Python Class

#Python Notes

Functions in Python

Function in Python

- Function is a group of related statements that perform a specific task.
- **Syntax :-**

```
def function_name(parameters):
    statement(s)
return [expression]
```

Cont…

- Here,
- **def** keyword marks the start of function header.
- A **function name** to uniquely identify it.
- A **colon (:)** to mark the end of function header.
- The **function body**.
- An optional **return statement** to return a value from the function.

Example

File Edit View Navigate Code Help

scratch.py

```
def my_function():
    print("Hello World!")

my_function()
my_function()
```

Run: scratch

```
Hello World!
Hello World!

Process finished with exit code 0
```

Example With Argument

File Edit View Navigate Code Help

scratch.py

```
def my_function(name):
    print("Name is : ",name)

my_function("Akash Techolabs")
```

Run: scratch

```
Name is :  Akash Techolabs

Process finished with exit code 0
```

Example with return Statement

File Edit View Navigate Code Help

scratch.py

```python
def my_function(name):
    return name

name = my_function("Akash Techolabs")
print(name)
```

Run: scratch

```
Akash Techolabs

Process finished with exit code 0
```

Example With Multiple Return Statement

File Edit View Navigate Code Help

scratch.py

```python
def my_function():
    name = "Akash Technolabs"
    contact = 9978621654
    return name,contact

name,contact = my_function()
print("Name :" ,name)
print("Contact :",contact)
```

```
Name : Akash Technolabs

Contact : 9978621654
```

Python function arguments

- There are three types of Python function arguments using which we can call a function.
  - Default Arguments
  - Keyword Arguments
  - Variable-length Arguments

Python Default Arguments

- Sometimes we may want to use parameters in a function that takes default values in case the user doesn’t want to provide a value for them.

Example

File Edit View Navigate Code Help

scratch.py

```python
def sum(a=5, b=7):
    """ This function will print sum of two numbers
        if the arguments are not supplied
        it will add the default value """
    print (a+b)

sum(10,20)  #calling with arguments
sum( )     #calling without arguments
```

```
30
12
```

Python Keyword Arguments

- In function, the values passed through arguments are assigned to parameters in order, by their position.
- With Keyword arguments, we can use the name of the parameter irrespective of its position while calling the function to supply the values.

Example

File Edit View Navigate Code Help

scratch.py

```python
def sum(a, b):

  print ("Sum is :",a+b)

sum(b = 10,a = 20)   #calling with arguments
```

```
Sum is : 30
```

Variable-length Arguments

- Sometimes you may need more arguments to process function then you mentioned in the definition.
- If we don’t know in advance about the arguments needed in function, we can use variable-length arguments also called arbitrary arguments.

Cont..

- We can pass a variable number of arguments to a function using special symbols. There are two special symbols:
  - * (Non Keyword Arguments)
  - ** (Keyword Arguments)
- An asterisk (*) is placed before a parameter in function definition which can hold non-keyworded variable-length arguments.
- A double asterisk (**) is placed before a parameter in function which can hold keyworded variable-length arguments.

Example (Non Keyword Arguments)

File Edit View Navigate Code Help

scratch.py

```python
def add(*num):
    sum = 0

    for n in num:
        sum = sum + n
    print("Sum:",sum)

add(10,20)
add(10,20,30)
```

```
Sum: 30
Sum: 60
```

Example (Keyword Arguments)

scratch.py

```python
def my_func(**arg):
    for i, j in arg.items():
        print(i, j)

my_func(Name='Akash', Lastname='Padhiyar')
```

```
Name Akash
Lastname Padhiyar
```

Scope of Variables

- There are two basic scopes of variables in Python –
  1. Global variables
  2. Local variables

- Variables that are defined inside a function body have a local scope, and those defined outside have a global scope.

Example

File Edit View Navigate Code Help

scratch.py

```python
def my_func():
    x = 10
    print("Value inside function:",x)

x = 20
my_func()
print("Value outside function:",x)
```

Run: scratch

```
Value inside function: 10
Value outside function: 20

Process finished with exit code 0
```

Significance of Indentation (Space) in Python

- We take a simple example with "print" command. When we write "print" function right below the def my_function ():
- It will show an "**indentation error: expected an indented block**".

