



Python Class

#Python Notes

Comments, Variable & Datatype

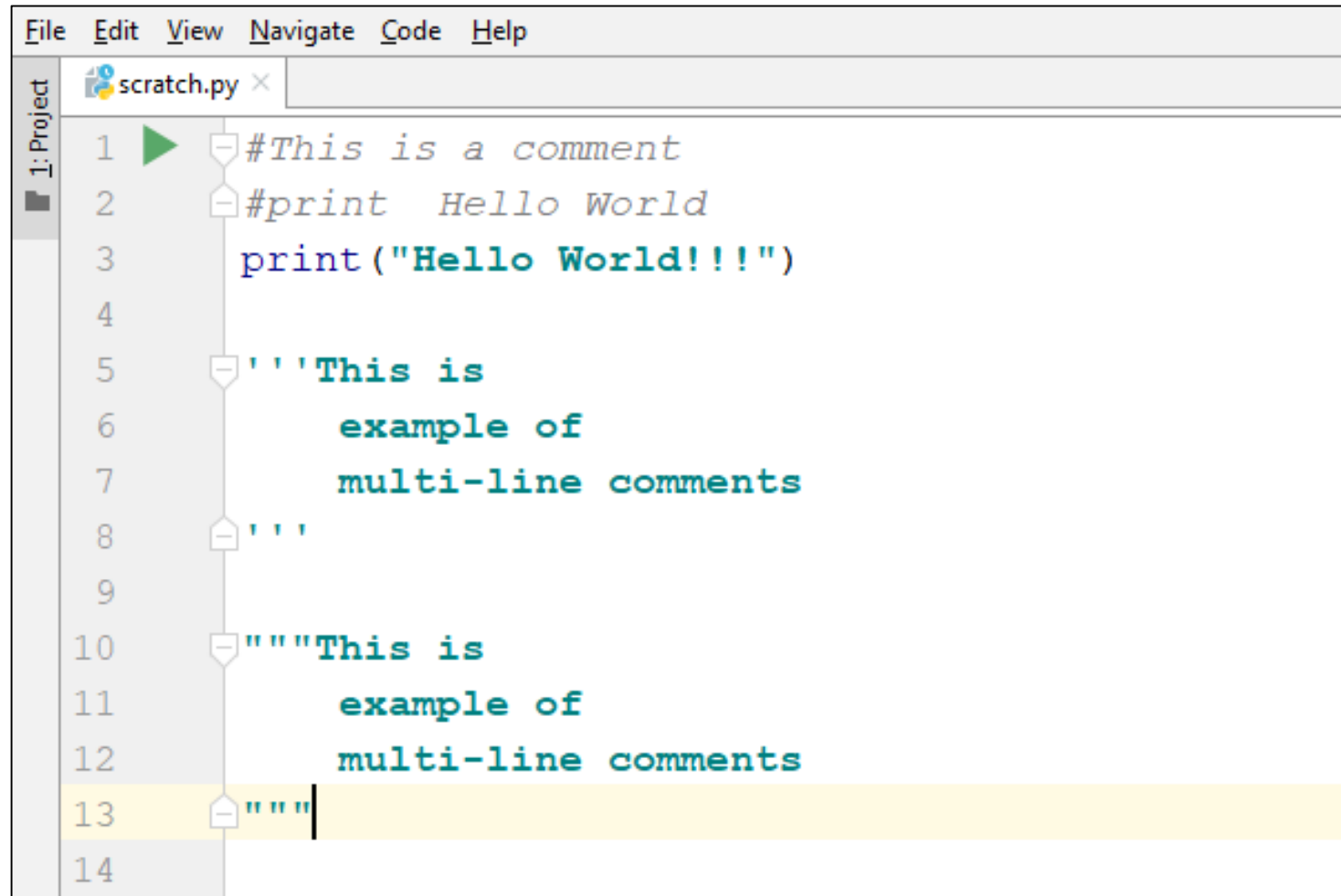


Comments

- In Python, we use the **hash (#) symbol** to start writing a comment.
- Another way of doing this is to use triple quotes, either **''' or '''** or **""" or """**.



Example



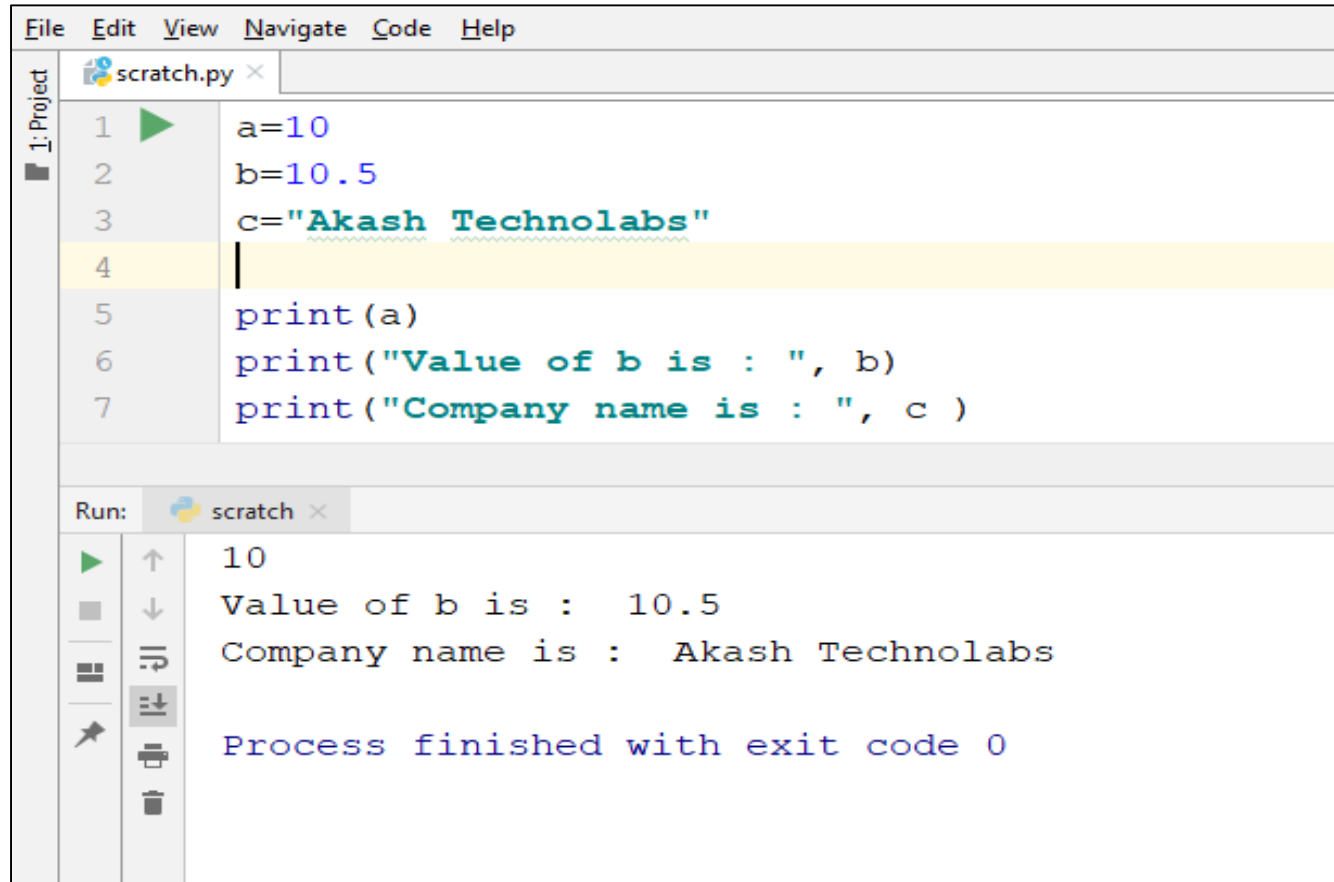
```
File Edit View Navigate Code Help
scratch.py x
1  #This is a comment
2  #print Hello World
3  print("Hello World!!!")
4
5  '''This is
6      example of
7      multi-line comments
8  '''
9
10 """"This is
11      example of
12      multi-line comments
13 """"
14
```

Variable in Python



How to Declare and use a Variable

- A variable is a named location used to store data in the memory.



