CodroidHub Summer Training

Index

SUBMITTED BY: ASMIT

ROLL NO: (2322811)

BRANCH: AI&ML

SUBMITTED TO: Mr. DEVASHISH SIR

(FOUNDER, CodroidHub Private Limited)

Loops in Python

Loops in Python are used to execute a block of code repeatedly. There are two main types of loops in Python for loops and while loops.

While loop in python

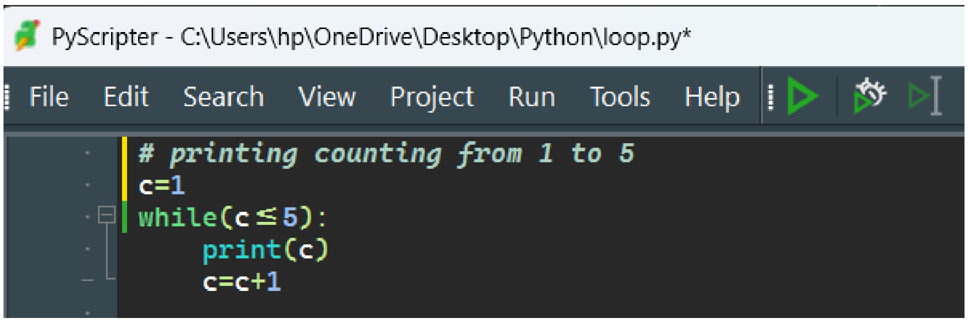
while loop is used to execute a block of statements repeatedly until a given condition is satisfied. When the condition becomes false, the line immediately after the loop in the program is executed.

While loop syntax

While expression:

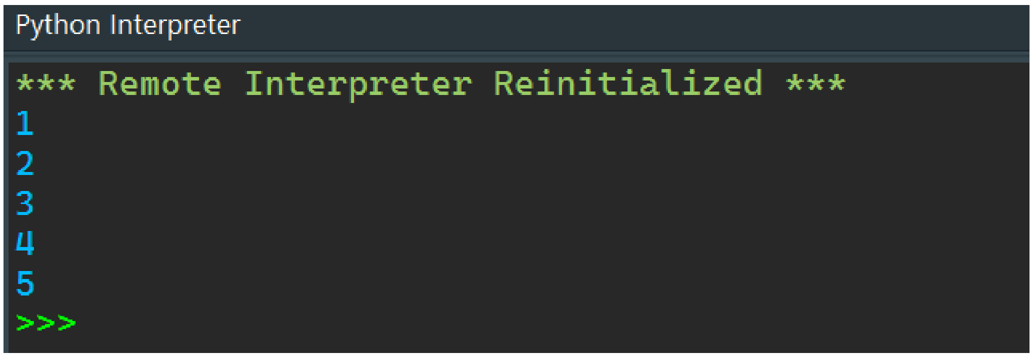
        Statements(x)

Example of while loop:


PyScripter - C:\Users\hp\OneDrive\Desktop\Python\loop.py*
File Edit Search View Project Run Tools Help

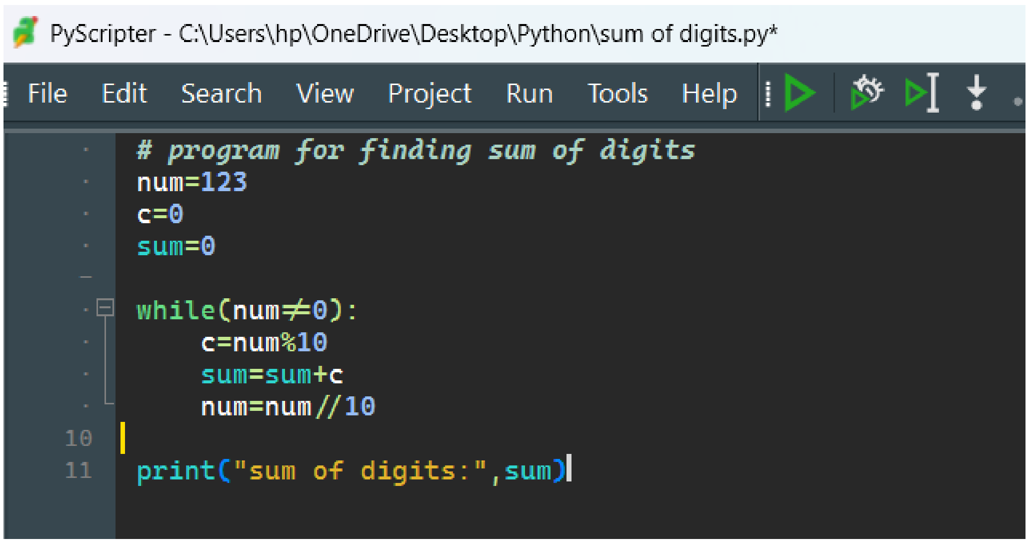
```
# printing counting from 1 to 5
c=1
while(c≤5):
    print(c)
    c=c+1
```