Example

File Edit View Navigate Code Help

scratch.py

```
1  def my_function():
2  print("Hello World!")
3
4  my_function()
5  my_function()
6
```

Run: scratch

```
  File "C:/Users/[redacted]/.PyCharmEdu2019.1/config/scratches/scratch.py", line 2
    print("Hello World!")
        ^
IndentationError: expected an indented block

Process finished with exit code 1
```

Modules in Python

- A file containing Python code, **for example : scratch.py, is called a module and its module name would be scratch.**
- We use modules to break down large programs into small manageable and organized files.
- We can define our most used functions in a module and import it, instead of copying their definitions into different programs.

The import Statement

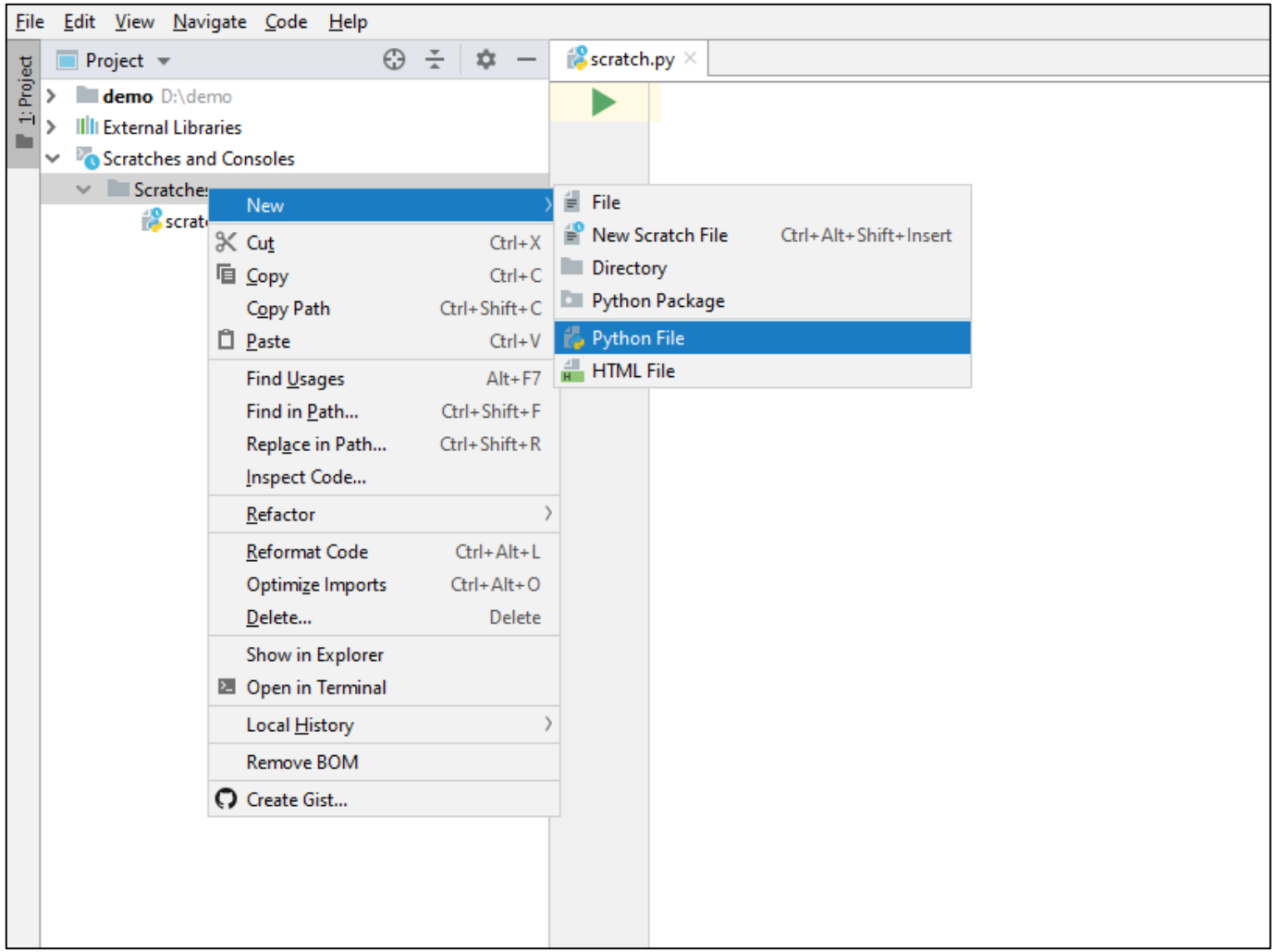
- The ***import*** keyword is used to import module to another module
- **Syntax :-**

```
import module_name
```

