CodroidHub Summer Training

**Index**

* Loops in python
* While loop
* For loop
* Nested loop
* Set
* String
* List
* Tuple
* Dictionary

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

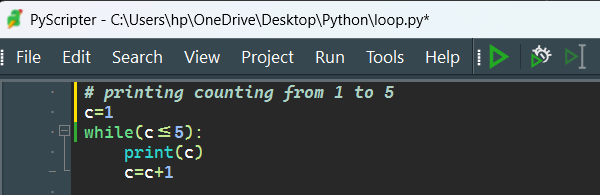
while loop is used to execute a block of statements repeatedly until a given condition is satisfied. When the condition becomes false, the line immediately after the loop in the program is executed.

While loop syntax

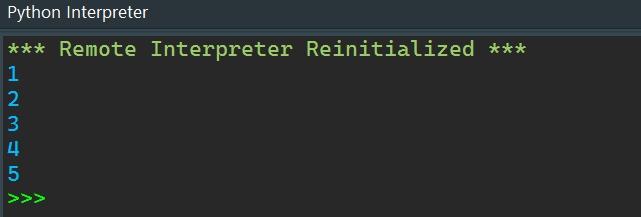
While expression:

Statements(x)

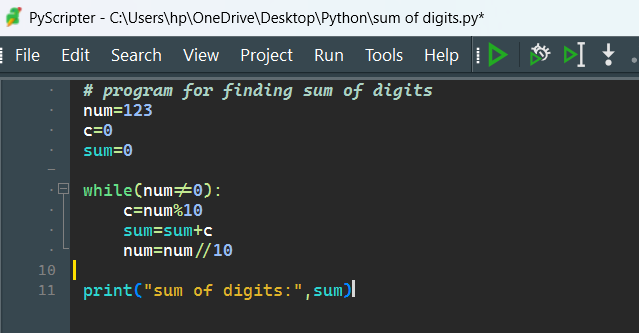
Example of while loop:



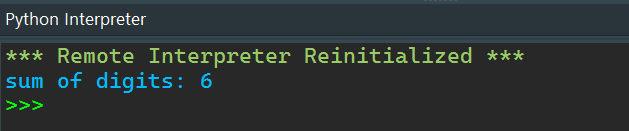
Output:



Example :



Output:



For loop in python

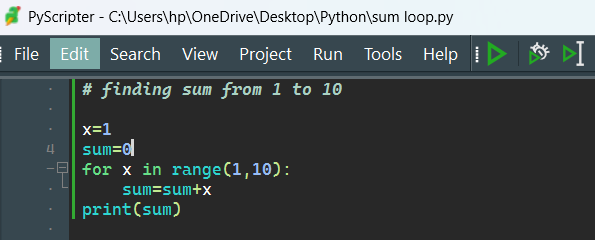
For loops are used for sequential traversal within the specified range. There is “for in” loop which is similar to foreach loop in other languages.

For loop syntax

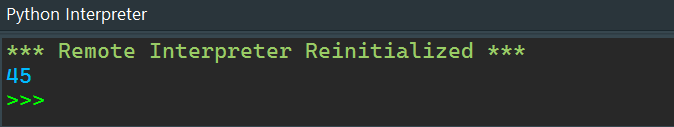
for iterator\_var in sequence:

statements(x)

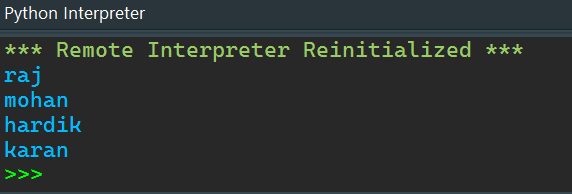
Example:



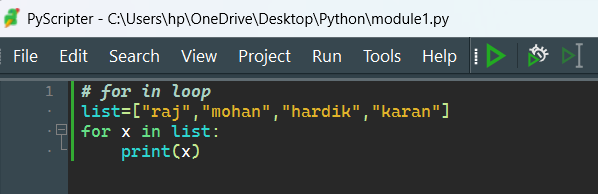
Output:



Example of for in loop:



Output:



Nested for loop

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

Nested for loop syntax

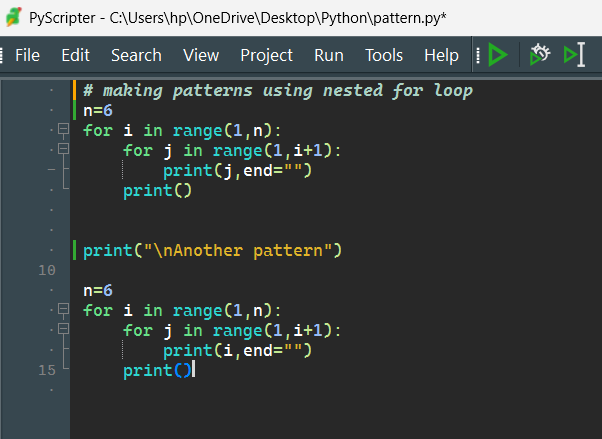
for iterator\_var in sequence:

for iterator\_var in sequence:

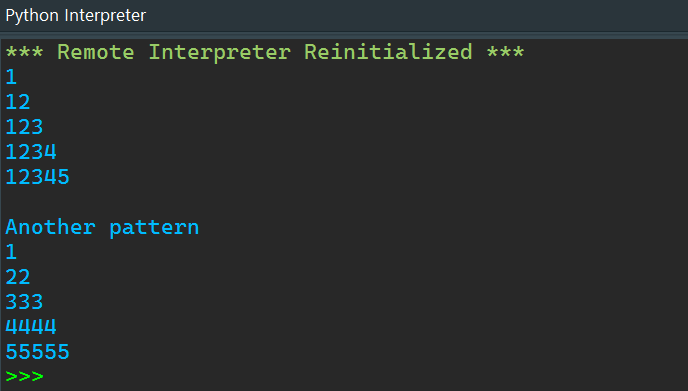
statements(x)

statements(x)

Example:



Output:



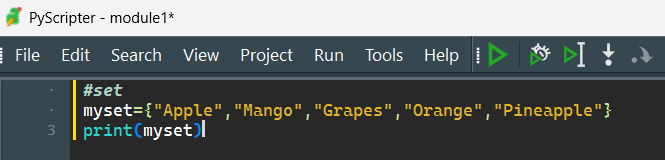
Python set

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.