Output:



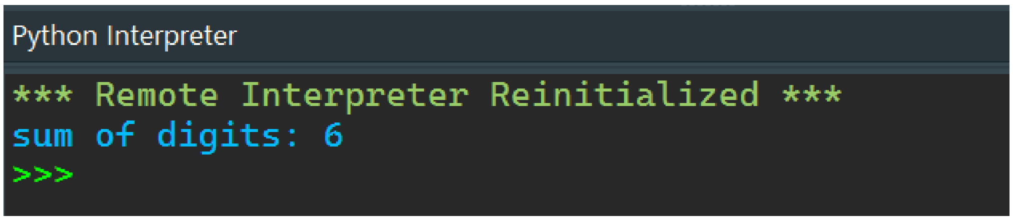
```
*** Remote Interpreter Reinitialized ***
1
2
3
4
5
>>>
```

Example :


PyScripter - C:\Users\hp\OneDrive\Desktop\Python\sum of digits.py*
File
Edit
Search
View
Project
Run
Tools
Help

```
# program for finding sum of digits
num=123
c=0
sum=0

while(num!=0):
    c=num%10
    sum=sum+c
    num=num//10

print("sum of digits:",sum)
```

Output:



```
*** Remote Interpreter Reinitialized ***
sum of digits: 6
>>>
```

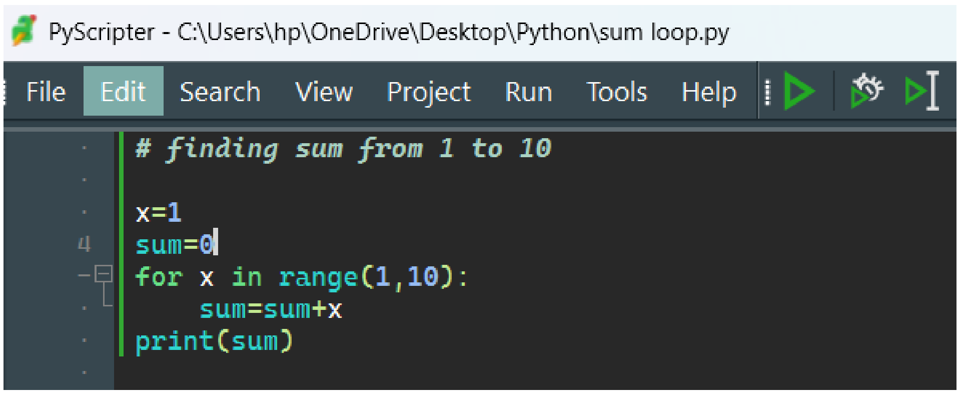
For loop in python

For loops are used for sequential traversal within the specified range. There is “for in” loop which is similar to foreach loop in other languages.

For loop syntax

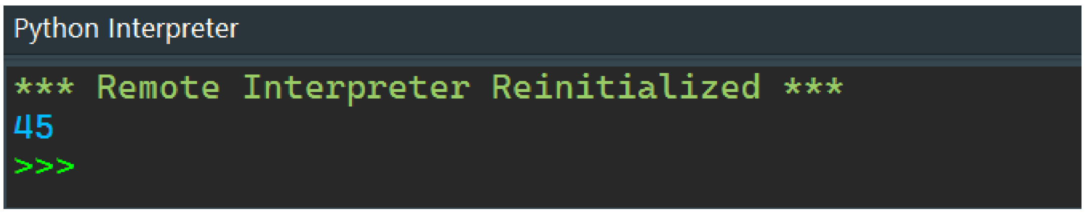
```
for iterator_var in sequence:
   statements(x)
```

Example:


PyScripter - C:\Users\hp\OneDrive\Desktop\Python\sum loop.py
File Edit Search View Project Run Tools Help