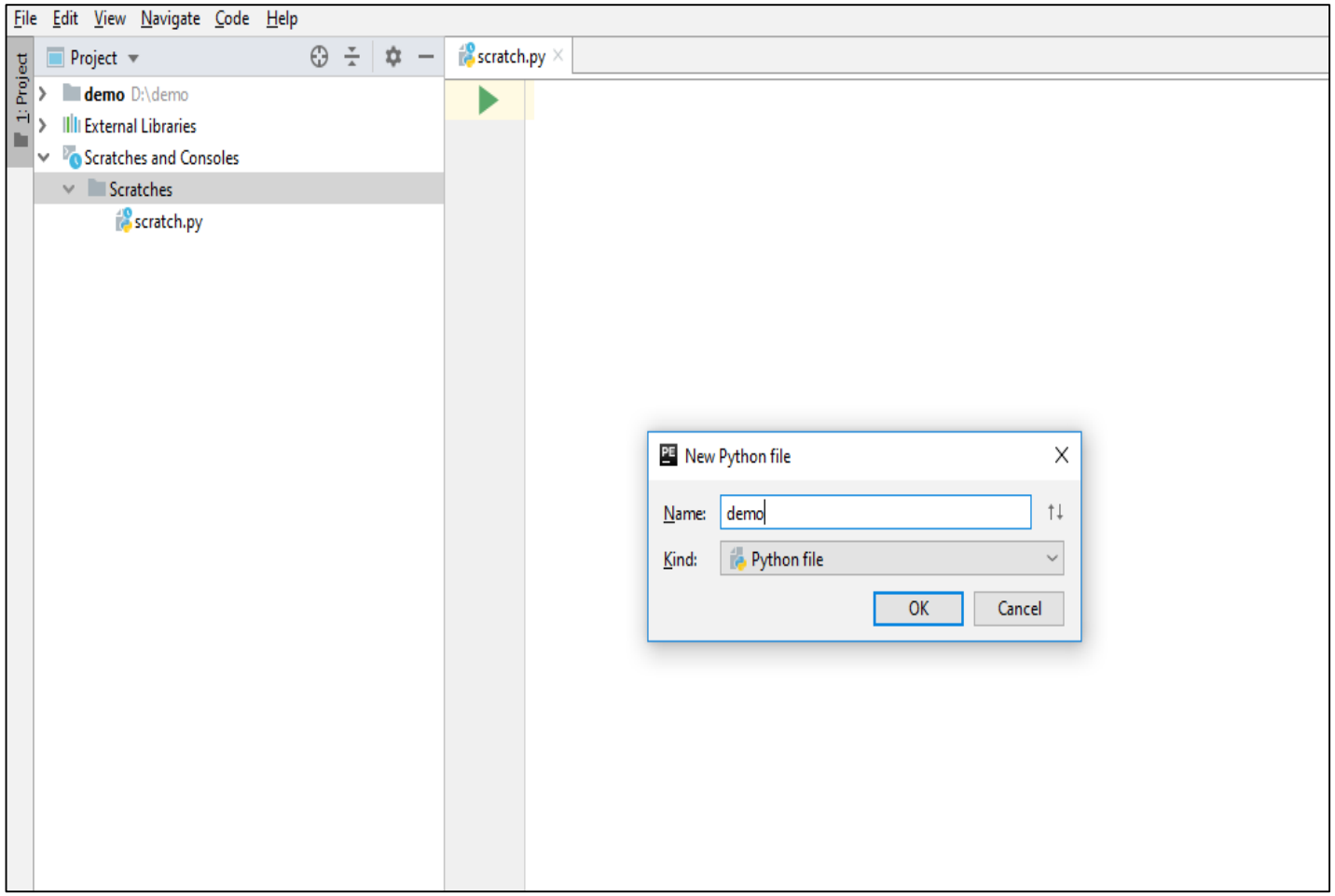
Example

- In the following example we create 2 files :-
  - scratch.py
  - demo.py
- Scratch.py file contain user define function my_function() that we will import in demo.py file.

