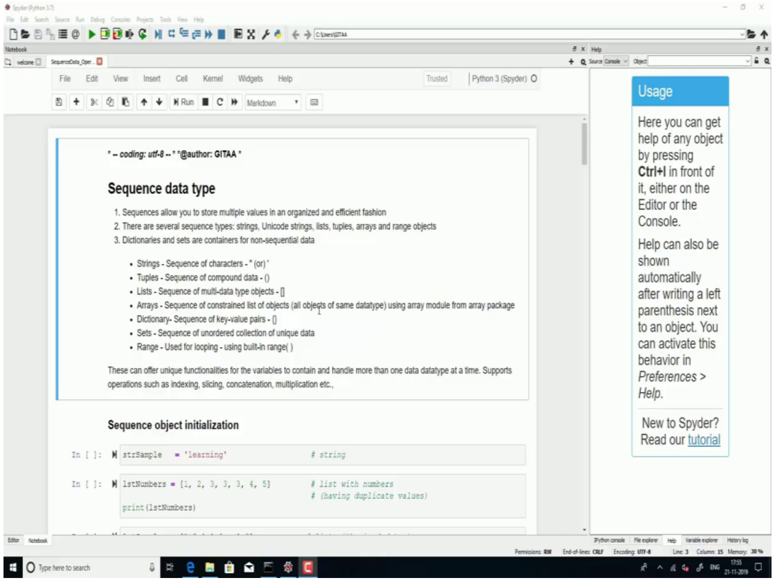
Python for Data Science
Prof. Ragunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 8
Sequence Data Part 1

Hello all welcome to the lecture on the sequence data type. So, in this lecture we are going to deal with sequence data type. So, before getting into that let us get started with what sequence data types are.

(Refer Slide Time: 00:26)



* -- *coding: utf-8* -- * *@author: GITAA *

Sequence data type

1. Sequences allow you to store multiple values in an organized and efficient fashion
2. There are several sequence types: strings, Unicode strings, lists, tuples, arrays and range objects
3. Dictionaries and sets are containers for non-sequential data

- Strings - Sequence of characters - " (or) '
- Tuples - Sequence of compound data - ()
- Lists - Sequence of multi-data type objects - []
- Arrays - Sequence of constrained list of objects (all objects of same datatype) using array module from array package
- Dictionary- Sequence of key-value pairs - {}
- Sets - Sequence of unordered collection of unique data
- Range - Used for looping - using built-in range( )

These can offer unique functionalities for the variables to contain and handle more than one data datatype at a time. Supports operations such as indexing, slicing, concatenation, multiplication etc.,

Sequence object initialization

```
In [ ]:  strSample   = 'learning'                # string

In [ ]:  lstNumbers = [1, 2, 3, 3, 3, 4, 5]       # list with numbers
                                                  # (having duplicate values)
         print(lstNumbers)
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

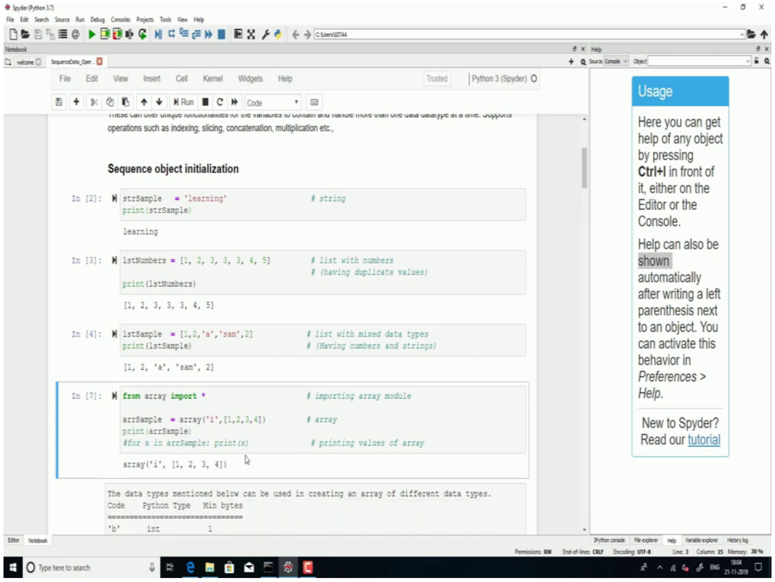
New to Spyder? Read our tutorial

Basically sequence data type allows you to create or store multiple values in an organised and efficient fashion. So there are several sequence for example strings, Unicode strings, lists, tuples arrays and range objects. The two of the other data types are called as dictionaries and sets or containers for non sequential data. In this lecture we are going to look at the examples for all of the sequential and non sequential data type.

The first let us look at what string data type is. A string is a sequence of one or more characters, for example, it can contain letters, numbers and symbols and that can be either a constant or variable. And strings are basically a mutable sequence in Python. So, to create a string we can close a sequence of characters inside single, double or triple quotes. Let us see an example on creating string. Create string includes A sequence of characters inside single, double or triple quotes.

I am creating the string called strsample and it has a sequence of characters which describes the word learning and storing that on two variable called strsample. So that becomes a string. So we can print out strings by simply calling the print function. So when we print the string strsample will be getting an output call learning. So the strsample is now a string. So let us move to the next sequence data that is called list.

(Refer Slide Time: 01:58)



operations such as indexing, slicing, concatenation, multiplication etc.,

Sequence object initialization

