str type:

str represents String data type.

A String is a sequence of characters enclosed within single quotes or double quotes.

By using single quotes or double quotes we cannot represent multi line string literals.

For this requirement we should go for triple single quotes(''') or triple double quotes(""")

We can also use triple quotes to use single quote or double quote in our String.

''' This is " character'''
' This i " Character '
We can embed one string in another string

'''This "Python class very helpful" for java students'''

Slicing of Strings:

slice means a piece
[ ] operator is called slice operator,which can be used to retrieve parts of String.
In Python Strings follows zero based index.
The index can be either +ve or -ve.
+ve index means forward direction from Left to Right
-ve index means backward direction from Right to Left

Note:
1. In Python the following data types are considered as Fundamental Data types

- int
- float
- complex
- bool
- str

2. In Python,we can represent char values also by using str type and explicitly char type is not available.

Eg:

```
1) >>> c='a'
2) >>> type(c)
3) <class 'str'>
```

We can convert one type value to another type. This conversion is called Typecasting or Type coersion.
The following are various inbuilt functions for type casting.

1. int()
2. float()
3. complex()
4. bool()
5. str()

1.int():

We can use this function to convert values from other types to int

Note:

1. We can convert from any type to int except complex type.
2. If we want to convert str type to int type, compulsary str should contain only integral value and should be specified in base-10

2. float():

We can use float() function to convert other type values to float type.

Note:

1. We can convert any type value to float type except complex type.

2. Whenever we are trying to convert str type to float type compulsary str should be either integral or floating point literal and should be specified only in base-10.

3.complex():

We can use complex() function to convert other types to complex type.

Form-1: complex(x)
We can use this function to convert x into complex number with real part x and imaginary part 0.

Form-2: complex(x,y)

We can use this method to convert x and y into complex number such that x will be real part and y will be imaginary part.

4. bool():

We can use this function to convert other type values to bool type.

5. str():

We can use this method to convert other type values to str type

bytes Data Type:

bytes data type represens a group of byte numbers just like an array.

Conclusion 1:

The only allowed values for byte data type are 0 to 256. By mistake if we are trying to provide any other values then we will get value error.

Conclusion 2:

Once we creates bytes data type value, we cannot change its values,otherwise we will get TypeError.

bytearray Data type:

bytearray is exactly same as bytes data type except that its elements can be modified.

list data type:

If we want to represent a group of values as a single entity where insertion order required to preserve and duplicates are allowed then we should go for list data type.

1. insertion order is preserved
2. heterogeneous objects are allowed
3. duplicates are allowed
4. Growable in nature
5. values should be enclosed within square brackets.

list is growable in nature. i.e based on our requirement we can increase or decrease the size.

tuple data type:

tuple data type is exactly same as list data type except that it is immutable.i.e we cannot chage values.

Tuple elements can be represented within parenthesis.

Note: tuple is the read only version of list

range Data Type:

range Data Type represents a sequence of numbers.
The elements present in range Data type are not modifiable. i.e range Data type is immutable.

Form-1: range(10)
generate numbers from 0 to 9

Eg:
```
r=range(10)
for i in r : print(i)    0 to 9
```

Form-2: range(10,20)

generate numbers from 10 to 19

```
r = range(10,20)
for i in r : print(i)    10 to 19
```

Form-3: range(10,20,2)

2 means increment value

```
r = range(10,20,2)
for i in r : print(i)    10,12,14,16,18
```

set Data Type:

If we want to represent a group of values without duplicates where order is not important then we should go for set Data Type.

1. insertion order is not preserved
2. duplicates are not allowed
3. heterogeneous objects are allowed
4. index concept is not applicable
5. It is mutable collection
6. Growable in nature

frozenset Data Type:

It is exactly same as set except that it is immutable.
Hence we cannot use add or remove functions.

dict Data Type:

If we want to represent a group of values as key-value pairs then we should go for dict data type.

Duplicate keys are not allowed but values can be duplicated. If we are trying to insert an entry with duplicate key then old value will be replaced with new value.

None Data Type:

None means Nothing or No value associated.
If the value is not available,then to handle such type of cases None introduced.
It is something like null value in Java.

Eg:

```
def  m1():
  a=10

print(m1())
  None
```

Escape Characters:

In String literals we can use esacpe characters to associate a special meaning.

The following are various important escape characters in Python

1) \n==>New Line
2) \t===>Horizontal tab
3) \r ==>Carriage Return
4) \b===>Back space
5) \f===>Form Feed
6) \v==>Vertical tab
7) \'===>Single quote
8) \"===>Double quote
9) \\===>back slash symbol

....

Constants:

Constants concept is not applicable in Python.