The screenshot shows a Python IDE window with a menu bar (File, Edit, View, Navigate, Code, Help) and a tab for 'scratch.py'. The code editor contains the following Python code:

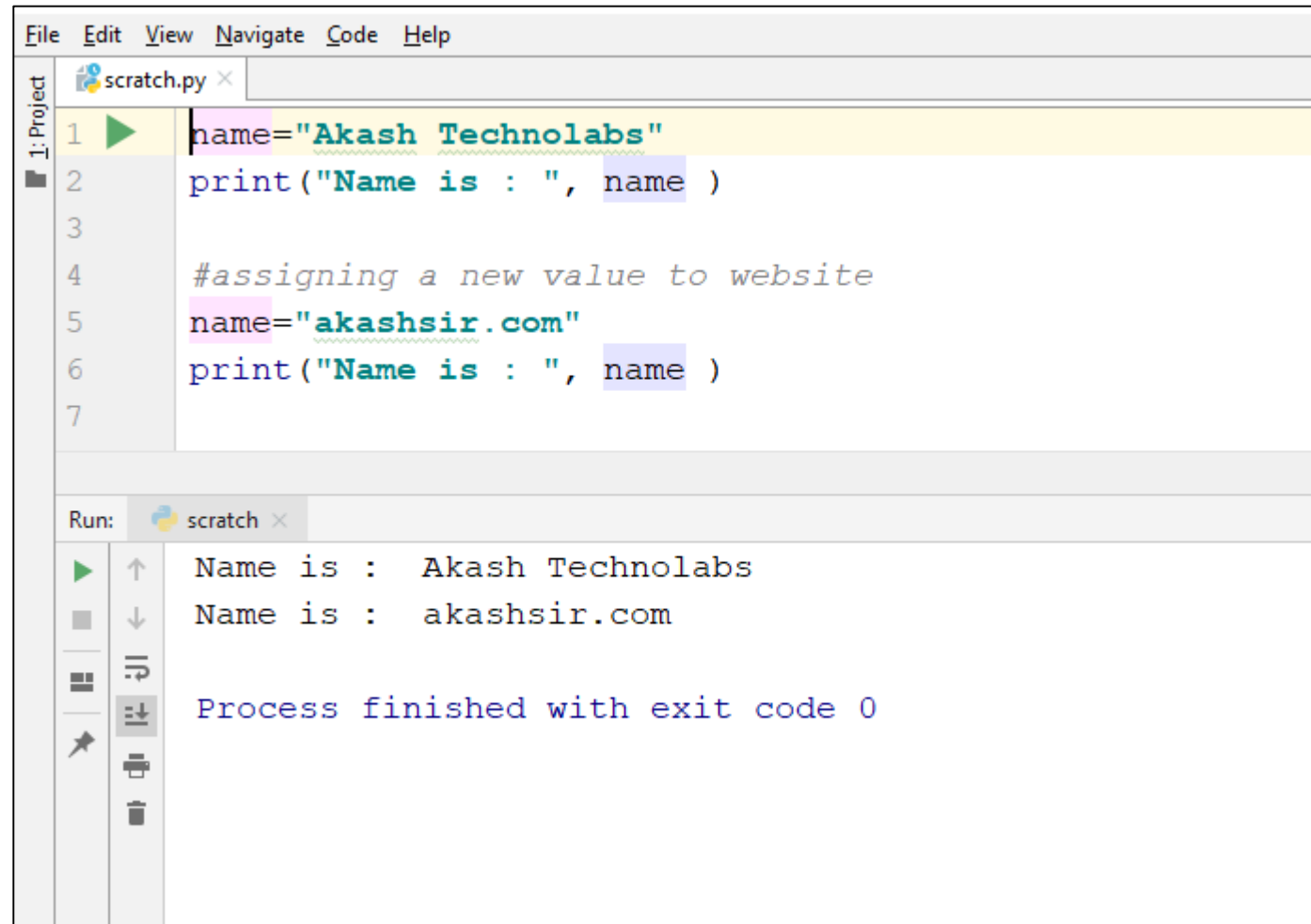
```
1 a=10
2 b=10.5
3 c="Akash Technolabs"
4
5 print(a)
6 print("Value of b is : ", b)
7 print("Company name is : ", c )
```

Below the code editor is a 'Run' panel with a tab for 'scratch'. It shows the output of the executed code:

```
10
Value of b is : 10.5
Company name is : Akash Technolabs

Process finished with exit code 0
```

Changing the value of a variable



The screenshot shows a Python IDE with a file named 'scratch.py'. The code in the editor is as follows:

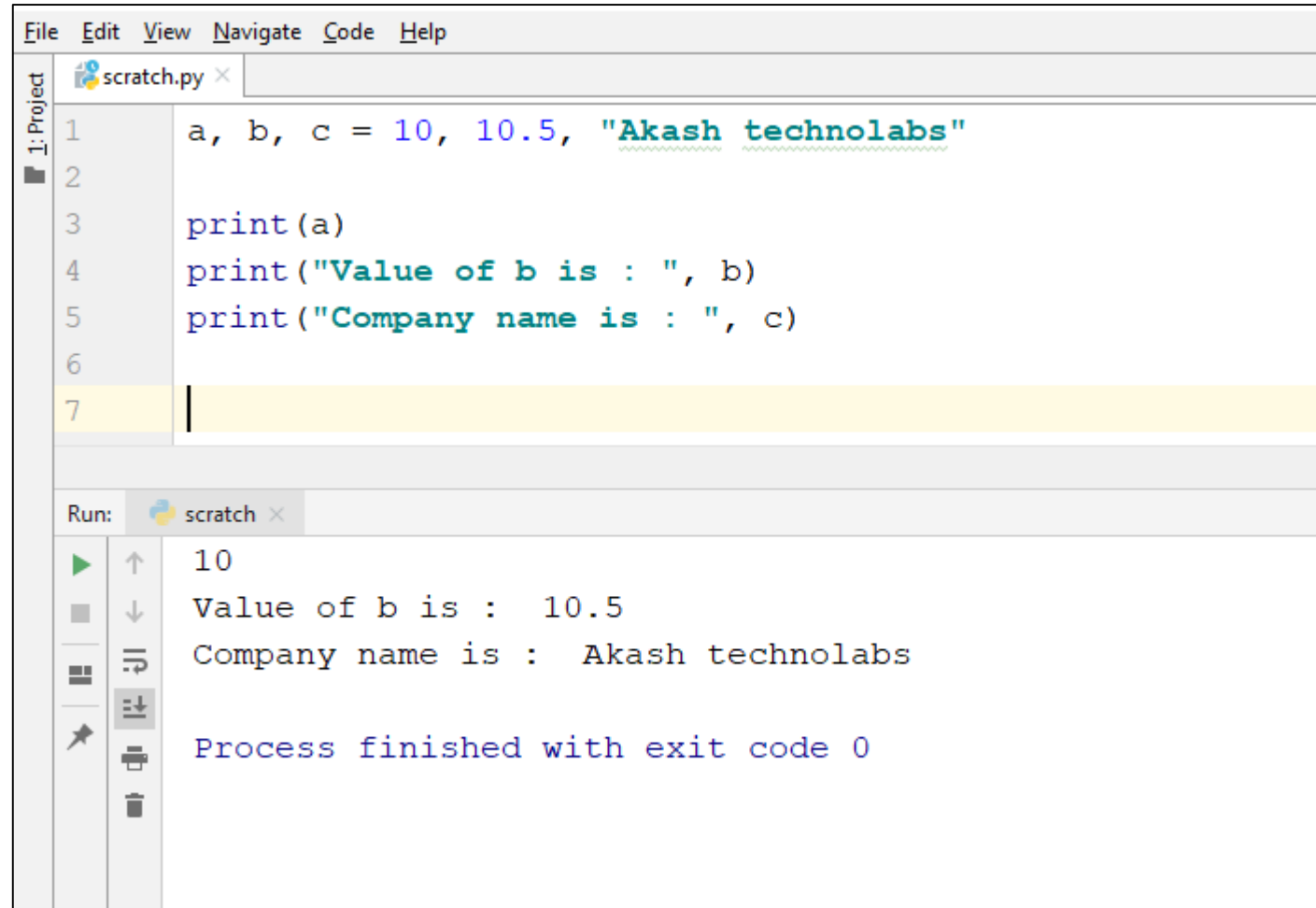
```
1 name="Akash Technolabs"
2 print("Name is : ", name )
3
4 #assigning a new value to website
5 name="akashsir.com"
6 print("Name is : ", name )
7
```

Below the editor, the 'Run' window shows the output of the program:

```
Name is : Akash Technolabs
Name is : akashsir.com

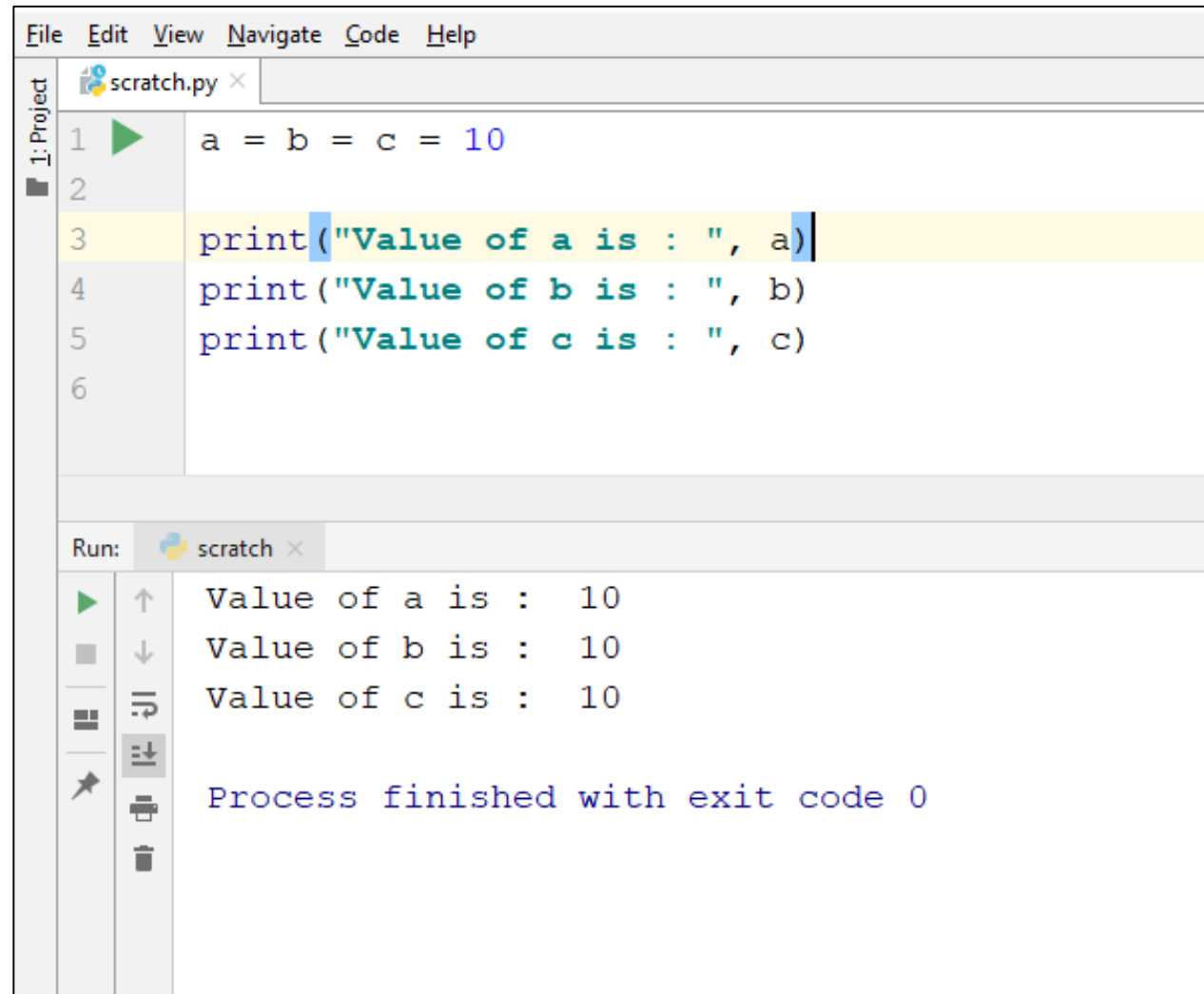
Process finished with exit code 0
```

Assigning multiple values to multiple variables



```
File Edit View Navigate Code Help
scratch.py x
1: Project
1 a, b, c = 10, 10.5, "Akash technolabs"
2
3 print(a)
4 print("Value of b is : ", b)
5 print("Company name is : ", c)
6
7
Run: scratch x
10
Value of b is : 10.5
Company name is : Akash technolabs
Process finished with exit code 0
```


Assign the same value to multiple variables



The screenshot shows a Python IDE window with a menu bar (File, Edit, View, Navigate, Code, Help) and a tab for 'scratch.py'. The code editor contains the following Python code:

```
1 a = b = c = 10
2
3 print("Value of a is : ", a)
4 print("Value of b is : ", b)
5 print("Value of c is : ", c)
6
```

Below the code editor is a 'Run' panel with a tab for 'scratch'. It shows the output of the code execution:

```
Value of a is : 10
Value of b is : 10
Value of c is : 10

Process finished with exit code 0
```

Data Types in Python



Data Types

- Python has 5 standard data types namely.
 - a. Numbers
 - b. String
 - c. List
 - d. Tuple
 - e. Dictionary
 - f. Boolean – In Python True and False are boolean literals. But the following values are also considered as false.