```
In [2]:  strSample   = 'learning'                # string
         print(strSample)

         learning

In [3]:  lstNumbers = [1, 2, 3, 3, 3, 4, 5]      # list with numbers
                                                 # (having duplicate values)
         print(lstNumbers)

         [1, 2, 3, 3, 3, 4, 5]

In [4]:  lstSample  = [1,2,'a','sam',2]         # list with mixed data types
         print(lstSample)                       # (Having numbers and strings)

         [1, 2, 'a', 'sam', 2]

In [7]:  from array import *                     # importing array module

         arrSample  = array('i',[1,2,3,4])      # array
         print(arrSample)
         #for x in arrSample: print(x)           # printing values of array

         array('i', [1, 2, 3, 4])
```

```
The data types mentioned below can be used in creating an array of different data types.
Code    Python Type   Min bytes
================================
'b'     int           1
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

So list in Python can be created by just placing the sequence inside the square brackets. So, here I am creating a list call lisnumber and it is going to contain only the numbers. And if you look at this example in this list I have an element which is repeating basically I am having duplicate values. Because the list may contain duplicate values with their distinct positions and hence multiple this thing or duplicate values can be passed as a sequence at the time of the list creation itself.

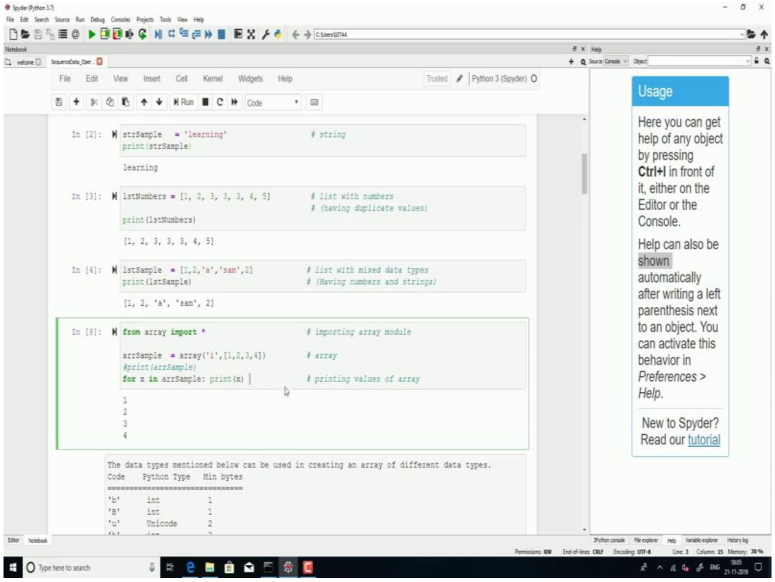
So here lstnumber is the lissample which has only numbers and it also has duplicate values to it. So let us print the list and see what is the output. So you be able to see the output here it has values 1, 2, 3, 3, 3, 4 and 5 but a single is may contain data types like integers strings as well as object. So list can contain elements of multiple data types and list also mutable and hence they can be altered even after their creation.

So now I am creating another list call list sample lstsample which has both numbers and strings to it. So, for example 1, 2, a, sam, and as well as 2 here, a lstsample has multiple data types to it. Let us print that and see. So the one advantage of using list is basically you can have elements of multiple data type. So, next we will look at another sequence data call Array and an array is nothing but it is a collection of items stored at contiguous memory locations.

And we can use the array to store multiple items of the same data types together and arrays in Python can be created by importing the array module. So let us import the array module so from array I am just important it as asterix so that I can use just a function from array without even calling them. And the syntax or the way we can use the array function is we need to specify the data type and the value list as argument to the function array.