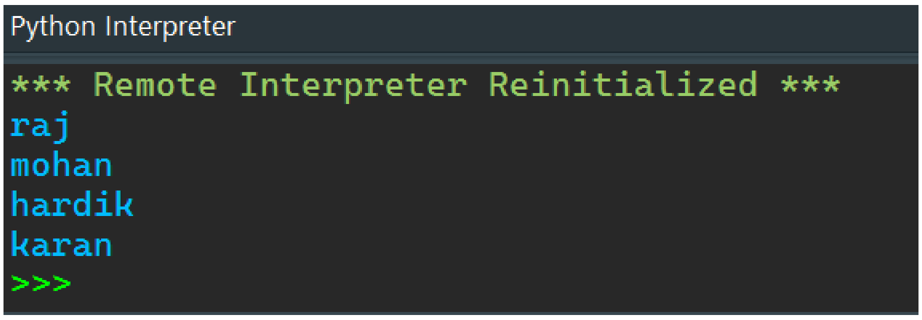
```
# finding sum from 1 to 10

x=1
sum=0
for x in range(1,10):
    sum=sum+x
print(sum)
```

Output:



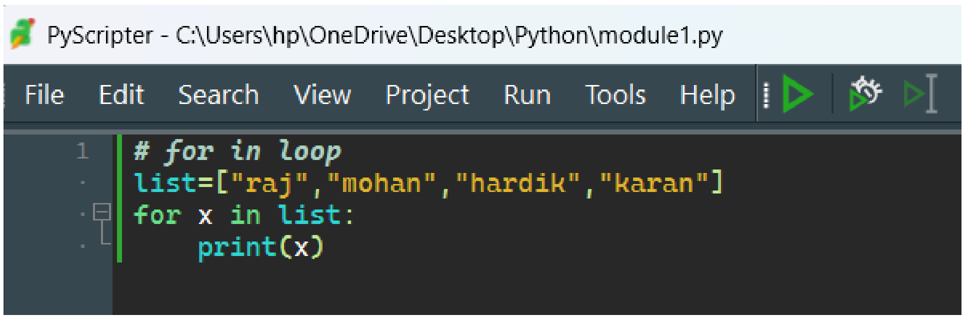
```
*** Remote Interpreter Reinitialized ***
45
>>>
```

Example of for in loop:



```
*** Remote Interpreter Reinitialized ***
raj
mohan
hardik
karan
>>>
```

Output:


PyScripter - C:\Users\hp\OneDrive\Desktop\Python\module1.py
File Edit Search View Project Run Tools Help

```
# for in loop
list=["raj","mohan","hardik","karan"]
for x in list:
    print(x)
```

