COLLECTION RELATED DATA TYPES:

If you want to represent group of values in a single entity we can use collection related data types.

**There are four collection data types in the Python programming language:**

* List is a collection which is ordered and changeable. Allows duplicate members.
* Tuple is a collection which is ordered and unchangeable. Allows duplicate members.
* Set is a collection which is unordered, unchangeable\*, and unindexed. No duplicate members.
* Frozen set -
* Dictionary is a collection which is ordered\*\* and changeable. No duplicate members.
* Range
* Bytes
* Bytes Array

Ex : Collection relate data types.

List:

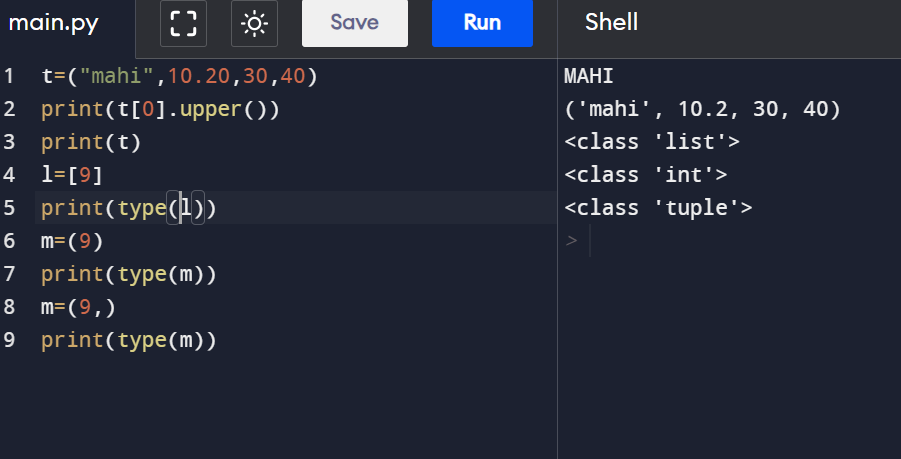
* Ordered(order preserved)
* Duplicates are allowed
* Changeable
* Heterogeneous objects are allowed.
* Indexing concept and slicing concept applicable
* Growable in nature.
* List is mutable.
* Lists are created using square brackets:

Ex:

mylist = ["apple", "banana", "cherry"]

Tuple:

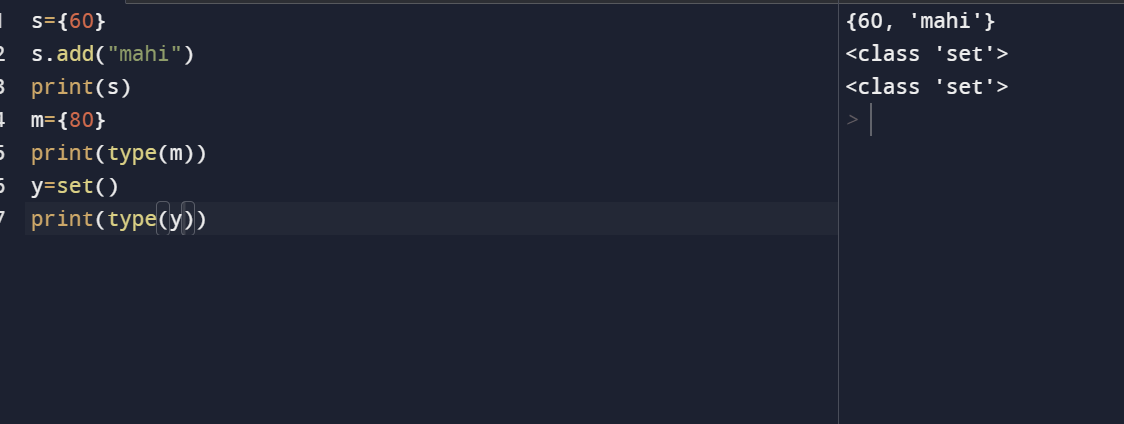
* Exactly same as list
* Only difference ,it is immutable
* Once we create object we can’t change anything
* Read only version of list is tuple
* Can’t perform append()and remove()function, if we try we will get AttributeError.
* Single value tuple should ends with comma(ex.9,)not(9)like this
* To create a tuple we will use () operators.
* Ex: var = ("Geeks", "for", "Geeks")
* print(var)



|  |  |
| --- | --- |
| **List** | **Tuple** |
| Mutable | Immutable |
| [ ] | ( ) |
| It takes more memory | It takes less memory |
| Performance is less | Faster Access(Performance is high) |
| If the content keeps on changing use List | If content was stable use Tuple |
| Single value List considered as List only | Single value tuple should ends with comma |
| Ex. l=[9]  Type-List | Ex.t=(9)  Type – int  T=(9,)  Type - Tuple |