So let us see how to create array using the array function. As a first argument as specified the data type and a second argument given the values as a list and storing that output as an object called array sample. So, now let us print the values of array and see. So here I am using for loop to print the values of the array sample.

(Refer Slide Time: 04:41)



```
In [2]:  strSample   = 'learning'                 # string
         print(strSample)

         learning

In [3]:  lstNumbers = [1, 2, 3, 3, 3, 4, 5]        # list with numbers
                                                 # (having duplicate values)
         print(lstNumbers)

         [1, 2, 3, 3, 3, 4, 5]

In [4]:  lstSample   = [1,2,'a','sam',2]         # list with mixed data types
         print(lstSample)                       # (Having numbers and strings)

         [1, 2, 'a', 'sam', 2]

In [8]:  from array import *                    # importing array module

         arrSample   = array('i',[1,2,3,4])     # array
         #print(arrSample)
         for x in arrSample: print(x)           # printing values of array

         1
         2
         3
         4
```

```
The data types mentioned below can be used in creating an array of different data types.
Code    Python Type   Min bytes
==============================
'b'      int           1
'B'      int           1
'u'      Unicode       2
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

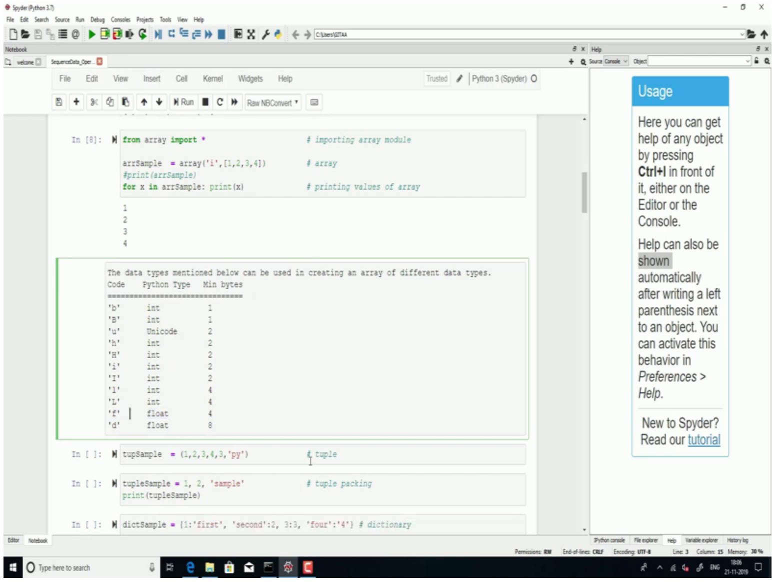
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

So, if you just use print of your array name, if you use the print function it will just going to give you the same input that you give in but if you want to print the values of your array then you can use the for loop to get the values of your array. So, basically in the first itration is going to print the first value and for loop goes to the next titration. So it is going to print all the values in your array sample. Now we got the values. So the values are 1, 2, 3 and 4.

So this is how we create an array. As you can see here that array you created now is just one dimensional array, we will see how to create or how to deal with multidimensional array in the upcoming session.

(Refer Slide Time: 05:32)



```
In [8]:  from array import *                        # importing array module

          arrSample  = array('i',[1,2,3,4])          # array
          #print(arrSample)
          for x in arrSample: print(x)               # printing values of array

          1
          2
          3
          4
```

```
The data types mentioned below can be used in creating an array of different data types.
Code    Python Type   Min bytes
================================
'b'     int           1
'B'     int           1
'u'     Unicode       2
'h'     int           2
'H'     int           2
'i'     int           2
'I'     int           2
'l'     int           4
'L'     int           4
'f'     float         4
'd'     float         8
```

```
In [ ]:  tupSample  = (1,2,3,4,3,'py')              # tuple

In [ ]:  tupleSample = 1, 2, 'sample'               # tuple packing
         print(tupleSample)