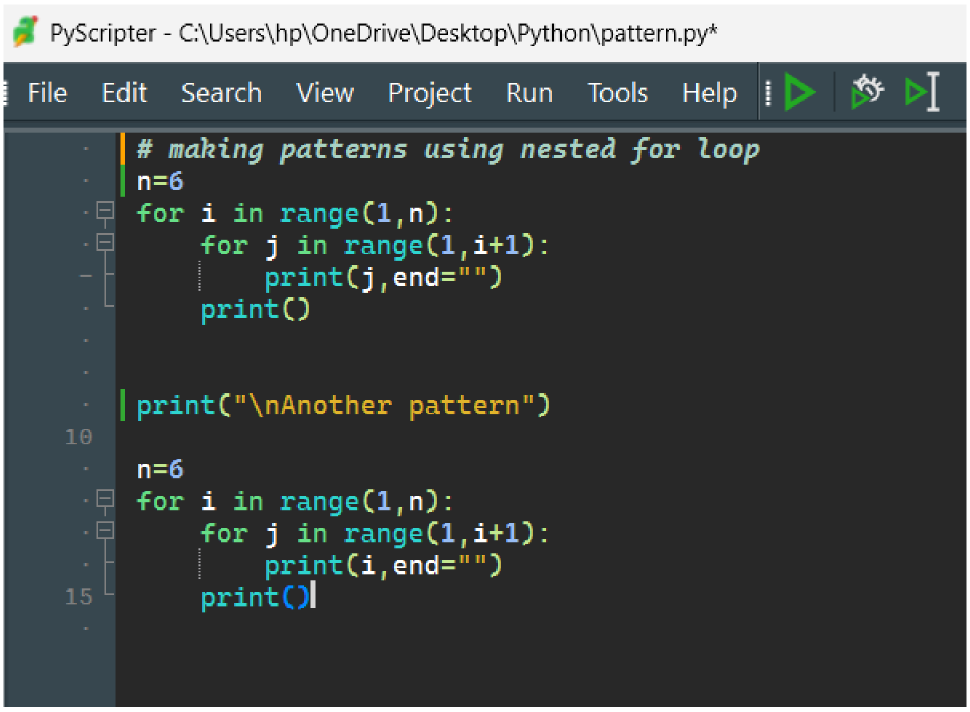
Nested for loop

Python programming language allows to use one loop inside another loop which is called nested loop.

Nested for loop syntax

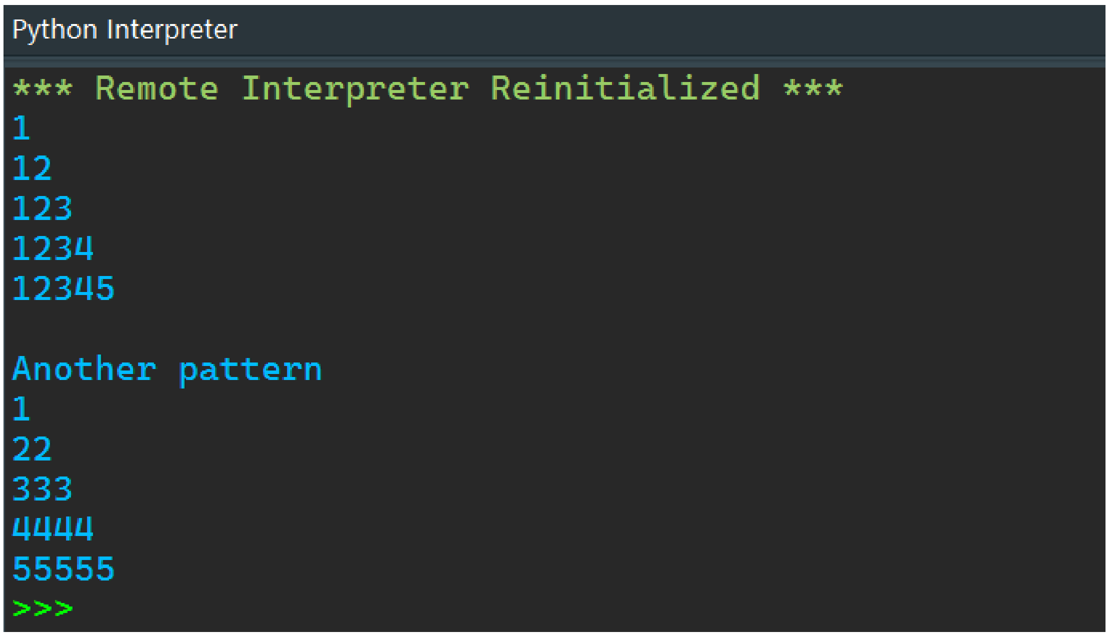
```
for iterator_var in sequence:
   for iterator_var in sequence:
       statements(x)
   statements(x)
```

Example:


PyScripter - C:\Users\hp\OneDrive\Desktop\Python\pattern.py*
File Edit Search View Project Run Tools Help

```python
# making patterns using nested for loop
n=6
for i in range(1,n):
    for j in range(1,i+1):
        print(j,end="")
    print()


print("\nAnother pattern")

n=6
for i in range(1,n):
    for j in range(1,i+1):
        print(i,end="")
    print()
```

Output:



```
*** Remote Interpreter Reinitialized ***
1
12
123
1234
12345

Another pattern
1
22
333
4444
55555
>>>
```