SET DATA TYPE:

* Set is mutable
* Order is not important
* Duplicates are not allowed
* No Indexing and slicing operator applicable
* Heterogeneous objects are allowed
* Append() method not allowed, instead of append we have use **add()** method
* Growable
* S={ } empty set type was **dict**
* By default **{ } is a dictionary not a set**
* **Dict** is mostly used in python compared to **set**
* So **dict** has higher priority than **set**
* If you want to create empty set use **set( )** method



|  |  |
| --- | --- |
| Append() | Add() |
| Append method always add objects at last position | Add method can add objects at any place |
| Append method used for List Data type | Add method used for Set data type |

FROZENSET DATA TYPE:

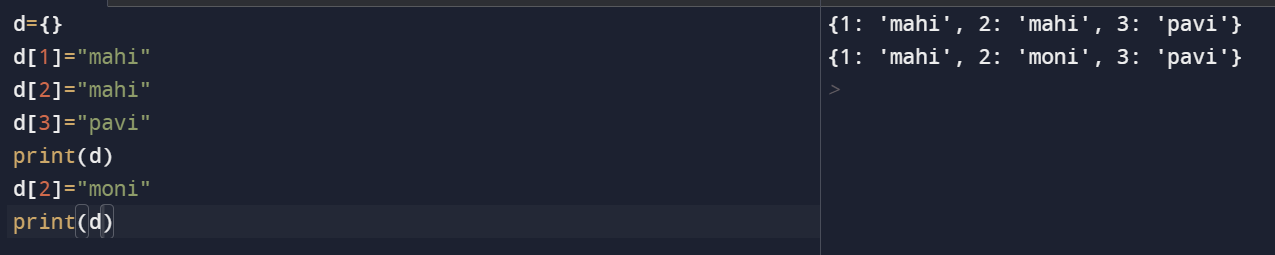
* Exactly same as Set but it’s immutable
* To create a frozenset wehave to call frozenset() function
* Freeze the list, and make it unchangeable:
* mylist = ['apple', 'banana', 'cherry']  
  x = frozenset(mylist)

|  |  |
| --- | --- |
| TUPLE | FROZENSET |
| Immutable | Immutable |
| Order Preserved | Order not applicable |
| Duplicates are allowed | Duplicates are not allowed |

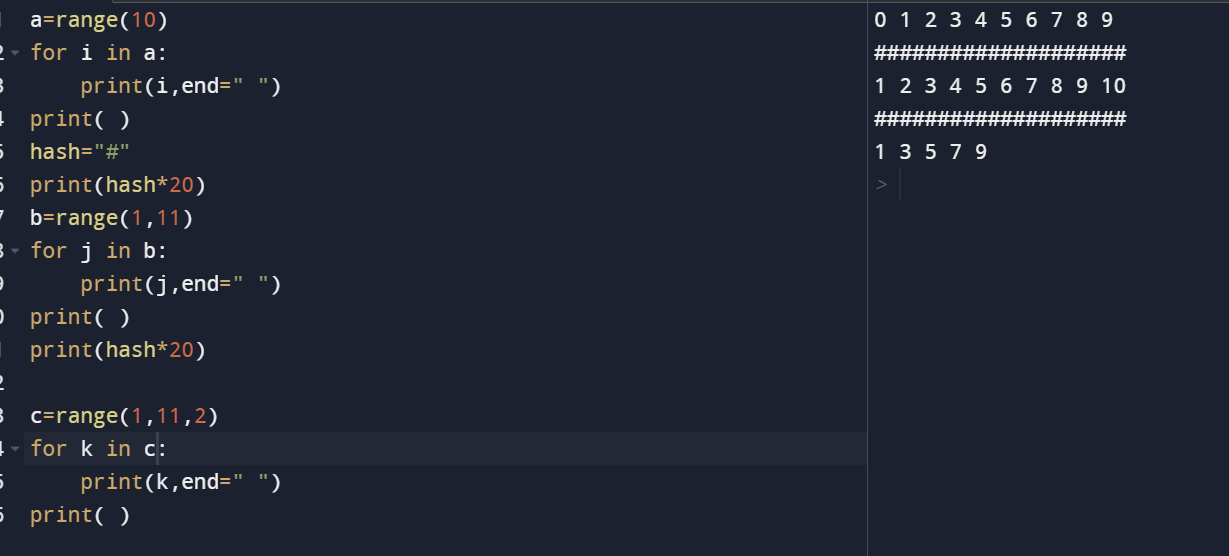
|  |  |
| --- | --- |
| Index, slice applicable | Index, slice not applicable |