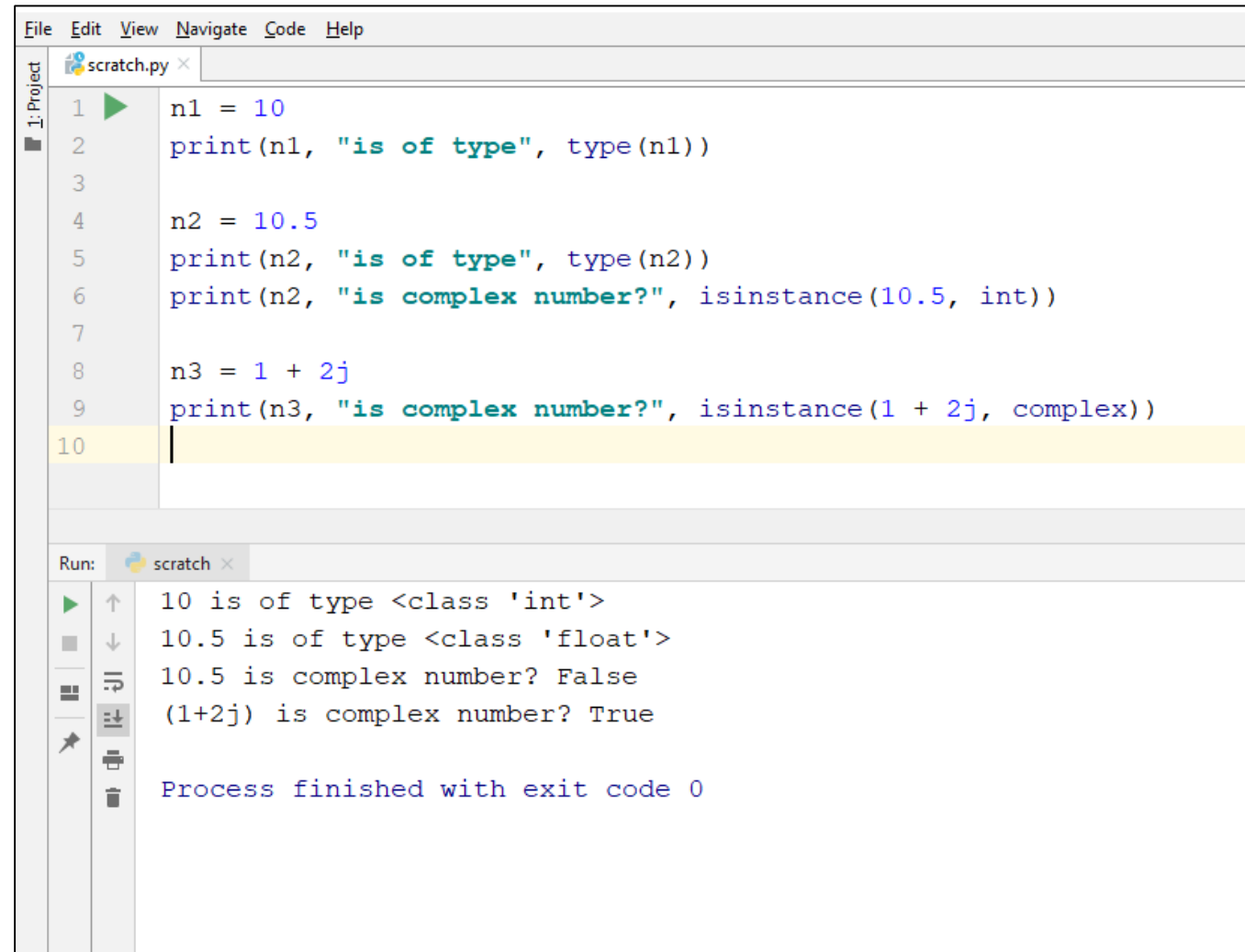


Number Datatype

- Number data types store numeric values. Integers, floating point numbers and complex numbers falls under number datatype.
- They are defined as int, float and complex class in Python.
- **type()** function is used to know which class a variable or a value belongs to.
- The **isinstance()** function is used to check if an object belongs to a particular class.



Example



The screenshot shows a Python IDE with a file named 'scratch.py' open. The code in the editor is as follows:

```
1 n1 = 10
2 print(n1, "is of type", type(n1))
3
4 n2 = 10.5
5 print(n2, "is of type", type(n2))
6 print(n2, "is complex number?", isinstance(10.5, int))
7
8 n3 = 1 + 2j
9 print(n3, "is complex number?", isinstance(1 + 2j, complex))
10
```

Below the editor, the 'Run' console shows the output of the code:

```
10 is of type <class 'int'>
10.5 is of type <class 'float'>
10.5 is complex number? False
(1+2j) is complex number? True

Process finished with exit code 0
```

String Datatype

- Strings are defined either with a single quote or a double quotes.



```
1 name = "Akash Technolabs "  
2  
3 # Prints complete string  
4 print("Name is : ", name)  
5  
6 # Prints first character of the string  
7 print(name[0])  
8  
9 # Prints characters starting from 3rd to 5th  
10 print(name[2:5])  
11  
12 # Prints string starting from 3rd character  
13 print(name[2:])  
14  
15 # Prints string two times  
16 print(name * 2)  
17  
18 # Prints concatenated string  
19 print(name + "Hello")  
20
```

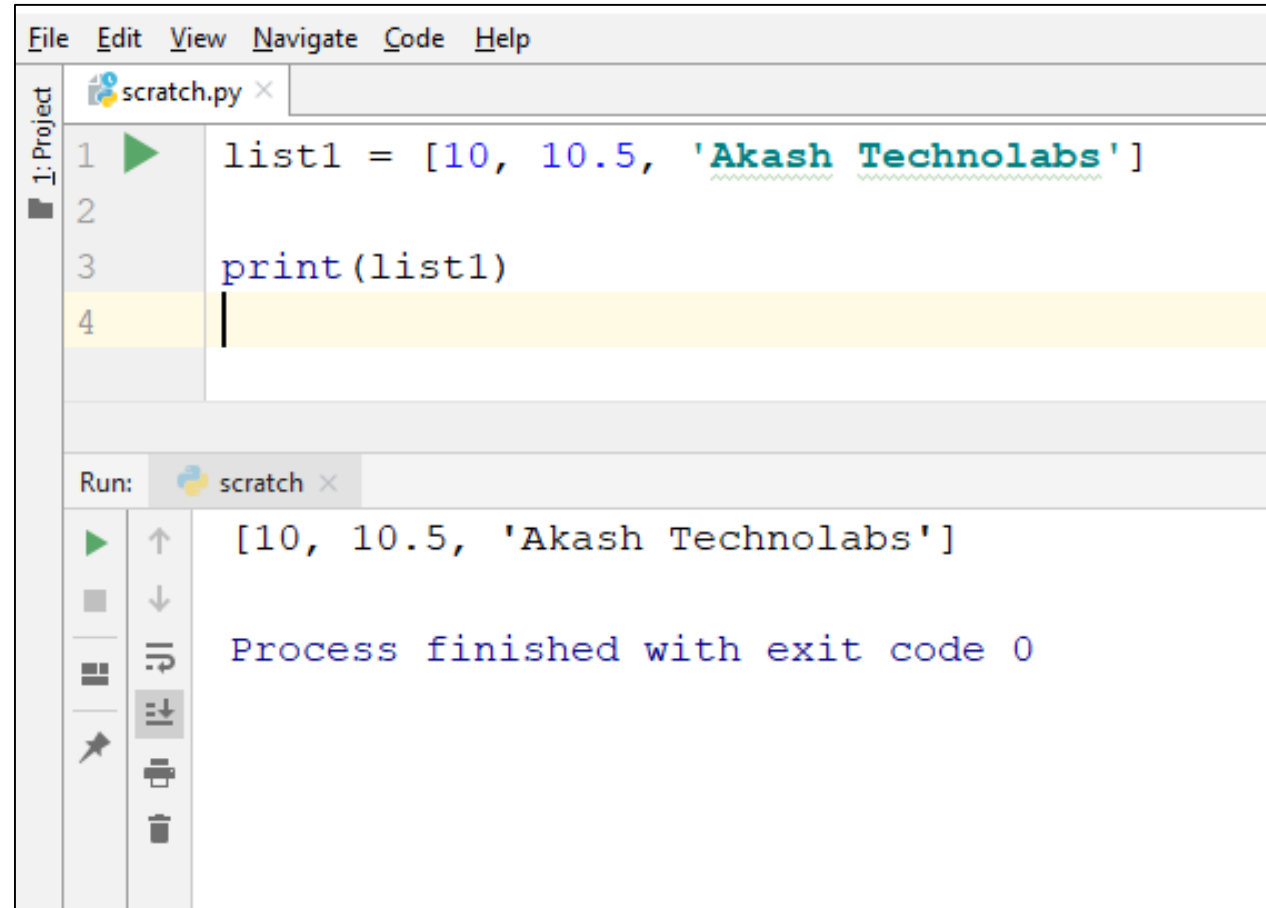
```
↑ Name is : Akash Technolabs  
↓ A  
↶ ash  
↷ ash Technolabs  
↶ Akash Technolabs Akash Technolabs  
↷ Akash Technolabs Hello  
  
Process finished with exit code 0
```