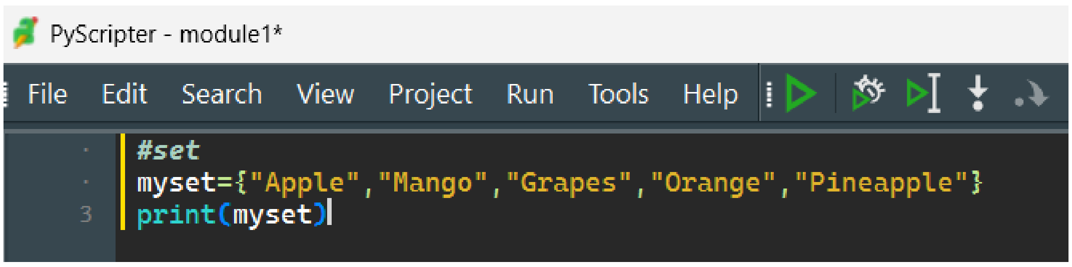
Python set

A set in Python is an unordered collection of unique items. Sets are mutable, meaning you can add or remove items after a set is created, but they do not allow duplicate elements. Sets are useful for membership tests, eliminating duplicate entries, and performing mathematical set operations like union, intersection, difference, and symmetric difference.

Creating Sets

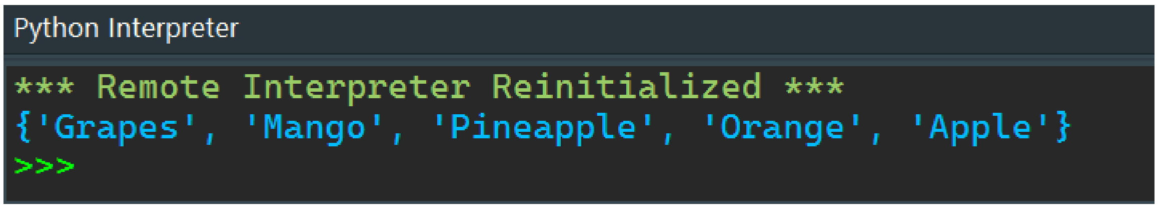
You can create sets using curly braces `{}` or the `set()` function.

Example:



```
#set
myset={"Apple","Mango","Grapes","Orange","Pineapple"}
print(myset)
```

Output:



```
*** Remote Interpreter Reinitialized ***
{'Grapes', 'Mango', 'Pineapple', 'Orange', 'Apple'}
>>>
```

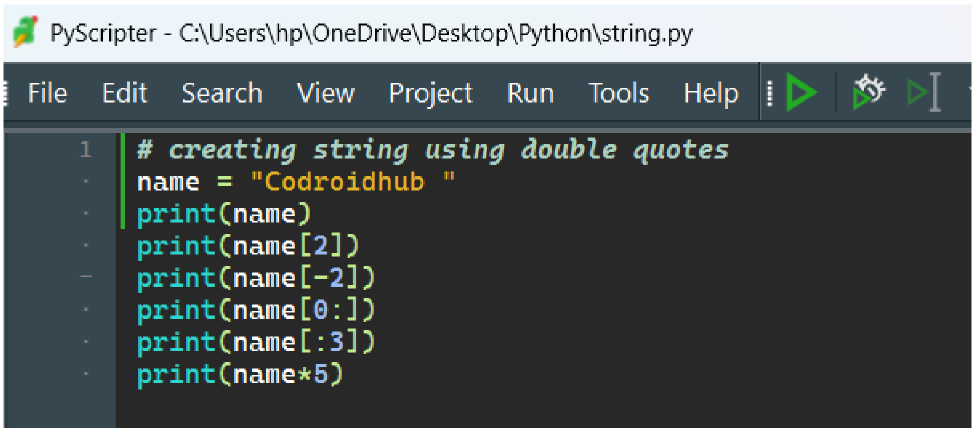
Python String

A String is a data structure in Python Programming that represents a sequence of characters. It is an immutable data type, meaning that once you have created a string, you cannot change it.

Creating String

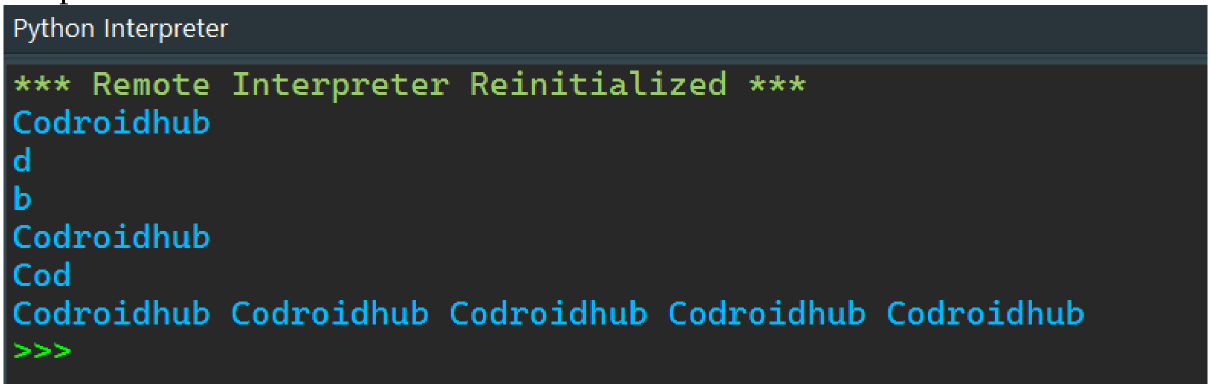
Strings can be created by enclosing it in single quotes ('), double quotes ("), or triple quotes (''' or """).