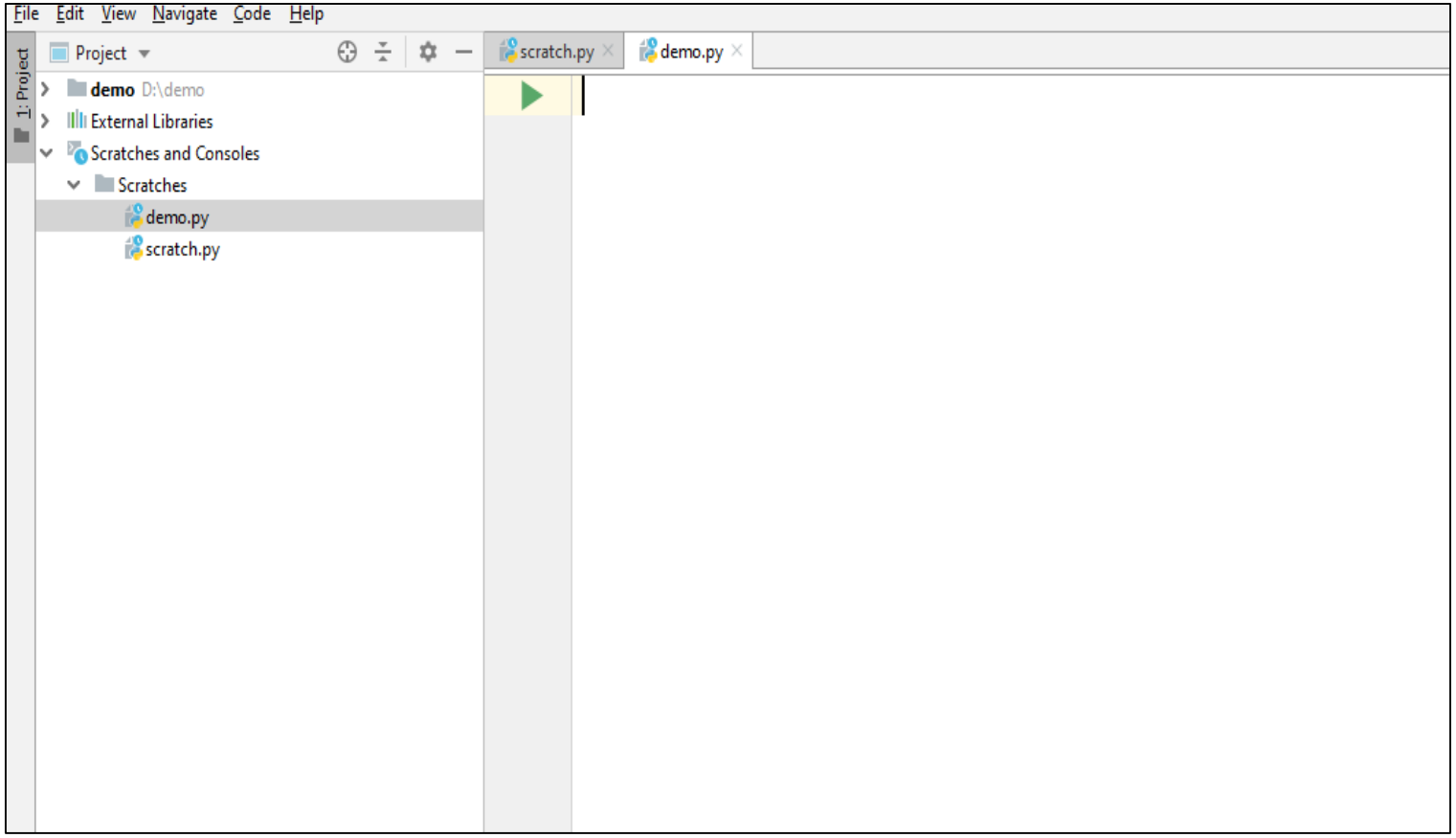
Create a new file by right clicking on Scratches folder