List Datatype

- Lists are very similar to arrays. It is one of the most used datatype in Python and is very flexible.
- They can contain any type of variable, and they can contain as many variables as you wish.
- All the items in a list do not need to be of the same type.



Example



The screenshot shows a Python IDE with a menu bar (File, Edit, View, Navigate, Code, Help) and a toolbar. The main editor window displays a file named 'scratch.py' with the following code:

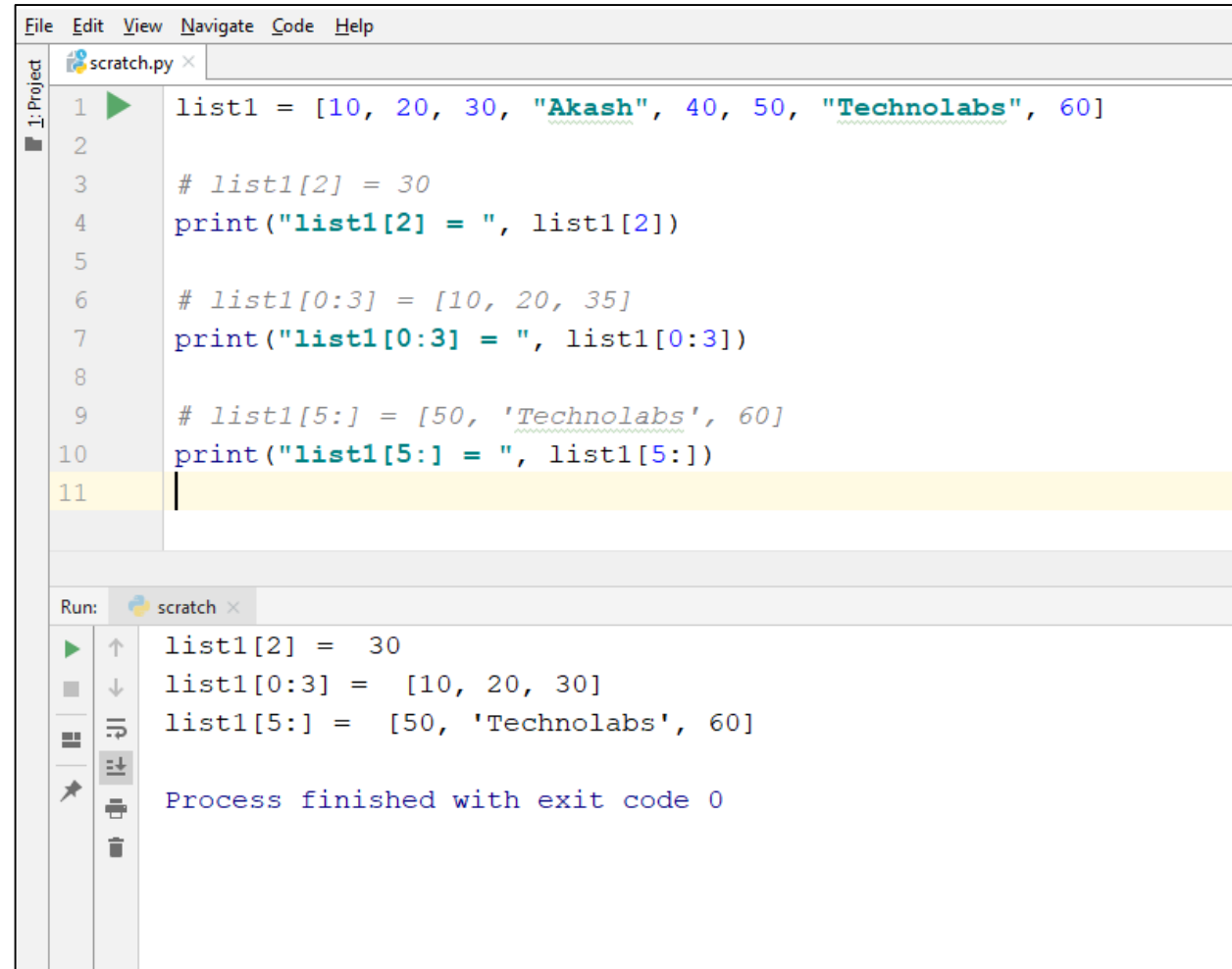
```
1 list1 = [10, 10.5, 'Akash Technolabs']  
2  
3 print(list1)  
4
```

The fourth line is highlighted in yellow. Below the editor, the 'Run' panel shows the output of the code:

```
[10, 10.5, 'Akash Technolabs']  
  
Process finished with exit code 0
```

The 'Run' panel also includes a toolbar with icons for running, stepping through, and other debugging actions.