In [ ]:  dictSample = {1:'first', 'second':2, 3:3, 'four':'4'} # dictionary
```

Usage

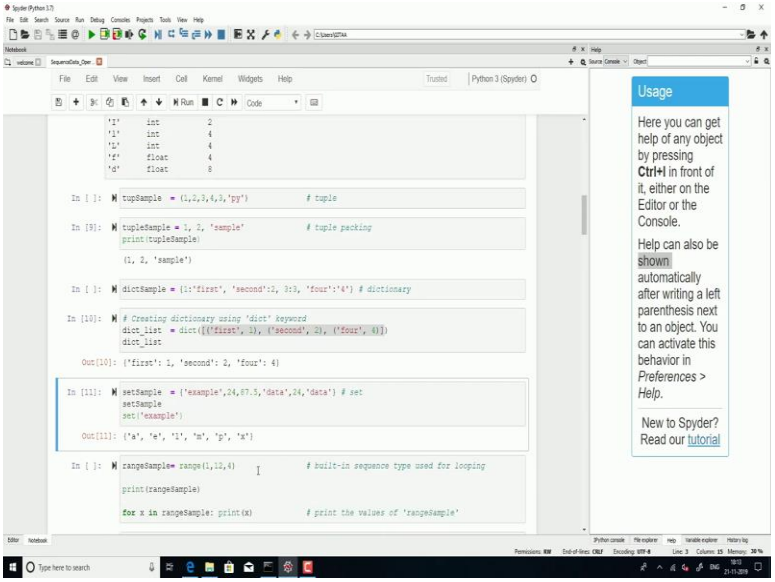
Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

So here I specify the data type as integer to represent integer I have given i similarly if you want to create an array with different data types, then you can use the notation to denote different types of data. So here the data types mentioned below can be used in creating an array of different data types. So, I have given the quote here to represent the Python type and also given the minimum number of bytes. For example, if you want to create an array with the data type float, then you can use the code f.

(Refer Slide Time: 06:13)



```
 'I'      int        2
 'l'      int        4
 'L'      int        4
 'f'      float      4
 'd'      float      8

In [ ]:   tupSample  = (1,2,3,4,3,'py')              # tuple

In [9]:   tupleSample = 1, 2, 'sample'               # tuple packing
          print(tupleSample)

          (1, 2, 'sample')

In [ ]:   dictSample = {1:'first', 'second':2, 3:3, 'four':'4'} # dictionary

In [10]:  # Creating dictionary using 'dict' keyword
          dict_list  = dict([('first', 1), ('second', 2), ('four', 4)])
          dict_list

Out[10]:  {'first': 1, 'second': 2, 'four': 4}

In [11]:  setSample  = {'example',24,87.5,'data',24,'data'} # set
          setSample
          set('example')

Out[11]:  {'a', 'e', 'l', 'm', 'p', 'x'}

In [ ]:   rangeSample= range(1,12,4)                 # built-in sequence type used for looping

          print(rangeSample)