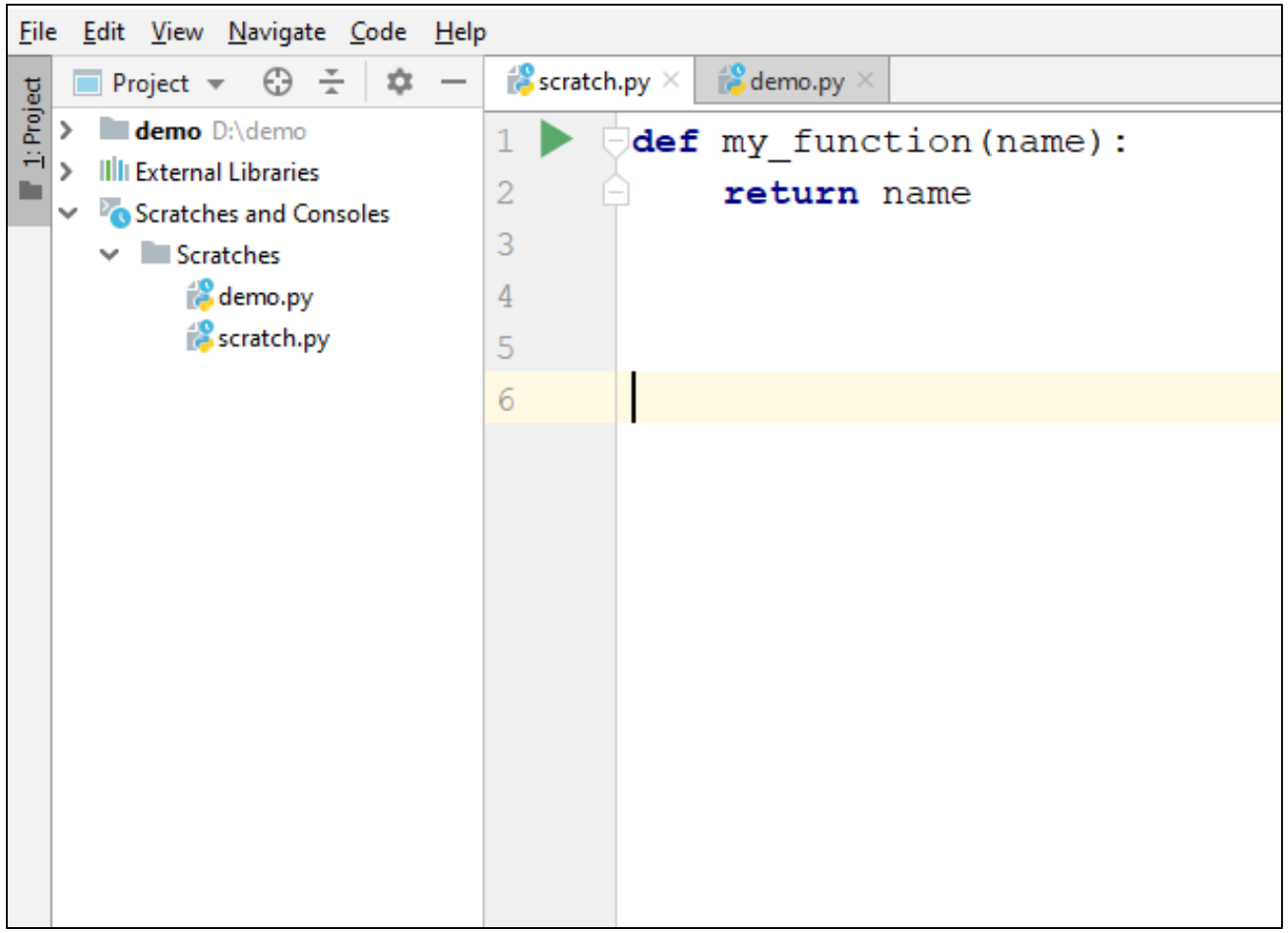
Give file name



File is created



scratch.py


File Edit View Navigate Code Help
1: Project
Project
demo D:\demo
External Libraries
Scratches and Consoles
Scratches
demo.py
scratch.py
scratch.py
demo.py
def my_function(name):
return name

