Python Data Types

B. Ramar(Ph.D)
Resource Faculty,
AU TVS Centre for Quality Management, Anna University.

Content

1. Numbers
2. List
3. Tuple
4. Strings
5. Set
6. Dictionary
7. Conversion between data types

Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

type(a)

Numbers

int

```
a = 5
print(a, "is of type", type(a))
```

float

```
a = 2.0
print(a, "is of type", type(a))
```

complex

```
a = 1+2j
print(a, "is complex number?", isinstance(1+2j,complex))
```

```
>>> a = 12345678901234567 89
>>> a
12345678901234567 89
>>> b = 0.12345678901234567 89
>>> b
0.1234567890123456 8
>>> c = 1+2j
>>> c
(1+2j)
```

List

List is an ordered sequence of items. It is one of the most used data type in Python and is very flexible.

All the items in a list do not need to be of the same type.

1. Items separated by commas are enclosed within brackets [ ].

```
a = [1, 2.2, 'python']
```

Lists are mutable, meaning, value of elements of a list can be altered.

```
a = [5,10,15,20,25,30,35,40]

# a[2] = 15
print("a[2] = ", a[2])

# a[0:3] = [5, 10, 15]
print("a[0:3] = ", a[0:3])

# a[5:] = [30, 35, 40]
print("a[5:] = ", a[5:])
```

Tuple

Tuple is an ordered sequence of items same as list. The only difference is that tuples are immutable. Tuples once created cannot be modified.

Tuples are used to write-protect data and are usually faster than list as it cannot change dynamically.

It is defined within parentheses ().

```
>>> t = (5,'program', 1+3j)
```

```
t = (5,'program', 1+3j)

# t[1] = 'program'
print("t[1] = ", t[1])

# t[0:3] = (5, 'program', (1+3j))
print("t[0:3] = ", t[0:3])

# Generates error
# Tuples are immutable
t[0] = 10
```

Strings

String is sequence of Unicode characters. We can use single quotes or double quotes to represent strings. Multi-line strings can be denoted using triple quotes, ''' or """.

```
>>> s = "This is a string"
>>> s = '''a multiline
```

```
s = 'Hello world!'

# s[4] = 'o'
print("s[4] = ", s[4])

# s[6:11] = 'world'
print("s[6:11] = ", s[6:11])

# Generates error
# Strings are immutable in Python
s[5] ='d'
```

Set

Set is an unordered collection of unique items. Set is defined by values separated by comma inside braces { }. Items in a set are not ordered.

```
>>> a = {1,2,2,3,3,3}
>>> a
{1, 2, 3}
```

```
a = {5,2,3,1,4}

# printing set variable
print("a = ", a)

# data type of variable a
print(type(a))
```

Since, set are unordered collection, indexing has no meaning. Hence the slicing operator [ ] does not work.

```
>>> a = {1,2,3}
>>> a[1]
```

Dictionary

1. Dictionary is an unordered collection of key-value pairs.
2. It is generally used when we have a huge amount of data.
3. Dictionaries are optimized for retrieving data.
4. dictionaries are defined within braces {} with each item being a pair in the form key:value
5. Key and value can be of any type .

```
>>> d = {1:'value','key':2}
>>> type(d)
```

```
d = {1:'value','key':2}
print(type(d))

print("d[1] = ", d[1]);

print("d['key'] = ", d['key']);

# Generates error
print("d[2] = ", d[2]);
```

Conversion between data types

We can convert between different data types by using different type conversion functions like int(), float(), str() etc.

compatible values

```
>>> float(5)
>>> int(10.6)
>>> int(-10.6)
>>> float('2.5')
>>> str(25)
>>> int('1p')
```

convert one sequence

```
set([1,2,3])
tuple({5,6,7})
list('hello')
```

convert to dictionary

```
>>> dict([[1,2],[3,4]])
>>> dict([(3,26),(4,44)])
```

Content

1. Python Numbers
2. Python List
3. Python Tuple
4. Python Strings
5. Python Set
6. Python Dictionary
7. Conversion between data types