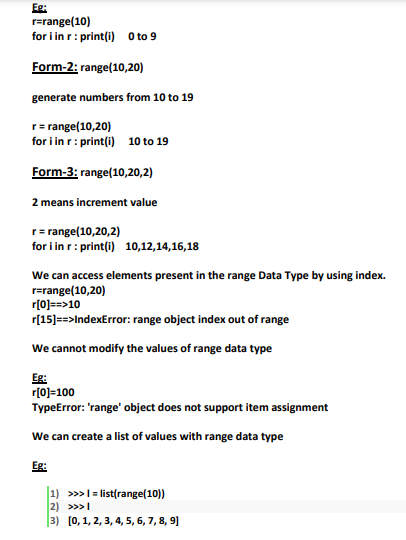
**17.What is range () in Python?**

range Data Type:

range Data Type represents a sequence of numbers.

The elements present in range Data type are not modifiable.

i.e range Data type is immutable.

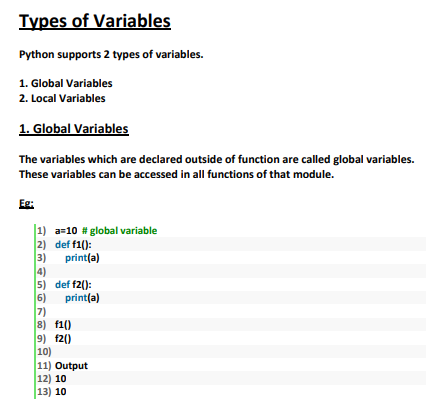
****

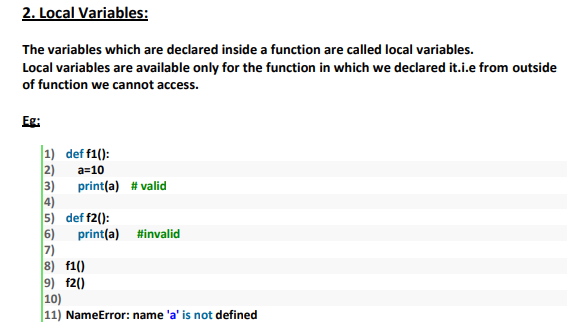
**18. What is a from import statement and write the syntax for it?**

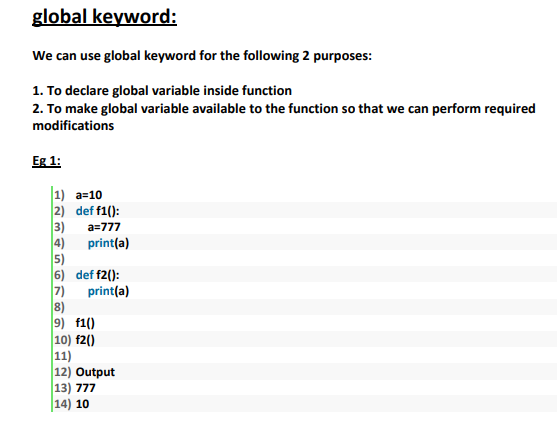
The Python import statement lets you import a module into your code. A module is a file that contains functions and values that you can reference from your program.

The import statement syntax is: import modulename.

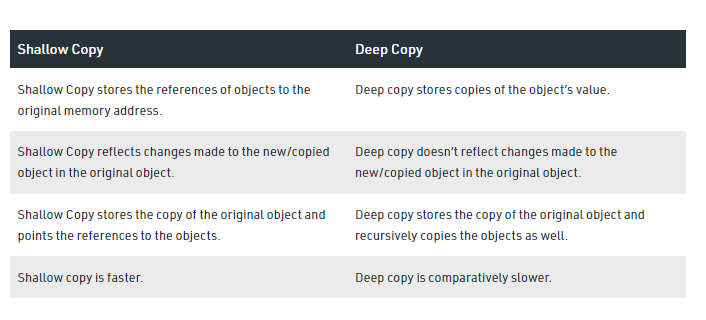
**19. What is the difference between locals() and globals()?**

****

****

****

**20. What is the difference between a shallow copy and deep copy?**

****

**21. What does PEP8 refer to?**

**Ans.**

PEP 8, sometimes spelled PEP8 or PEP-8, is a document that provides guidelines and best practices on how to write Python code. It was written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan. The primary focus of PEP 8 is to improve the readability and consistency of Python code.

PEP stands for Python Enhancement Proposal, and there are several of them. A PEP is a document that describes new features proposed for Python and documents aspects of Python, like design and style, for the community.

This tutorial outlines the key guidelines laid out in PEP 8. It’s aimed at beginner to intermediate programmers, and as such I have not covered some of the most advanced topics. You can learn about these by reading the full [PEP 8](https://www.python.org/dev/peps/pep-0008/) documentation.

**22.What are \*args and \*kwargs?**

**Ans.**

# Python \*args and \*\*kwargs

In this article, we will learn about Python \*args and \*\*kwargs ,their uses and functions with examples.

In programming, we define a function to make a reusable code that performs similar operation. To perform that operation, we call a function with the specific value, this value is called a function argument in Python.