demo.py

File Edit View Navigate Code Help

Project

demo D:\demo
External Libraries
Scratches and Consoles
Scratches
demo.py
scratch.py

scratch.py | demo.py

```
#import scratch module
import scratch

name = scratch.my_function("Akash Technolabs")
print(name)
```

Run: demo

```
Akash Technolabs

Process finished with exit code 0
```

Operators in Python

Operators in Python

- Arithmetic Operators
- Comparison operators
- Logical operators
- Assignment operators
- Membership Operators
- Identity Operators

Arithmetic Operators

Operator	Meaning
+	Add two operands or unary plus
-	Subtract right operand from the left or unary minus
*	Multiply two operands
/	Divide left operand by the right one (always results into float)
%	Modulus - remainder of the division of left operand by the right
//	Floor division - division that results into whole number adjusted to the left in the number line
**	Exponent - left operand raised to the power of right(x to the power y)

Example

File Edit View Navigate Code Help

scratch.py

```python
x = 10
y = 6

print('x + y =', x + y)

print('x - y =', x - y)

print('x * y =', x * y)

print('x / y =', x / y)

print('x // y =', x // y)

print('x ** y =', x ** y)
```

Run: scratch

```
x + y = 16
x - y = 4
x * y = 60
x / y = 1.6666666666666667
x // y = 1
x ** y = 1000000
```

Comparison operators

Operator	Meaning	Example
>	Greater than	x > y
<	Less than	x < y
==	Equal to	x == y
!=	Not equal to	x != y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Example

File Edit View Navigate Code Help

scratch.py

```python
x = 10
y = 6

print('x > y  is', x > y)

print('x < y  is', x < y)

print('x == y is', x == y)

print('x != y is', x != y)

print('x >= y is', x >= y)

print('x <= y is', x <= y)
```

Run: scratch

```
x > y  is True
x < y  is False
x == y is False
x != y is True
x >= y is True
x <= y is False
```

Logical operators

Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

Example Of and

Scrach.py

C: > Users > Devanshi > Desktop > Scrach.py > ...

```python
n1=10
n2=20
n3=30

if n1 > n2 and n1 > n3:
    print("n1 is the largest number")

if n2 > n1 and n2 > n3:
    print("n2 is the largest number.")

if n3 > n1 and n3 > n2:
    print("n3 is the largest number.")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Devanshi> & python c:/Users/Devanshi/Desktop/Scrach.py
n3 is the largest number.
PS C:\Users\Devanshi>
```

Example of or

Scrach.py

C: > Users > Devanshi > Desktop > Scrach.py > ...

```python
ch = input("Enter a character: ")

if(ch=='A' or ch=='a' or ch=='E' or ch =='e' or ch=='I'
    or ch=='i' or ch=='O' or ch=='o' or ch=='U' or ch=='u'):
        print(ch, "is a Vowel")
else:
    print(ch, "is a Consonant")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

```
PS C:\Users\Devanshi> & python c:/Users/Devanshi/Desktop/Scrach.py
Enter a character: a
a is a Vowel
PS C:\Users\Devanshi>
```

Assignment operators

Operator	Example	Equivatent to
=	x = y	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5

Membership Operators

- There are two membership operators that are used in Python.
- They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

Operator	Meaning
in	True if value/variable is found in the sequence
not in	True if value/variable is not found in the sequence

Example

File Edit View Navigate Code Help

scratch.py