Example:


PyScripter - C:\Users\hp\OneDrive\Desktop\Python\string.py
File Edit Search View Project Run Tools Help

```python
# creating string using double quotes
name = "Codroidhub "
print(name)
print(name[2])
print(name[-2])
print(name[0:])
print(name[:3])
print(name*5)
```

Output:



```
*** Remote Interpreter Reinitialized ***
Codroidhub
d
b
Codroidhub
Cod
Codroidhub Codroidhub Codroidhub Codroidhub Codroidhub
>>>
```

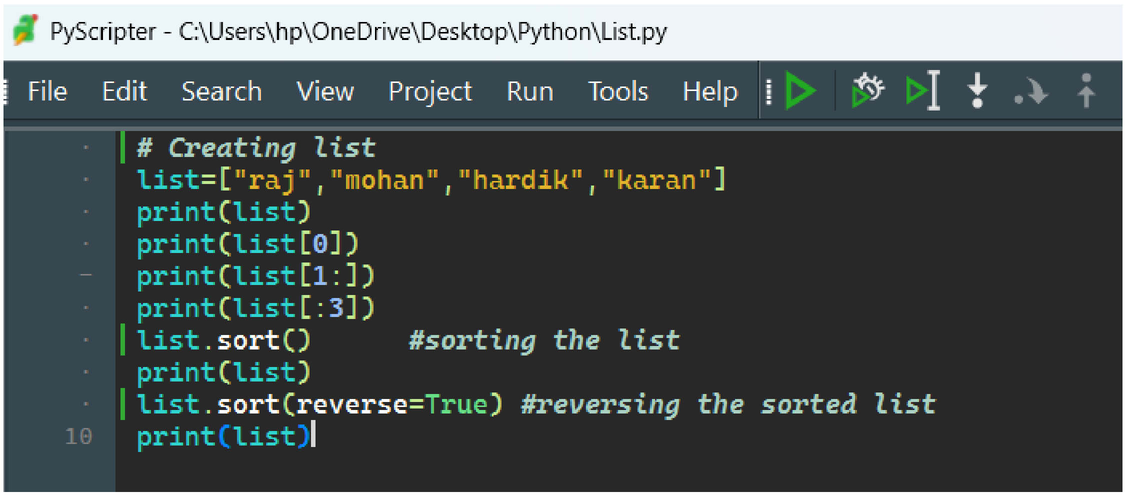
Python List

A list in Python is a built-in data structure that allows you to store an ordered collection of items. Lists are mutable, meaning you can modify their contents (add, remove, or change items) after they are created. Lists can contain items of different types, including integers, floats, strings, and even other lists.

Creating List

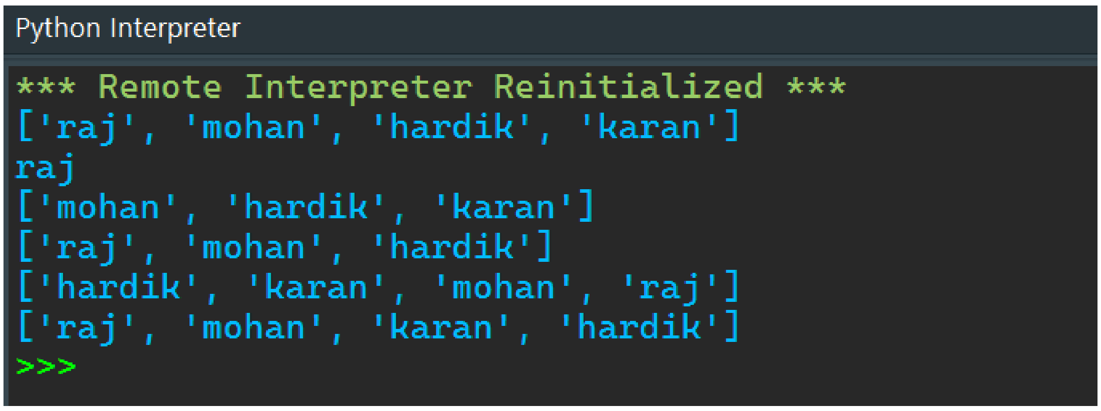
You can create lists using square brackets `[]` or the `list()` function.