We would recommend you to read [Python Function](https://www.programiz.com/python-programming/function) and [Python Function Arguments](https://www.programiz.com/python-programming/function-argument).

Suppose, we define a function for addition of 3 numbers.

### Example 1: Function to add 3 numbers

def adder(x,y,z):

print("sum:",x+y+z)

adder(10,12,13)

When we run the above program, the output will be

sum: 35

In above program we have adder() function with three arguments x, y and z. When we pass three values while calling adder() function, we get sum of the 3 numbers as the output.

Lets see what happens when we pass more than 3 arguments in the adder() function.

def adder(x,y,z):

print("sum:",x+y+z)

adder(5,10,15,20,25)

When we run the above program, the output will be

TypeError: adder() takes 3 positional arguments but 5 were given

In the above program, we passed 5 arguments to the adder() function instead of 3 arguments due to which we got TypeError.

## Introduction to \*args and \*\*kwargs in Python

In Python, we can pass a variable number of arguments to a function using special symbols. There are two special symbols:

1. \*args (Non Keyword Arguments)
2. \*\*kwargs (Keyword Arguments)

We use \*args and \*\*kwargs as an argument when we are unsure about the number of arguments to pass in the functions.

## Python \*args

As in the above example we are not sure about the number of arguments that can be passed to a function. Python has \*args which allow us to pass the variable number of non keyword arguments to function.

In the function, we should use an asterisk \* before the parameter name to pass variable length arguments.The arguments are passed as a tuple and these passed arguments make tuple inside the function with same name as the parameter excluding asterisk \*.

### Example 2: Using \*args to pass the variable length arguments to the function

def adder(\*num):

sum = 0

for n in num:

sum = sum + n

print("Sum:",sum)

adder(3,5)

adder(4,5,6,7)

adder(1,2,3,5,6)

When we run the above program, the output will be

Sum: 8

Sum: 22

Sum: 17

In the above program, we used \*num as a parameter which allows us to pass variable length argument list to the adder() function. Inside the function, we have a loop which adds the passed argument and prints the result. We passed 3 different tuples with variable length as an argument to the function.

## Python \*\*kwargs

Python passes variable length non keyword argument to function using \*args but we cannot use this to pass keyword argument. For this problem Python has got a solution called \*\*kwargs, it allows us to pass the variable length of keyword arguments to the function.

In the function, we use the double asterisk \*\* before the parameter name to denote this type of argument. The arguments are passed as a dictionary and these arguments make a dictionary inside function with name same as the parameter excluding double asterisk \*\*.

### Example 3: Using \*\*kwargs to pass the variable keyword arguments to the function

def intro(\*\*data):

print("\nData type of argument:",type(data))

for key, value in data.items():

print("{} is {}".format(key,value))

intro(Firstname="Sita", Lastname="Sharma", Age=22, Phone=1234567890)

intro(Firstname="John", Lastname="Wood", Email="johnwood@nomail.com", Country="Wakanda", Age=25, Phone=9876543210)

**23. What is namespace in Python?**

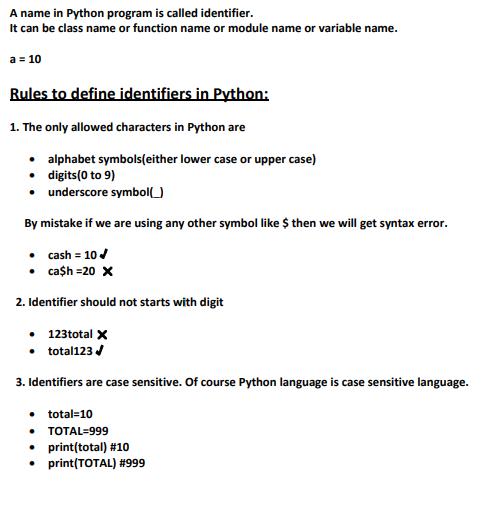
Ans.

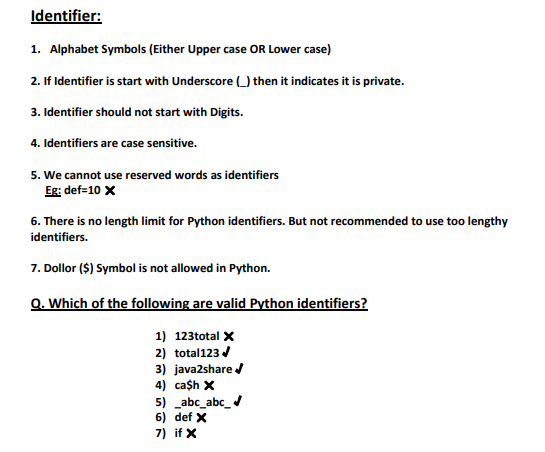
Namespaces in Python. A namespace is a collection of currently defined symbolic names along with information about the object that each name references. You can think of a namespace as a dictionary in which the keys are the object names and the values are the objects themselves

**24. What does len() do?**

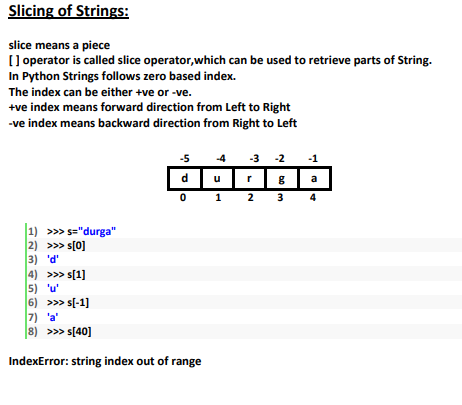
The len() function returns the number of items in an object. When the object is a string, the len() function returns the number of characters in the string.