Example 2



The screenshot shows a Python IDE with a file named 'scratch.py'. The code in the editor performs several list operations on a list named 'list1'. The output window shows the results of these operations, including the modification of individual elements and slices of the list. The process finishes with an exit code of 0.

```
File Edit View Navigate Code Help
scratch.py x
1 list1 = [10, 20, 30, "Akash", 40, 50, "Technolabs", 60]
2
3 # list1[2] = 30
4 print("list1[2] = ", list1[2])
5
6 # list1[0:3] = [10, 20, 35]
7 print("list1[0:3] = ", list1[0:3])
8
9 # list1[5:] = [50, 'Technolabs', 60]
10 print("list1[5:] = ", list1[5:])
11

Run: scratch x
list1[2] = 30
list1[0:3] = [10, 20, 30]
list1[5:] = [50, 'Technolabs', 60]

Process finished with exit code 0
```

Take value from the user

```
scratch.py x
1  # creating an empty list
2  lst = []
3
4  # number of elements as input
5  n = int(input("Enter number of elements : "))
6
7  # iterating till the range
8  for i in range(0, n):
9      ele = input("Enter Value: ")
10
11  lst.append(ele) # adding the element
12
13  print(lst)
```



Output

```
Enter number of elements : 3
Enter Value: Akash
Enter Value: TechnoLabs
Enter Value: 123
['Akash', 'Technolabs', '123']

Process finished with exit code 0
```

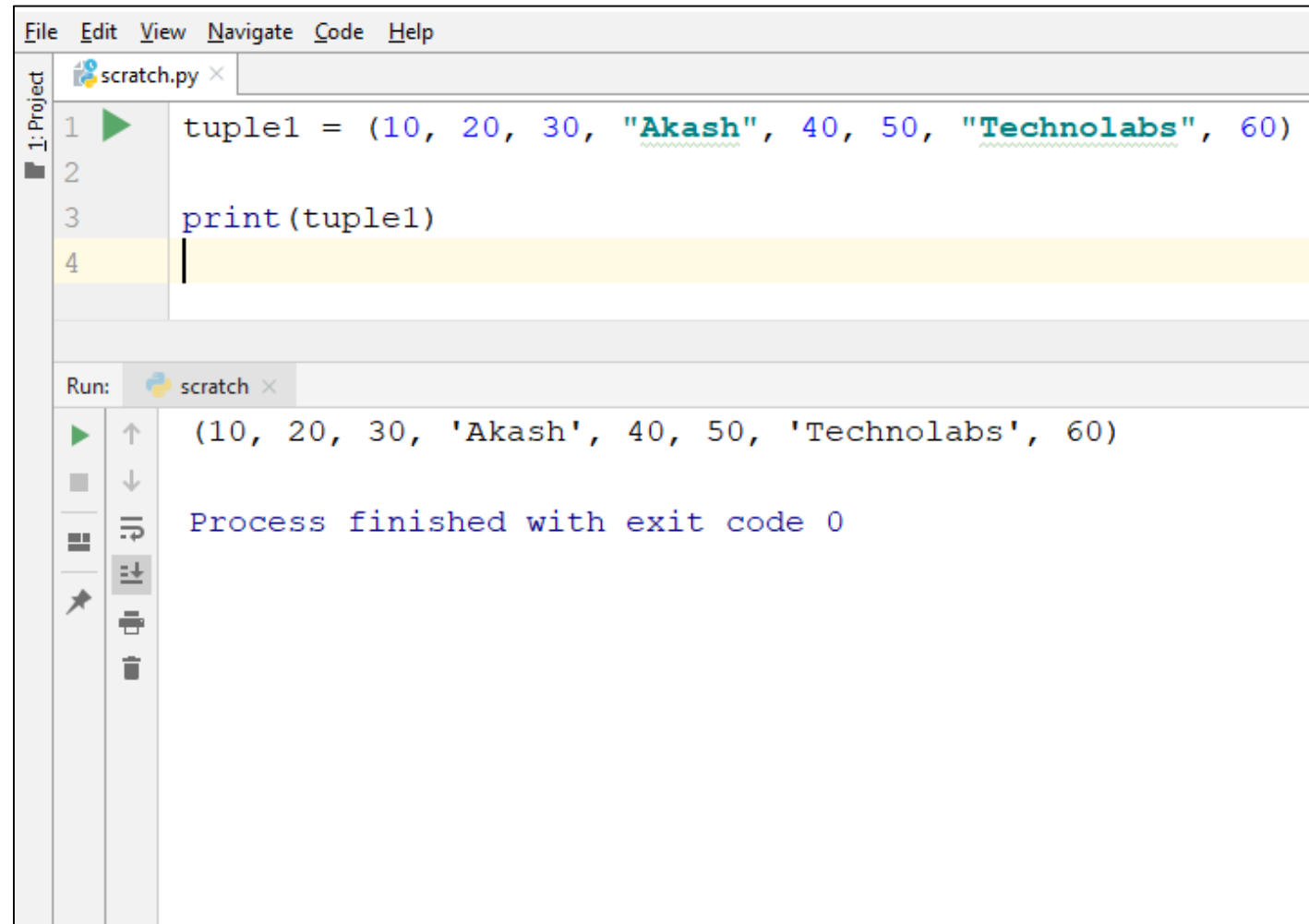


Tuple Datatype

- A tuple is another data type that is similar to the list.
- Tuples are immutable. Tuples once created cannot be modified.
- The main differences between lists and tuples are: Lists are enclosed in brackets (`[]`) and their elements and size can be changed, while tuples are enclosed in parentheses (`()`) and cannot be updated.



Example



The screenshot shows an IDE window with a menu bar (File, Edit, View, Navigate, Code, Help) and a tab for 'scratch.py'. The code editor contains the following Python code:

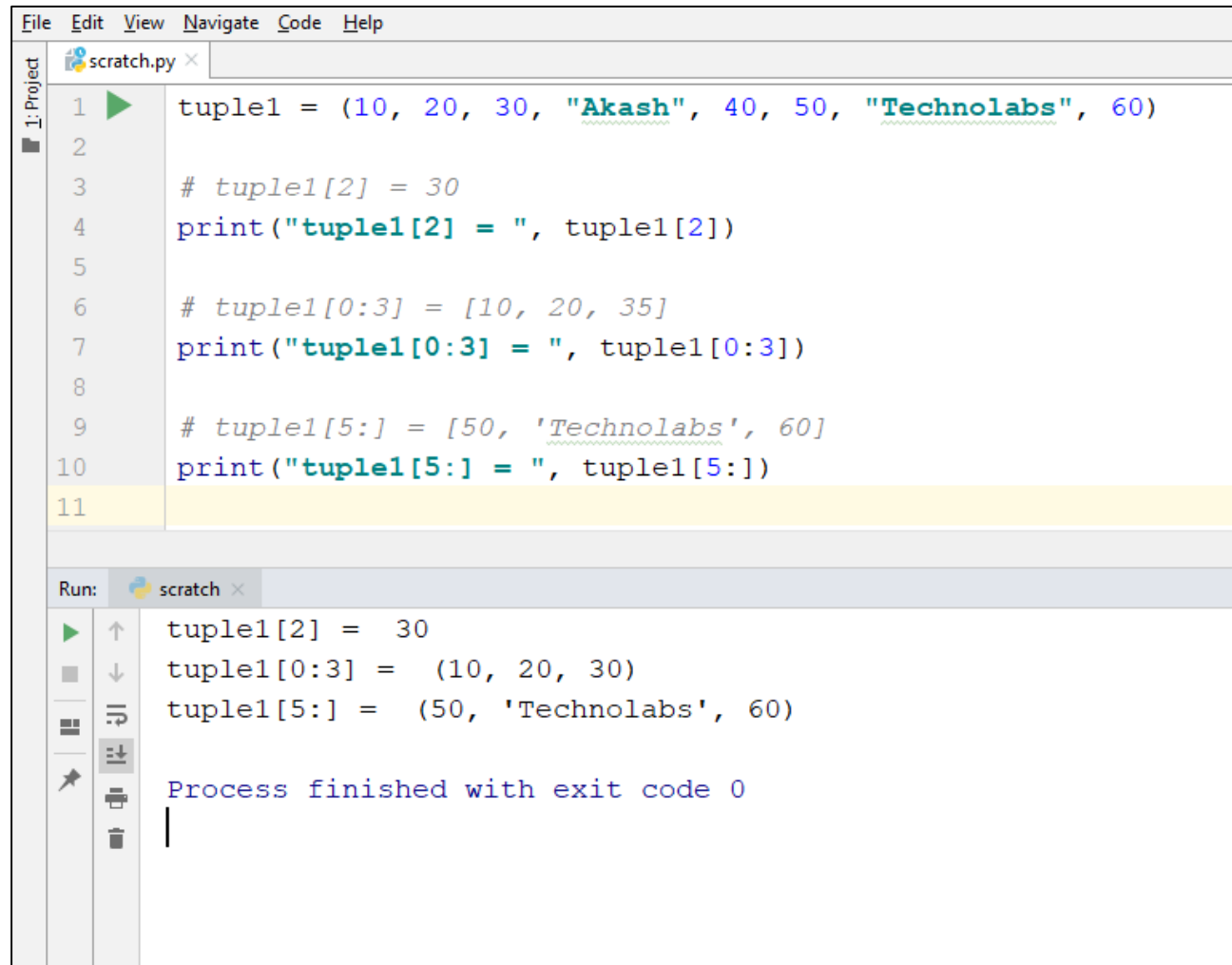
```
1 tuple1 = (10, 20, 30, "Akash", 40, 50, "Technolabs", 60)
2
3 print(tuple1)
4
```