```python
x = 10
y = 6
list1 = [10, 20, 30, 40, 50]

print(x in list1)

print(y in list1)

print(y not in list1)
```

Run: scratch

```
True
False
True

Process finished with exit code 0
```

Identity Operators

- To compare the memory location of two objects, Identity Operators are used. The two identify operators used in Python.

Operator	Meaning
is	True if the operands are identical (refer to the same object)
is not	True if the operands are not identical (do not refer to the same object)

Example

File Edit View Navigate Code Help

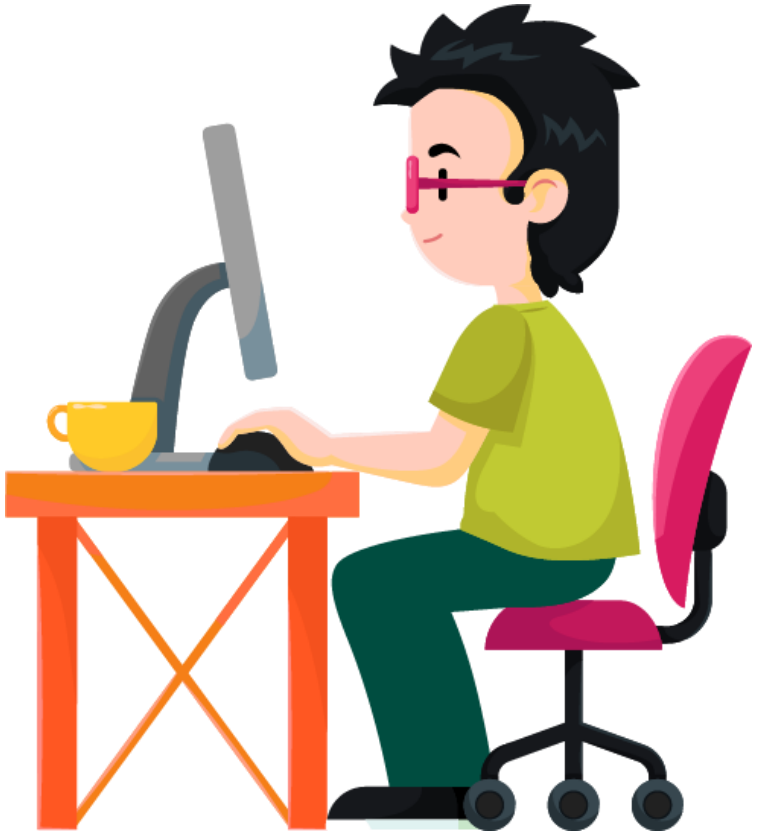
scratch.py

```
x = 20
y = 20

print(x is y)

print(x is not y)
```

Run: scratch

```
True
False

Process finished with exit code 0
```

Get Exclusive Video Tutorials



www.aptutorials.com

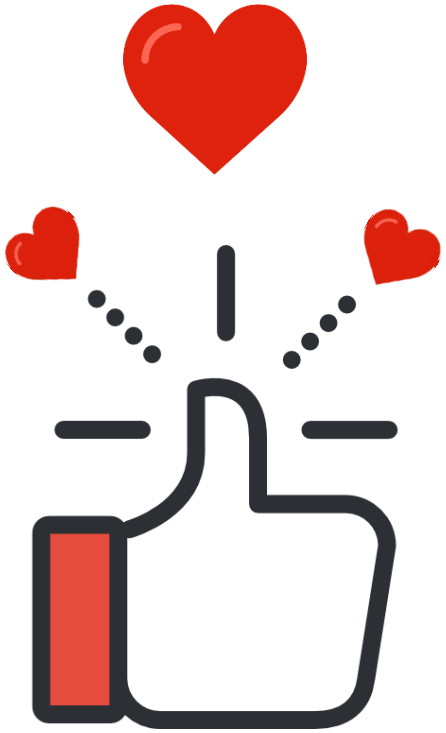
https://www.youtube.com/user/Akashtips

Get More Details

www.akashsir.com

If You Liked It !

Rating Us Now



Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET

Sulekha

https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad

Connect With Me

Akash Padhiyar
#AkashSir

www.akashsir.com
www.akashtechnolabs.com
www.akashpadhiyar.com
www.aptutorials.com

Social Info

Akash.padhiyar

Akashpadhiyar

Akash_padhiyar

+91 99786-21654

#Akashpadhiyar
#aptutorials