Example:


PySripter - C:\Users\hp\OneDrive\Desktop\Python\List.py
File Edit Search View Project Run Tools Help

```python
# Creating list
list=["raj","mohan","hardik","karan"]
print(list)
print(list[0])
print(list[1:])
print(list[:3])
list.sort()        #sorting the list
print(list)
list.sort(reverse=True) #reversing the sorted list
print(list)
```

Output:



```
*** Remote Interpreter Reinitialized ***
['raj', 'mohan', 'hardik', 'karan']
raj
['mohan', 'hardik', 'karan']
['raj', 'mohan', 'hardik']
['hardik', 'karan', 'mohan', 'raj']
['raj', 'mohan', 'karan', 'hardik']
>>>
```

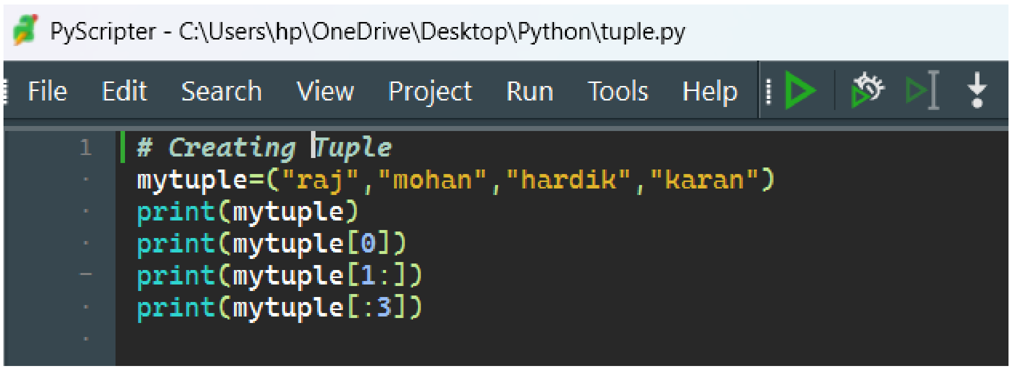
Python Tuple

A tuple in Python is an immutable, ordered collection of items. Once created, the items in a tuple cannot be changed, added, or removed. Tuples can contain items of different types, including integers, floats, strings, and even other tuples.

Creating Tuple

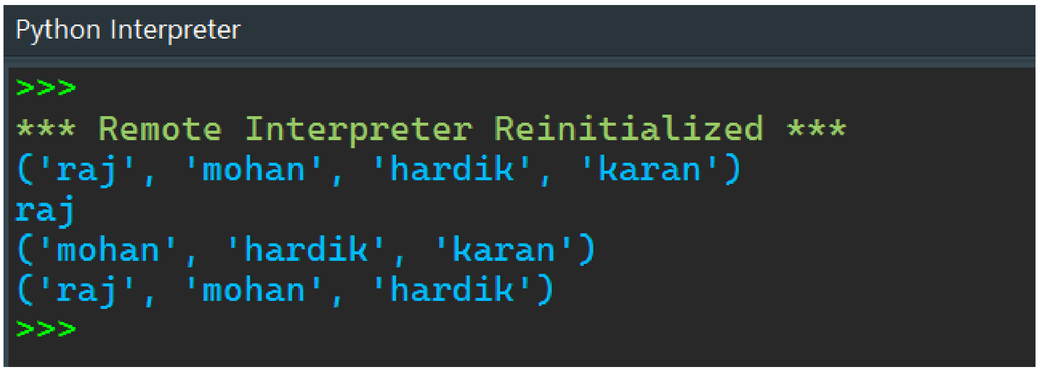
You can create tuples using parentheses () or the tuple() function.