Below the code editor is a 'Run' panel with a tab for 'scratch'. It shows the output of the script:

```
(10, 20, 30, 'Akash', 40, 50, 'Technolabs', 60)

Process finished with exit code 0
```

Example 2



The screenshot shows a Python IDE with a file named 'scratch.py'. The code defines a tuple 'tuple1' with 8 elements: 10, 20, 30, 'Akash', 40, 50, 'Technolabs', and 60. It then performs three operations: 1) Assigns the value 30 to the element at index 2. 2) Replaces the first three elements (indices 0 to 3) with the list [10, 20, 30]. 3) Replaces the elements from index 5 to the end with the list [50, 'Technolabs', 60]. The output console shows the results of these operations: 'tuple1[2] = 30', 'tuple1[0:3] = (10, 20, 30)', and 'tuple1[5:] = (50, 'Technolabs', 60)'. The process finished with exit code 0.

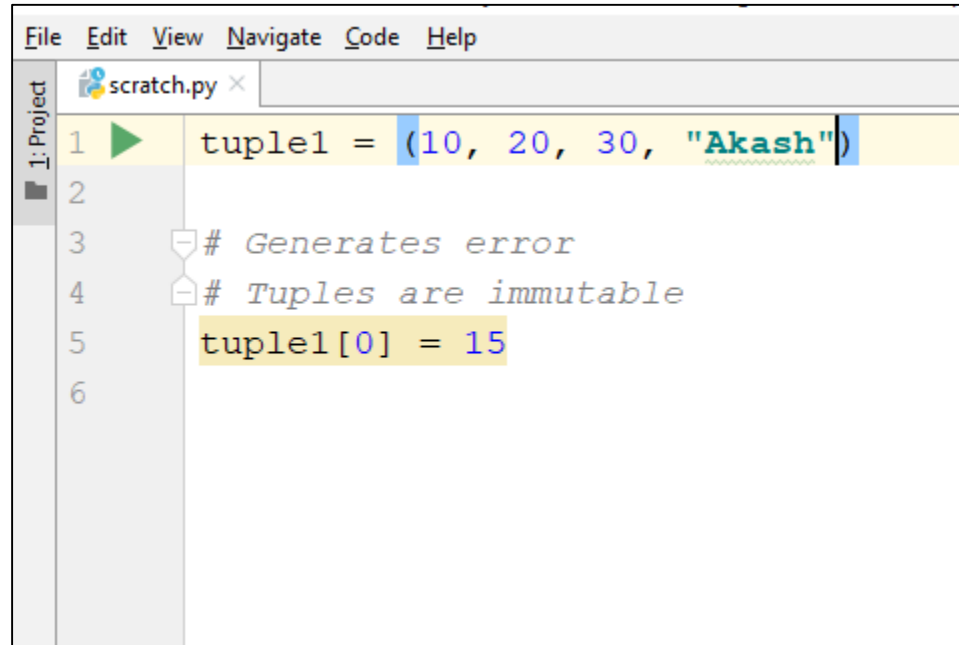
```
File Edit View Navigate Code Help
scratch.py x
1 tuple1 = (10, 20, 30, "Akash", 40, 50, "Technolabs", 60)
2
3 # tuple1[2] = 30
4 print("tuple1[2] = ", tuple1[2])
5
6 # tuple1[0:3] = [10, 20, 35]
7 print("tuple1[0:3] = ", tuple1[0:3])
8
9 # tuple1[5:] = [50, 'Technolabs', 60]
10 print("tuple1[5:] = ", tuple1[5:])
11

Run: scratch x
tuple1[2] = 30
tuple1[0:3] = (10, 20, 30)
tuple1[5:] = (50, 'Technolabs', 60)

Process finished with exit code 0
```

Error in Tuple

- We can use the slicing operator `[]` to extract items but we cannot change its value.



```
File Edit View Navigate Code Help
scratch.py x
1 tuple1 = (10, 20, 30, "Akash")
2
3 # Generates error
4 # Tuples are immutable
5 tuple1[0] = 15
6
```


Take values from the user

```
scratch.py x
1  # creating an empty list
2  lst = []
3
4  # number of elements as input
5  n = int(input("Enter number of elements : "))
6
7  # iterating till the range
8  for i in range(0, n):
9      ele = input("Enter Value: ")
10
11      lst.append(ele) # adding the element
12
13  tupl=tuple(lst)
14  print(lst)
15  print(tupl) |
```



Output

```
Enter number of elements : 3
Enter Value: 123
Enter Value: Akash
Enter Value: Technolabs
['123', 'Akash', 'Technolabs']
('123', 'Akash', 'Technolabs')

Process finished with exit code 0
|
```