          for x in rangeSample: print(x)             # print the values of 'rangeSample'
```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

So, the next will move on to tuple that is also one of the sequence data type. Tuple in Python is similar to a list but the difference between the two is that we cannot change the elements of a tuple once it is assigned whereas in a list elements can be changed. In Python tuples created by placing all the elements inside parents is separated by commas. And I am just separating all the values or elements by commas.

And tuple can have any number of items and they can be of different data types and here I have integer as well as string to the tuple that I am creating now and tuple can also be created without using the parenthesis as well and that is known as tuple packing. So here I have example to show you. So here I have not use the parenthesis to create the tuple but I am just given the values separated by commas.

So if you print the values of the value of your tuple, this is how it will be. So we called as tuple packing. So the next one is the next one is dictionary creation of dictionary for this is how we create a dictionary. Before looking at how to create a dictionary let me just tell you what dictionary is? So dictionary in Python is an unordered collection of data values, which is used to store data values like a map.

Unlike other data types that hold only single values in an element. Dictionary holds key value pairs. So key values provided in the dictionary to make it more optimise. So in Python dictionary can be created by placing sequence of elements within the curly braces like this and you can separate your key value pairs by a comma and dictionary holds a pair of values. For example this is a pair of values one being the key and the other corresponding pair element being its key value.

So here one is the key and key value here is first and values in a dictionary can be of any data type and can be duplicated as well. But you can have only unique keys, but you can have duplicated values. So the how we create a dictionary so dict sample as a dictionary now. Dictionary can also be created by the built-in function using dict that an empty dictionary can be created by just placing elements inside the curly braces.

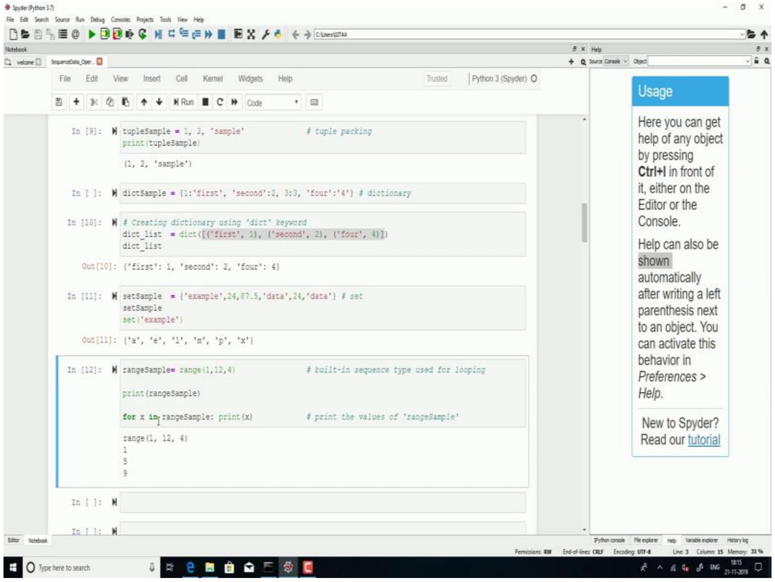
But you can also use the dist function to create a dictionary. So here I am creating the same set of dictionaries using dict built in function. So I am just given the key values as a tuple. So, each of the sets will be chosen as key value pairs. If you print the dict if you print the dictionary you will be getting the keys and values. Set sequence type basically sets may contain may consists of various elements, the order of elements in a set is undefined.

Any immutable data type can be an element of a set. So here I am using string, integer as elements of the set, for example a number a string a tuple. Here I am using string and numbers. So here I have an example. A set sample for one thing to note here is list cannot be an element of a set. As we have given for dictionary and pass the elements as a list for creating a dictionary, but that cannot be done to create a set because this cannot be an element of set. And another that cannot be an element of a set as well.

So you cannot create a set using a set itself you cannot have it as an element. You can define a set as simple as by meaning all its elements in brackets. So you can use the same curly brackets to create a set or you can also use the function that is called as set that is inbuilt so you can use set function to create set as well. So, the only exception is empty set which can be created using the functions set.

So if you are using the set function, you should use list, string or tuple as a parameter so it will return a set consists of its elements. For example, I am creating a set here. So it if you print it will return the set consists of its elements actually. So that is how were you create a set using a set function.

(Refer Slide Time: 10:58)



```
In [9]:  tupleSample = 1, 2, 'sample'            # tuple packing
         print(tupleSample)

         (1, 2, 'sample')

In [ ]:  dictSample = {1:'first', 'second':2, 3:3, 'four':'4'} # dictionary

In [10]: # Creating dictionary using 'dict' keyword
         dict_list  = dict([('first', 1), ('second', 2), ('four', 4)])
         dict_list

Out[10]: {'first': 1, 'second': 2, 'four': 4}

In [11]: setSample  = {'example',24,87.5,'data',24,'data'} # set
         setSample
         set('example')

Out[11]: {'a', 'e', 'l', 'm', 'p', 'x'}

In [12]: rangeSample= range(1,12,4)             # built-in sequence type used for looping

         print(rangeSample)

         for x in rangeSample: print(x)         # print the values of 'rangeSample'

         range(1, 12, 4)
         1
         5
         9

In [ ]:

In [ ]:
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\GTAA
SequenceData_Oper...
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (Spyder)
Run
Code
Source Console Object
IPython console File explorer Help Variable explorer History log
Type here to search

So the next thing that we can see here is range which is also the sequence data type in Python and Range is a built-in function of python. So you can use that whenever you want to perform any action for a specific number of time and it is commonly used in for looping and hence knowledge of same key aspect when you want to deal with any kind of python coding. Basically the range function takes in three arguments.

The first argument is a start value with integers starting from which the sequence of integers to be written. Next value is the stop value. So if you give 12 then you will be getting a value one before that that is n minus 1 so you will be getting a value before 12 that is 11. The last argument is the stop value you can give an integer before which the sequence of integers is to be written. So let us see. I am giving 1, 12, 4.

So I am going to create a sequence of value ranging from 1 to 12. Since I have given the step size is 4 you will be getting the output as 0.5 and 9 so that is how the range will be that is how the range function works. So here also I am using the, for loop to print the values of the range. If you just use the print function to get the values, you just get the input of your function.

So in this lecture what are sequence data types and then we have also seen some of the sequence data type that can be used as containers to store the data. Some of it are strings, tuples, lists, arrays, dictionary sets and range and we have also seen how to create a how to initialise the object for all of the sequence data type. In the upcoming lectures we will be seeing how to perform some operations on each of the sequence data type. Thank you.