Example:



```python
# Creating Tuple
mytuple=("raj","mohan","hardik","karan")
print(mytuple)
print(mytuple[0])
print(mytuple[1:])
print(mytuple[:3])
```

Output:



```
>>>
*** Remote Interpreter Reinitialized ***
('raj', 'mohan', 'hardik', 'karan')
raj
('mohan', 'hardik', 'karan')
('raj', 'mohan', 'hardik')
>>>
```

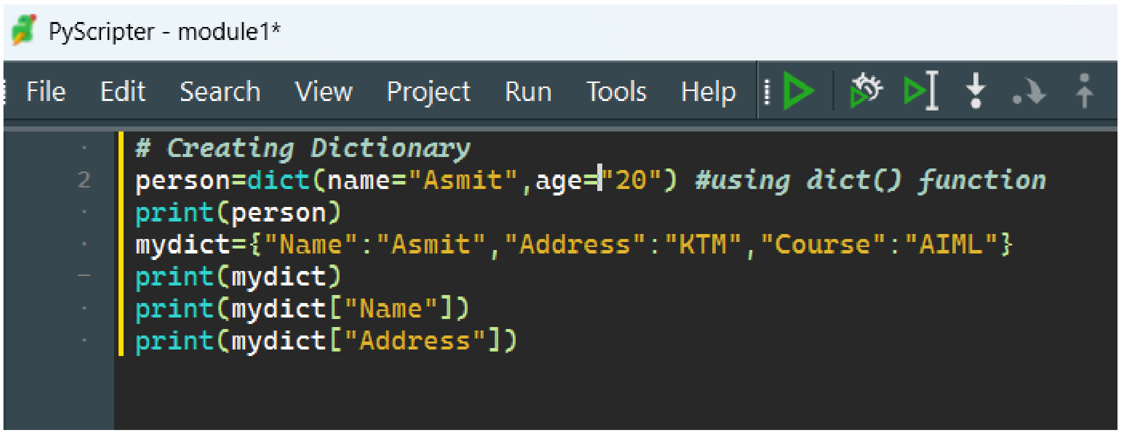
Python Dictionary

A dictionary in Python is a mutable, unordered collection of key-value pairs. Each key-value pair maps the key to its associated value. Dictionaries are indexed by keys, which can be any immutable type, such as strings, numbers, or tuples. Values can be of any type, including other dictionaries.

Creating Dictionary

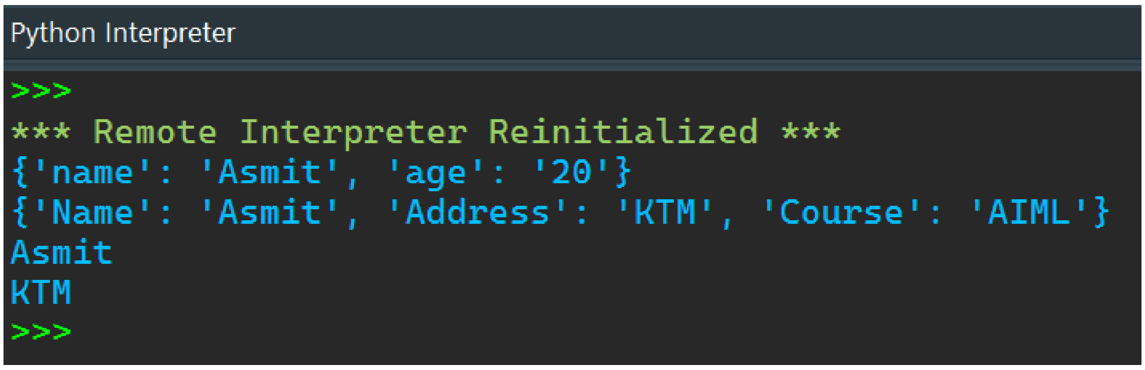
You can create dictionaries using curly braces {} with key-value pairs separated by colons, or by using the dict() function.

Example:



```python
# Creating Dictionary
person=dict(name="Asmit",age="20") #using dict() function
print(person)
mydict={"Name":"Asmit","Address":"KTM","Course":"AIML"}
print(mydict)
print(mydict["Name"])
print(mydict["Address"])
```

Output:



```
>>>
*** Remote Interpreter Reinitialized ***
{'name': 'Asmit', 'age': '20'}
{'Name': 'Asmit', 'Address': 'KTM', 'Course': 'AIML'}
Asmit
KTM
>>>
```