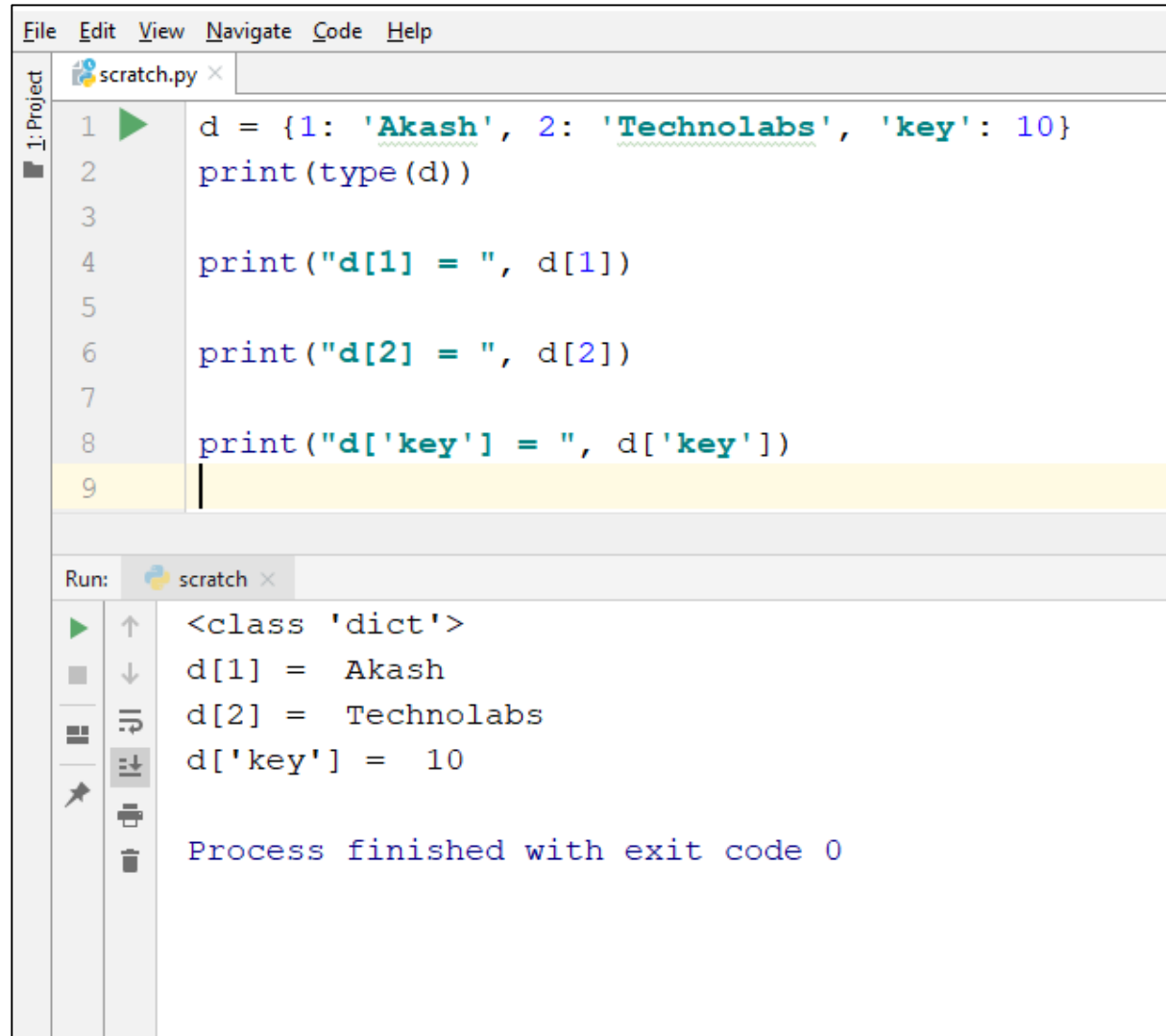


Dictionary Datatype

- Dictionaries are kind of hash table type.
- Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).



Example



The screenshot shows a Python IDE with a file named 'scratch.py'. The code defines a dictionary 'd' with three items: '1' mapped to 'Akash', '2' mapped to 'Technolabs', and 'key' mapped to 10. It then prints the type of 'd', and the values for 'd[1]', 'd[2]', and 'd['key']'. The 'Run' panel at the bottom shows the output of these operations, confirming the dictionary's structure and the values of its elements. The process finished with exit code 0.

```
File Edit View Navigate Code Help
scratch.py x
1 d = {1: 'Akash', 2: 'Technolabs', 'key': 10}
2 print(type(d))
3
4 print("d[1] = ", d[1])
5
6 print("d[2] = ", d[2])
7
8 print("d['key'] = ", d['key'])
9
Run: scratch x
> <class 'dict'>
> d[1] = Akash
> d[2] = Technolabs
> d['key'] = 10
Process finished with exit code 0
```

Take values from the user

```
scratch.py x
1  mydict = {}
2  for totnum in range(0,int(input('Input the total number : '))):
3      a, b = input('Enter the key value pair :').split()
4      if a in mydict:
5          mydict[a].append(b)
6      else:
7          mydict[a] = [b]
8
9  print(mydict)
```



Output

```
Input the total number : 3
Enter the key value pair :1 Akash
Enter the key value pair :2 TechnoLabs
Enter the key value pair :a 10
{'1': ['Akash'], '2': ['Technolabs'], 'a': ['10']}

Process finished with exit code 0
|
```

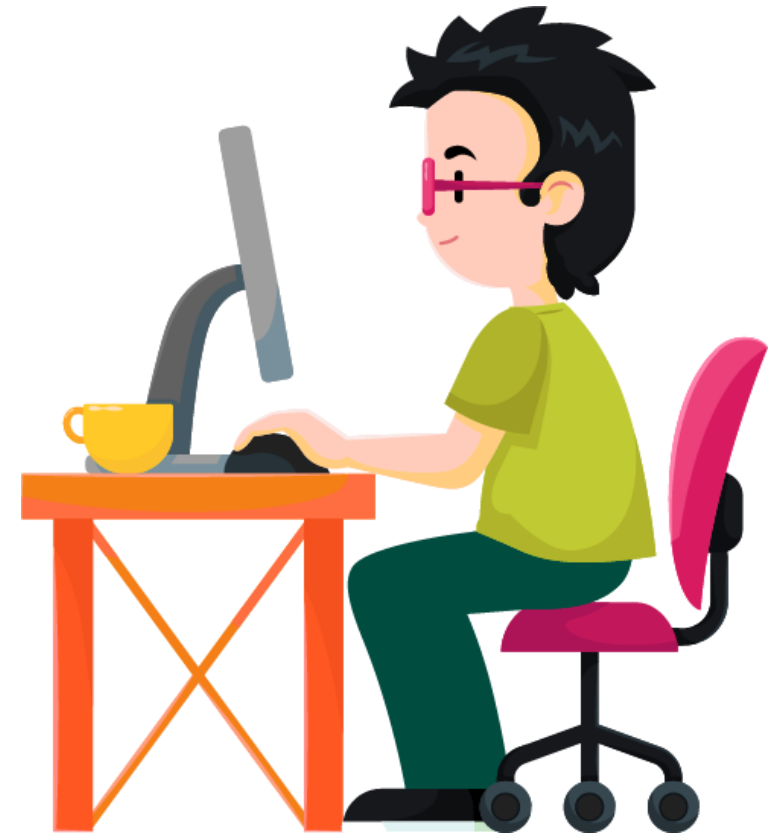


Get Exclusive Video Tutorials



www.apptutorials.com

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





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



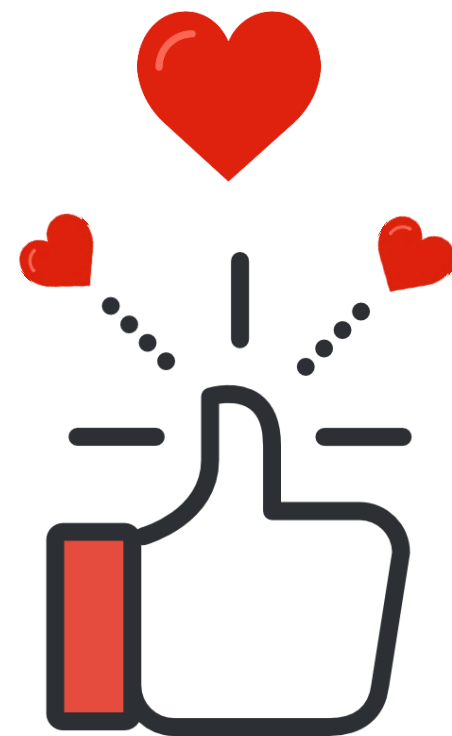
Just Dial

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.akashtechlabs.com

www.akashpadhiyar.com

www.aptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar
#aptutorials