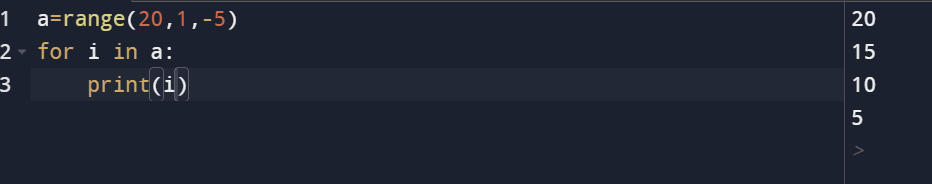
DICTIONARY:

* Key-value pair
* Syntax: d={key:value,….}
* thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
  }
* Mutable
* A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.
* Duplicate keys are not allowed
* Duplicate values are allowed
* If we are trying to add duplicate keys old value will be replaced for the particular key.
* Heterogeneous objects are allowed.
* dict' object has no attribute 'remove'
* No Indexing and slicing



RANGE:

* Range is python in- built data type
* Sequence data type
* Immutable
* Ordered
* Indexing and slicing possible
* **Method 1 Syntax:** range()
* Range starts from 0 to n-1
* **Another method 2:** range(begin,end-1)
* Ex: range(1,10)--🡪1……….9
* Using for loop to print range values
* **Method 3:** range(begin,end-1,increment/decrement



BYTES ARRAY:

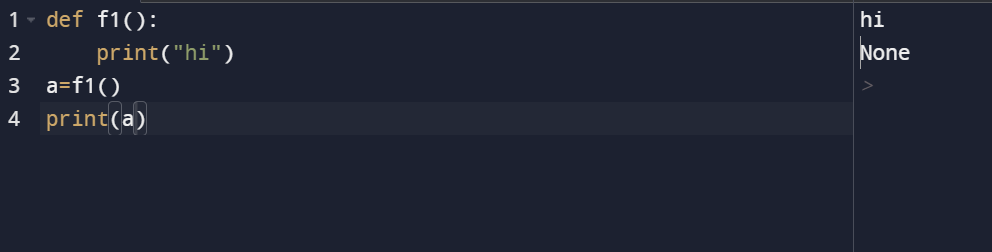
* If I want to create a group of byte values we can use byte() method
* If you want to handle binary data(Video,audio) there we must use byte() data type
* Value from 0 to 256(means 0 to 255)only
* If we are trying other values we will get value error
* **Immutable**
* Indexing/slicing applicable

Byte array:

* **Mutable**
* Using bytearray() function to create byte array
* Indexing/slicing applicable
* Value from 0 to 256(means 0 to 255)only

NONE DATA TYPE:

* None means nothing
* No value associated
* If function won’t return anything internally it returns None Data type.
* Only one None object was there in python



ESCAPE CHARACTERS:

* To insert characters that are illegal in a string, use an escape character.
* An escape character is a backslash \ followed by the character you want to insert.
* An example of an illegal character is a double quote inside a string that is surrounded by double quotes:
* You will get an error if you use double quotes inside a string that is surrounded by double quotes:
* txt = "We are the so-called "Vikings" from the north."
* To fix this problem, use the escape character \":

### **Example**

* The escape character allows you to use double quotes when you normally would not be allowed:
* txt = "We are the so-called \"Vikings\" from the north."

|  |  |
| --- | --- |
| **Code** | **Result** |
| \' | Single Quote |
| \\ | Backslash |
| \n | New Line |
| \r | Carriage Return |
| \t | Tab |
| \b | Backspace |
| \f | Form Feed |
| \ooo | Octal value |
| \xhh | Hex value |

CONSTANTS IN PYTHON:

* constants are not there in python
* by convention we can use uppercase eg:MAX-LENGTH=10 but we can change at any time