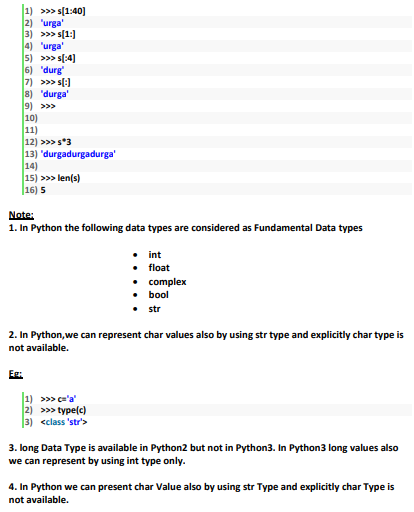
**25. What are the rules for legal Python names?**





**26. Define “slicing”.**

****

****

**27.What is a “negative index”?**

## Negative Indexing

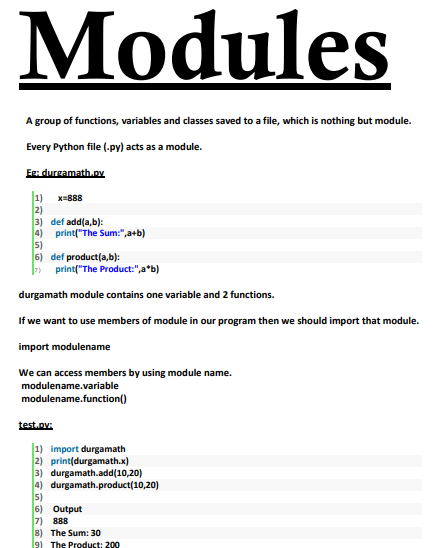
Use negative indexes to start the slice from the end of the string:

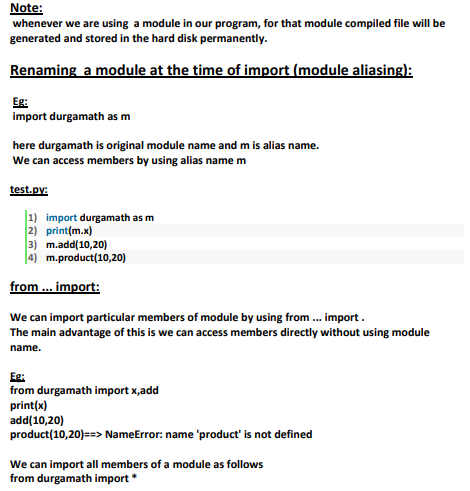
### Example

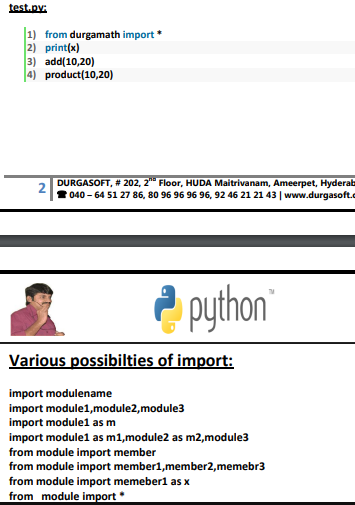
Get the characters from position 5 to position 1, starting the count from the end of the string:

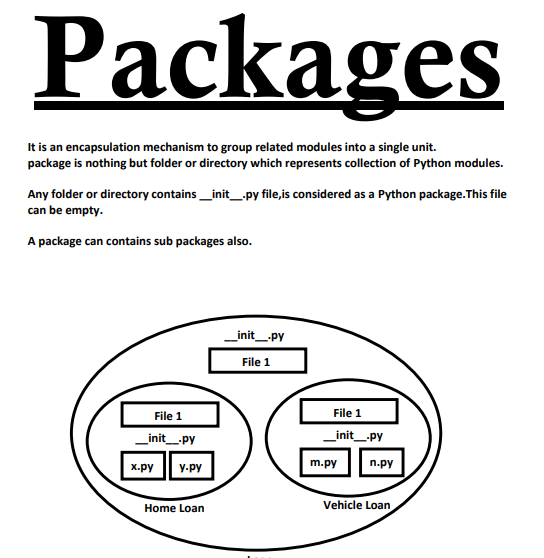
b = "Hello, World!"  
print(b[-5:-2])

**28. Define “module” and “package”.**

****

****

****

****