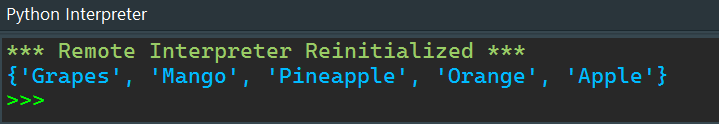
Creating Sets

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

Example:



Output:



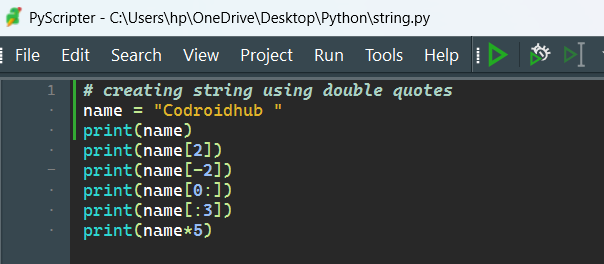
Python String

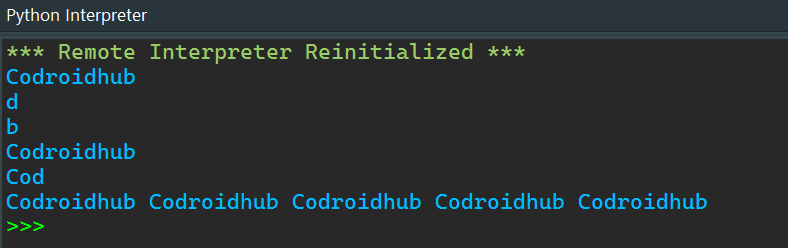
A String is a data structure in Python Programming that represents a sequence of characters. It is an immutable data type, meaning that once you have created a string, you cannot change it.

Creating String

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

Example:



Output: 

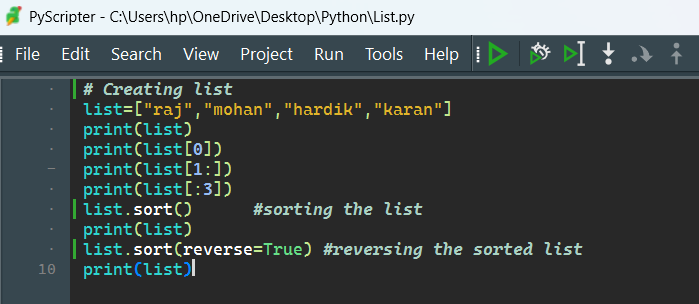
Python List

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.

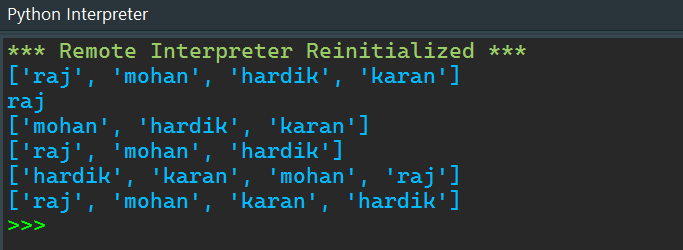
Creating List

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

Example:



Output:



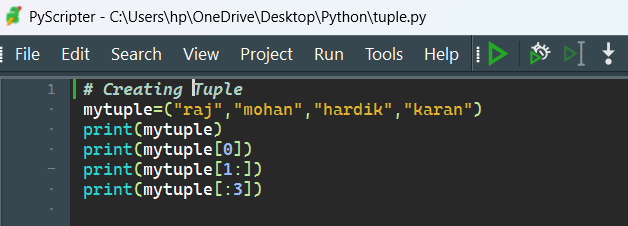
Python Tuple

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.

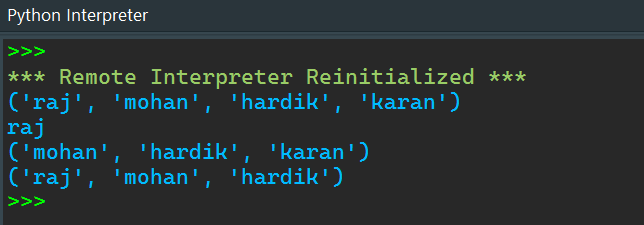
Creating Tuple

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

Example:



Output:



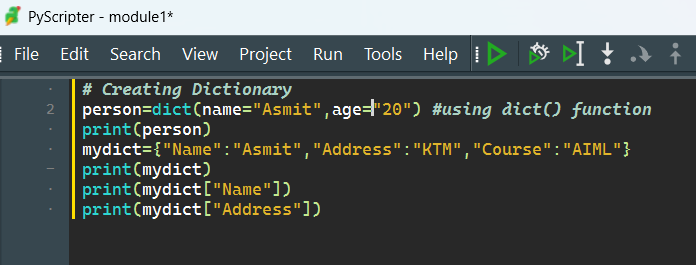
Python Dictionary

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:



Output:

