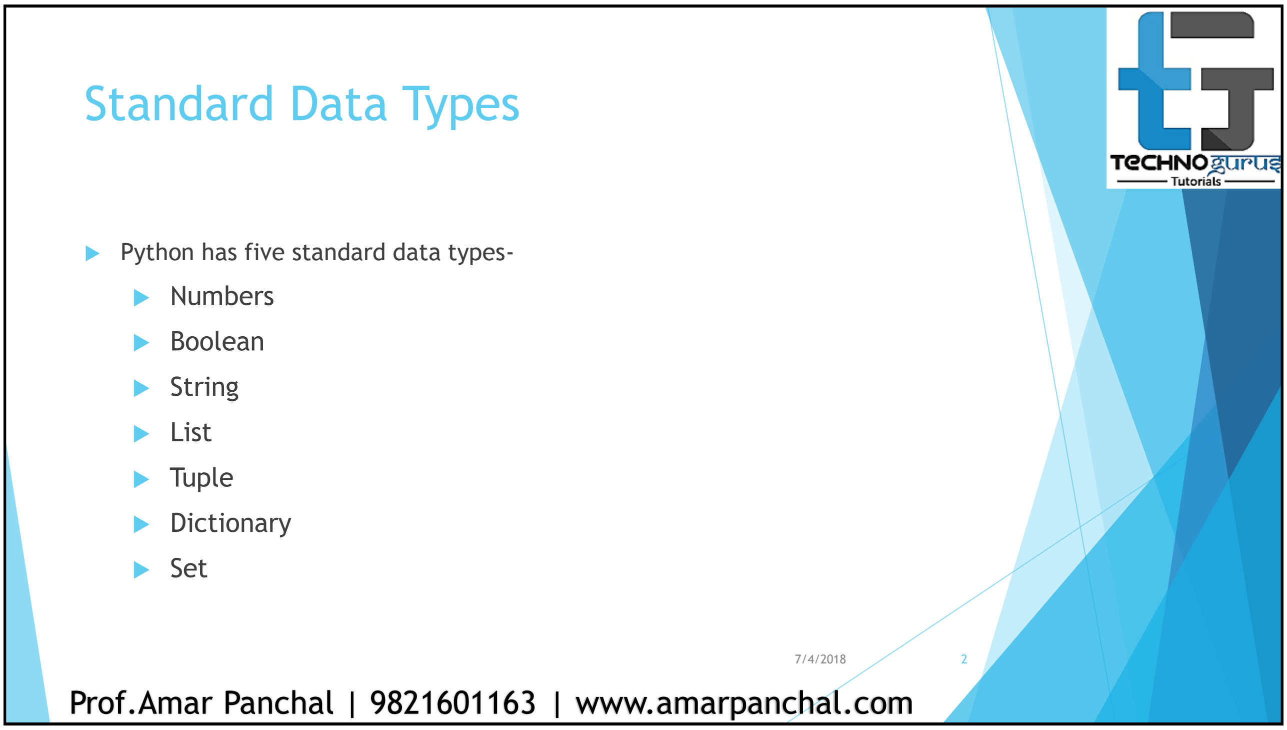
PYTHON DATATYPES-METHODS

BY

AMAR PANCHAL

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com

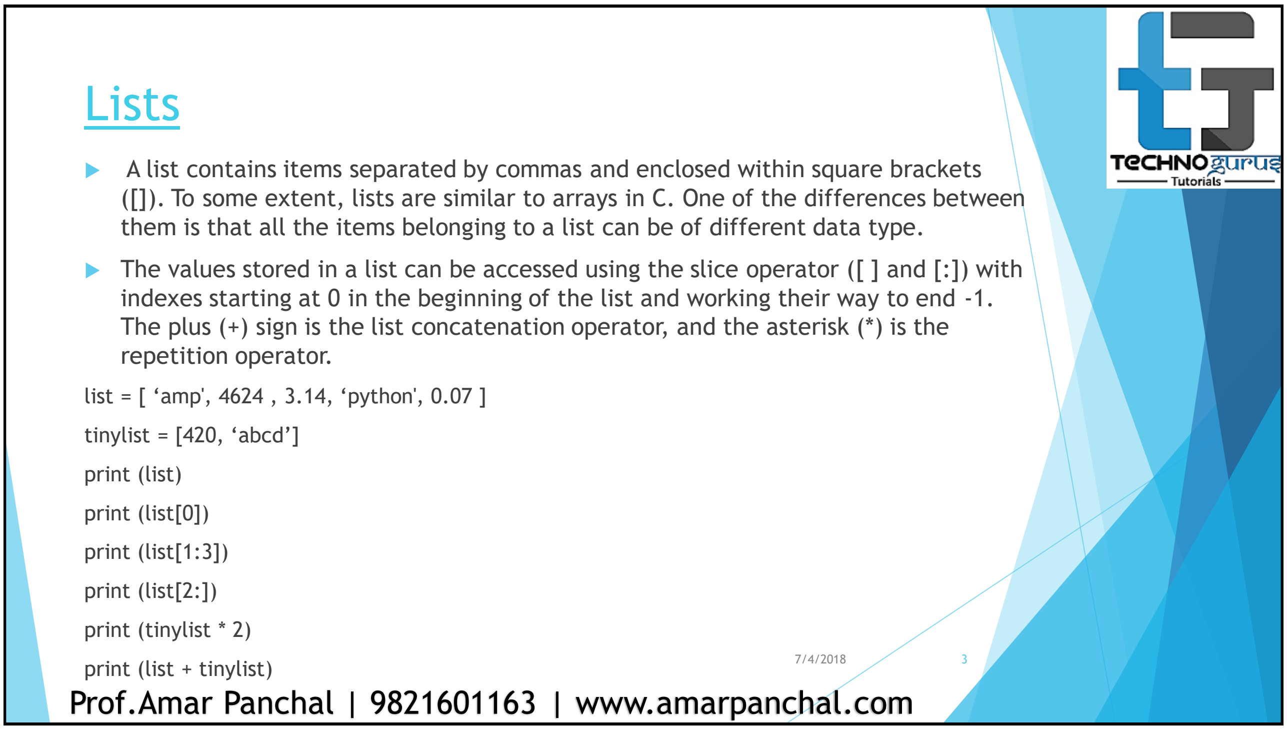


Standard Data Types

- Python has five standard data types-
  - Numbers
  - Boolean
  - String
  - List
  - Tuple
  - Dictionary
  - Set

7/4/2018 2

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



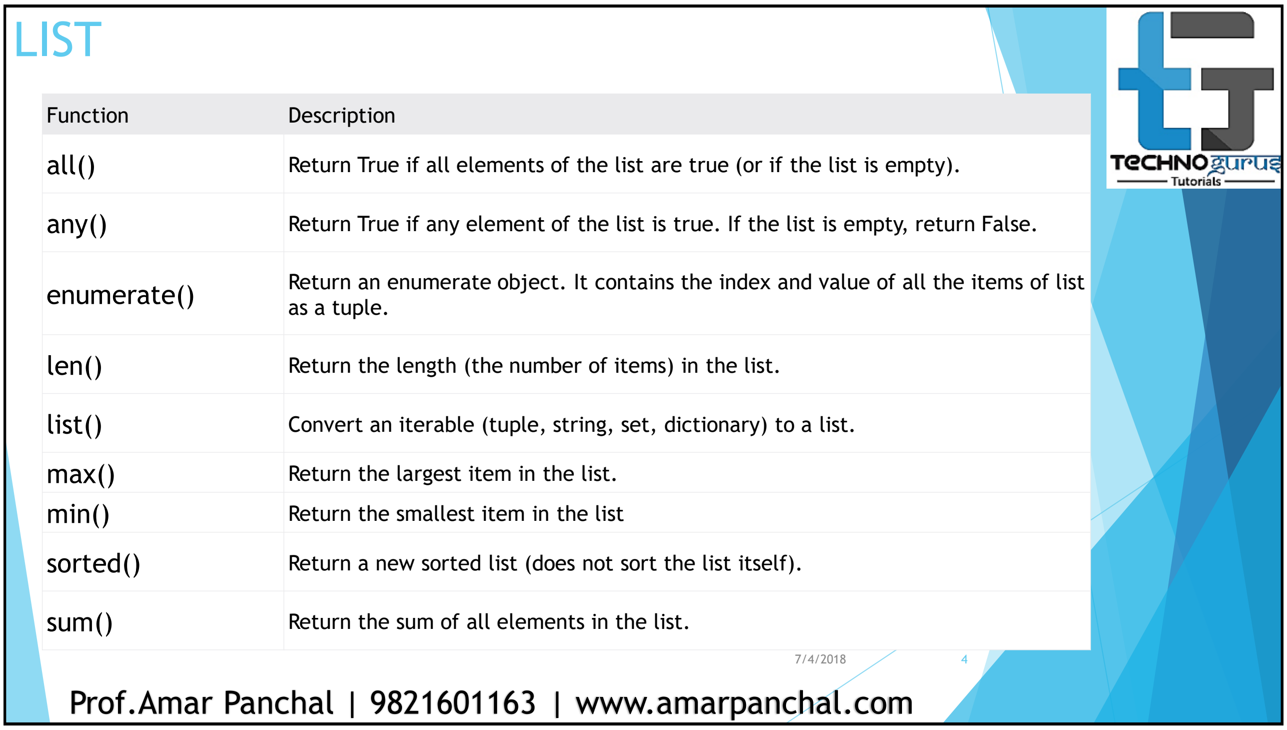
Lists

- A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One of the differences between them is that all the items belonging to a list can be of different data type.
- The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

```
list = [ ‘amp', 4624 , 3.14, ‘python', 0.07 ]
tinylist = [420, ‘abcd’]
print (list)
print (list[0])
print (list[1:3])
print (list[2:])
print (tinylist * 2)
print (list + tinylist)
```

7/4/2018 3

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com

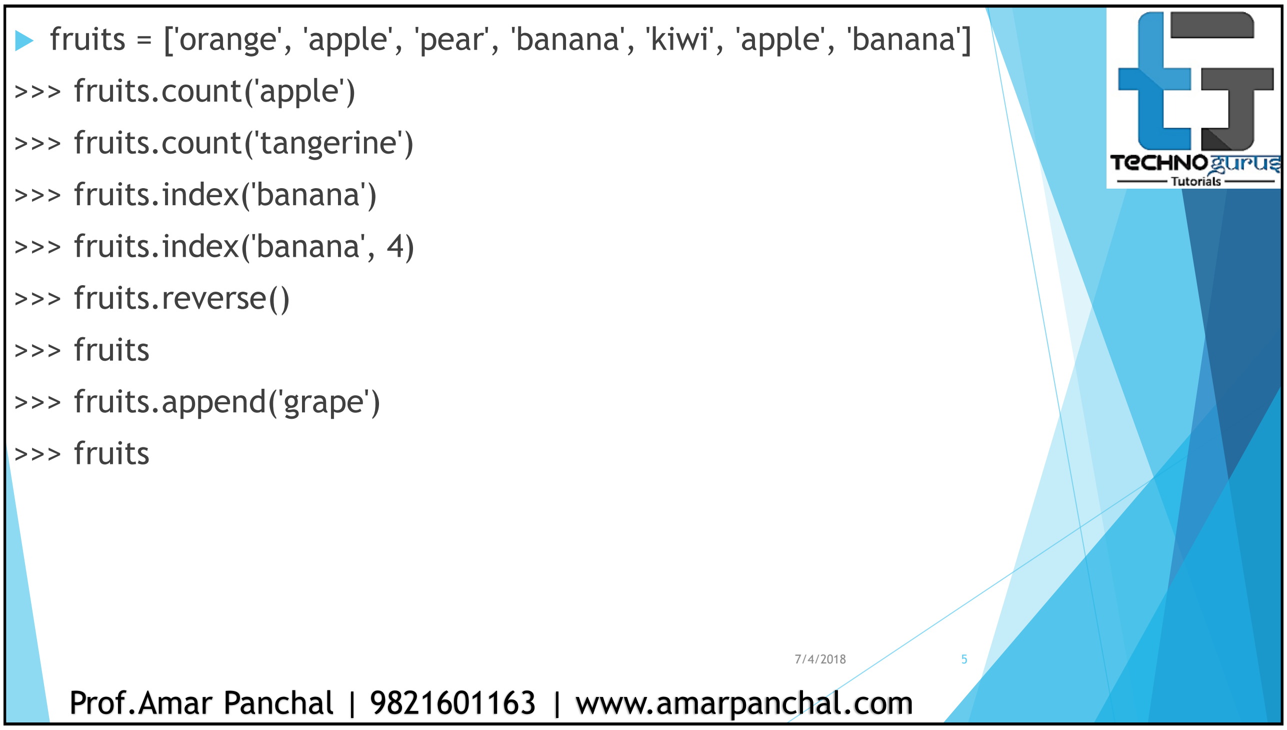


LIST

Function	Description
all()	Return True if all elements of the list are true (or if the list is empty).
any()	Return True if any element of the list is true. If the list is empty, return False.
enumerate()	Return an enumerate object. It contains the index and value of all the items of list as a tuple.
len()	Return the length (the number of items) in the list.
list()	Convert an iterable (tuple, string, set, dictionary) to a list.
max()	Return the largest item in the list.
min()	Return the smallest item in the list
sorted()	Return a new sorted list (does not sort the list itself).
sum()	Return the sum of all elements in the list.

7/4/2018 4

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com

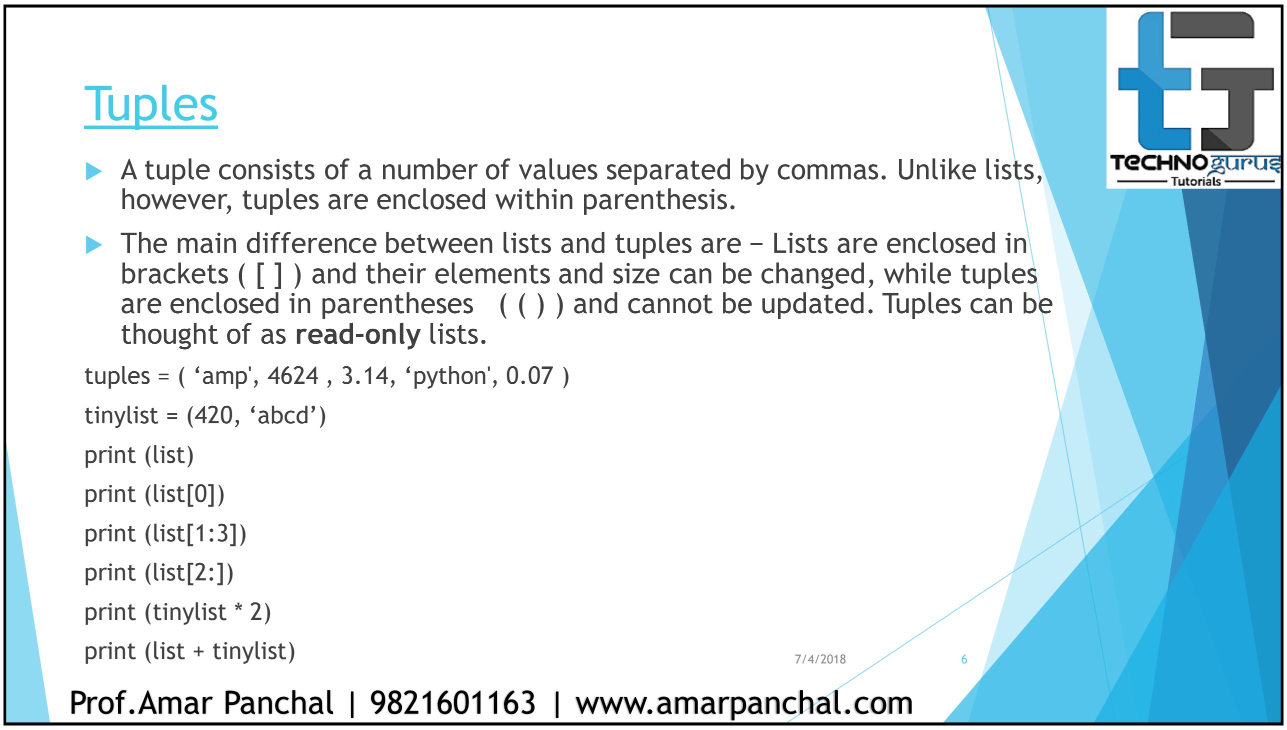


- fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']

```
>>> fruits.count('apple')
>>> fruits.count('tangerine')
>>> fruits.index('banana')
>>> fruits.index('banana', 4)
>>> fruits.reverse()
>>> fruits
>>> fruits.append('grape')
>>> fruits
```

7/4/2018 5

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



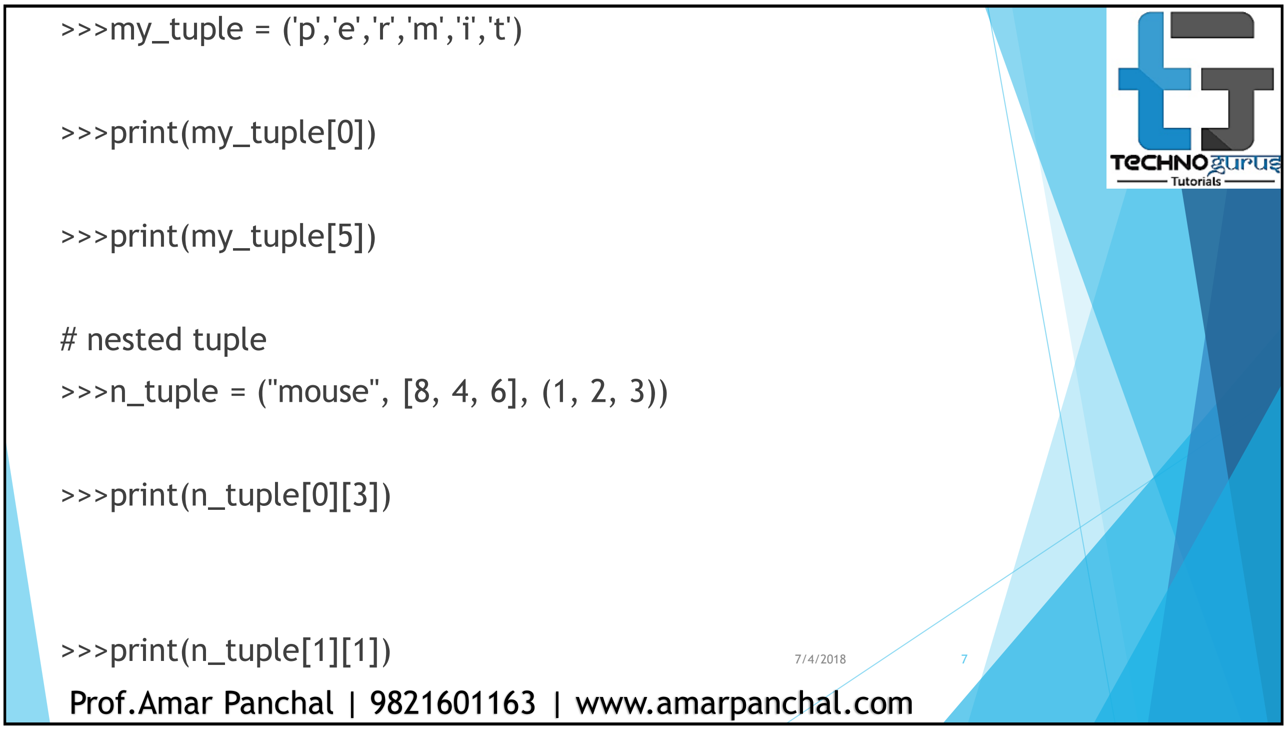
Tuples

- A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parenthesis.
- The main difference between lists and tuples are − Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as **read-only** lists.

```
tuples = ( ‘amp', 4624 , 3.14, ‘python', 0.07 )
tinylist = (420, ‘abcd’)
print (list)
print (list[0])
print (list[1:3])
print (list[2:])
print (tinylist * 2)
print (list + tinylist)
```

7/4/2018 6

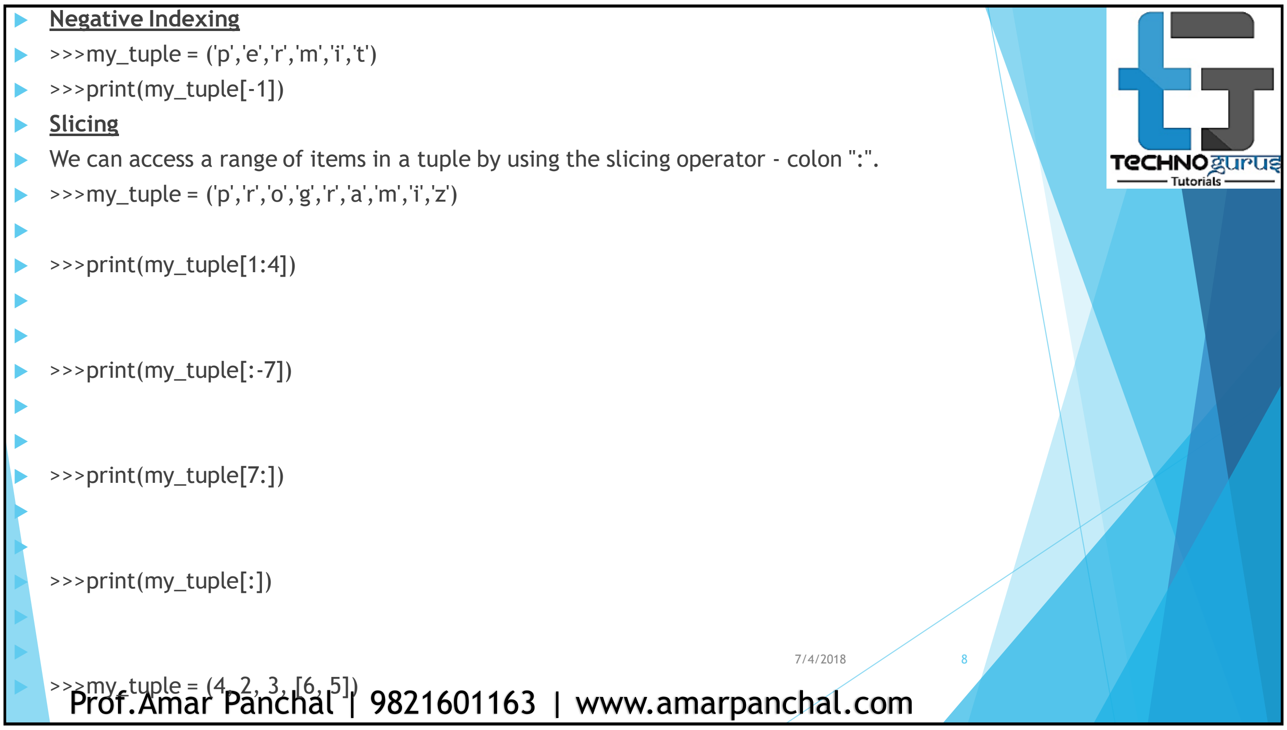
Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



```
>>>my_tuple = ('p','e','r','m','i','t')

>>>print(my_tuple[0])

>>>print(my_tuple[5])

# nested tuple
>>>n_tuple = ("mouse", [8, 4, 6], (1, 2, 3))

>>>print(n_tuple[0][3])



>>>print(n_tuple[1][1])
```

7/4/2018 7

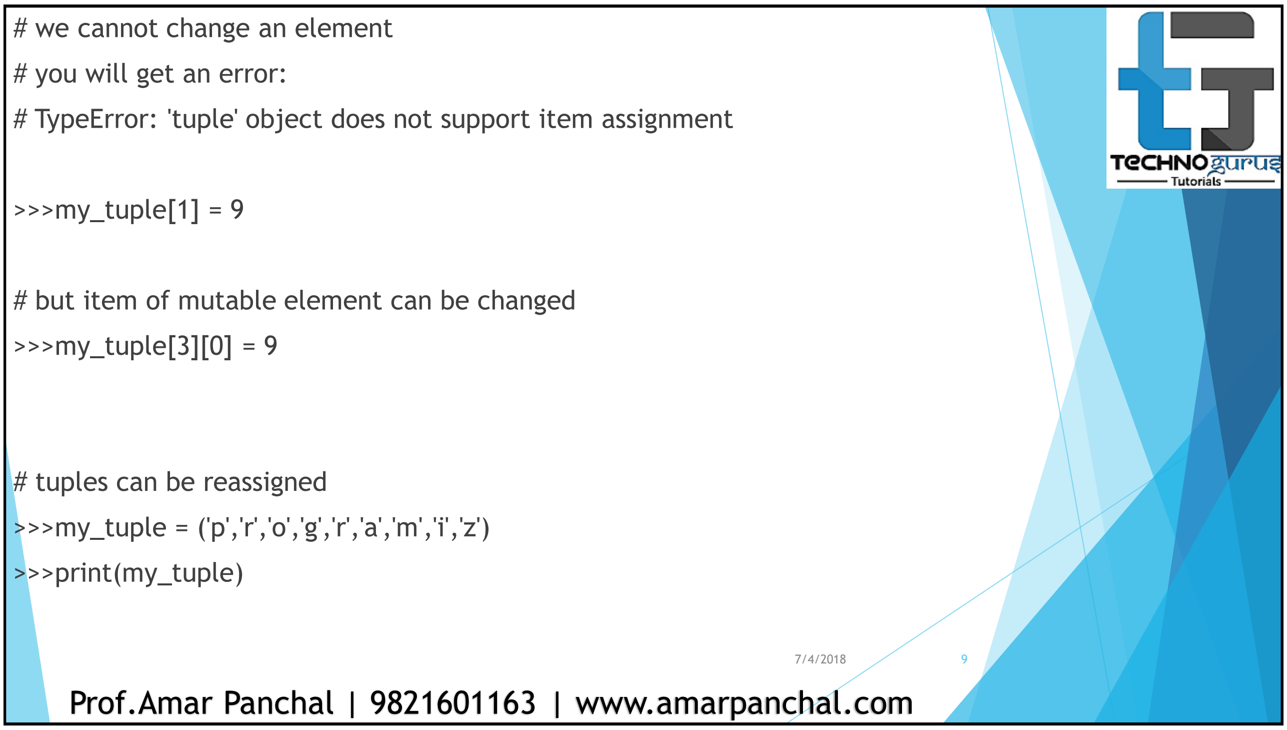
Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



- **Negative Indexing**
- >>>my_tuple = ('p','e','r','m','i','t')
- >>>print(my_tuple[-1])
- **Slicing**
- We can access a range of items in a tuple by using the slicing operator - colon ":".
- >>>my_tuple = ('p','r','o','g','r','a','m','i','z')
- >>>print(my_tuple[1:4])
- >>>print(my_tuple[:-7])
- >>>print(my_tuple[7:])
- >>>print(my_tuple[:])
- >>>my_tuple = (4, 2, 3, [6, 5])

7/4/2018 8

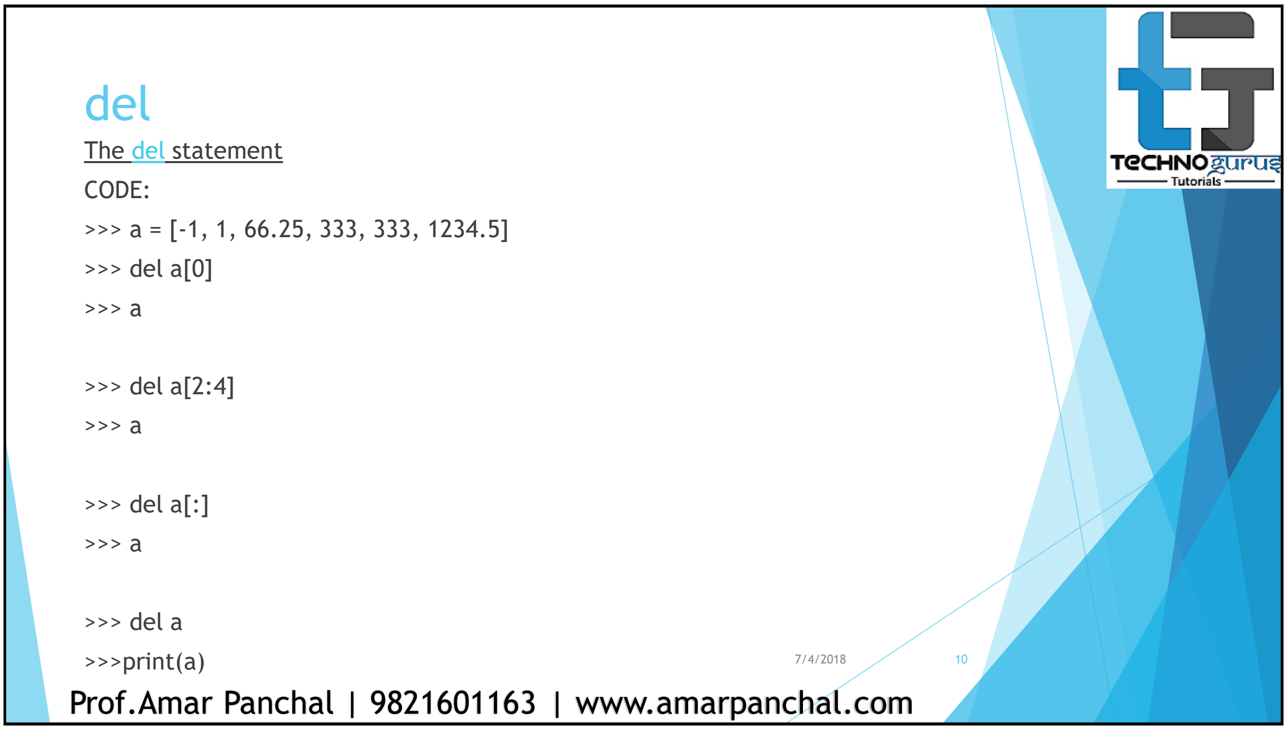
Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



```
# we cannot change an element
# you will get an error:
# TypeError: 'tuple' object does not support item assignment

>>>my_tuple[1] = 9

# but item of mutable element can be changed
>>>my_tuple[3][0] = 9


# tuples can be reassigned
>>>my_tuple = ('p','r','o','g','r','a','m','i','z')
>>>print(my_tuple)
```

7/4/2018 9

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



del

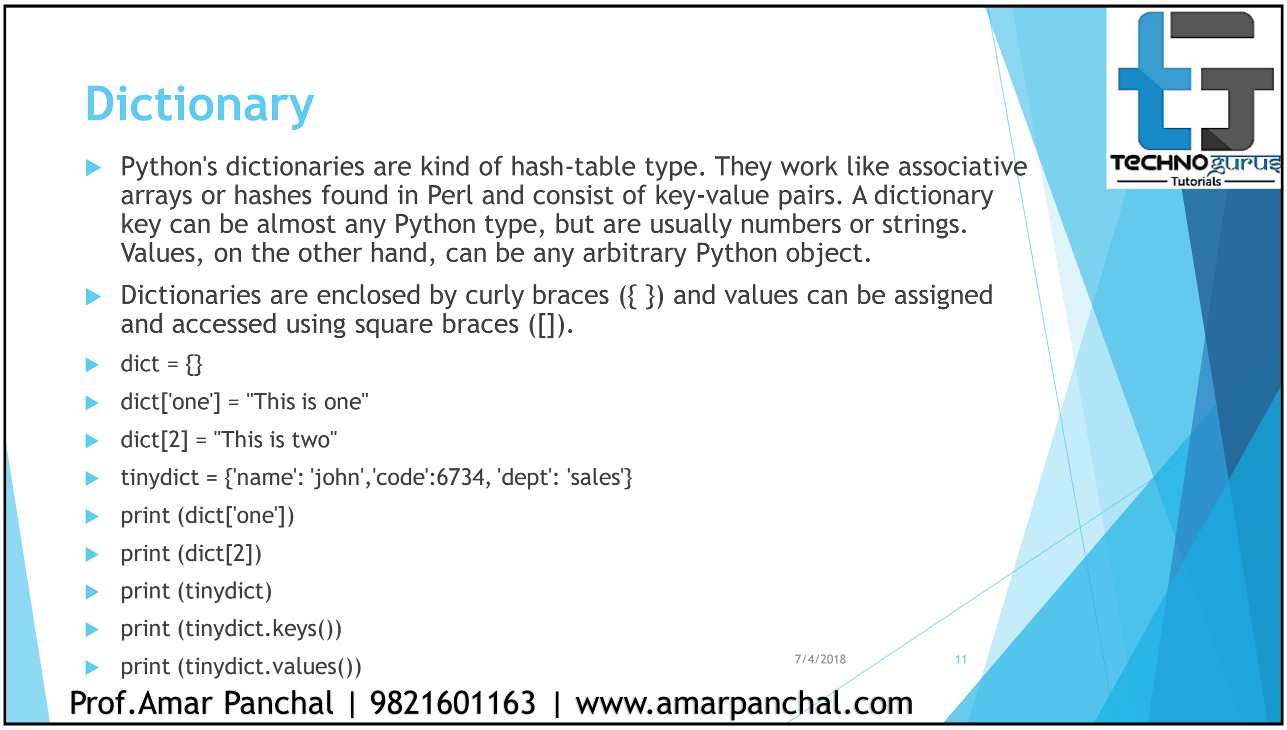
The del statement

CODE:

```
>>> a = [-1, 1, 66.25, 333, 333, 1234.5]
>>> del a[0]
>>> a

>>> del a[2:4]
>>> a

>>> del a[:]
>>> a

>>> del a
>>>print(a)
```

7/4/2018 10

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com

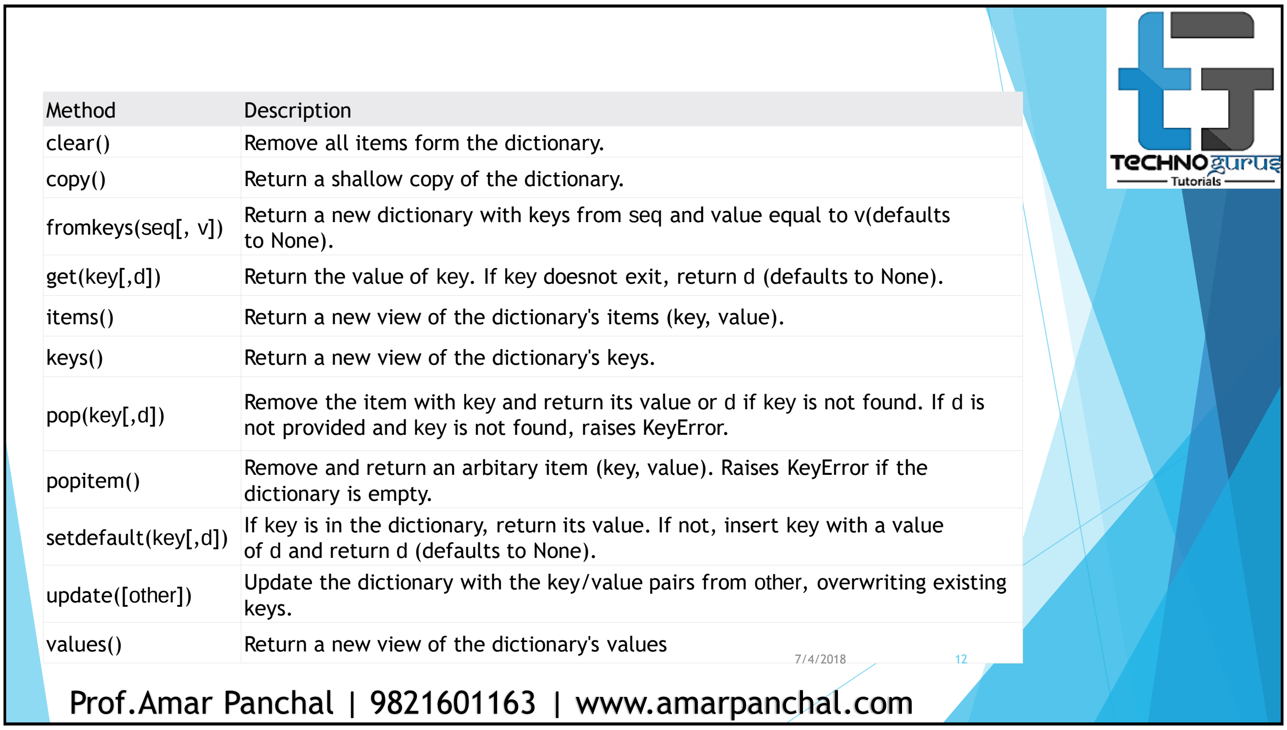


Dictionary

- Python's dictionaries are kind of hash-table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.
- Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).
- dict = {}
- dict['one'] = "This is one"
- dict[2] = "This is two"
- tinydict = {'name': 'john','code':6734, 'dept': 'sales'}
- print (dict['one'])
- print (dict[2])
- print (tinydict)
- print (tinydict.keys())
- print (tinydict.values())

7/4/2018 11

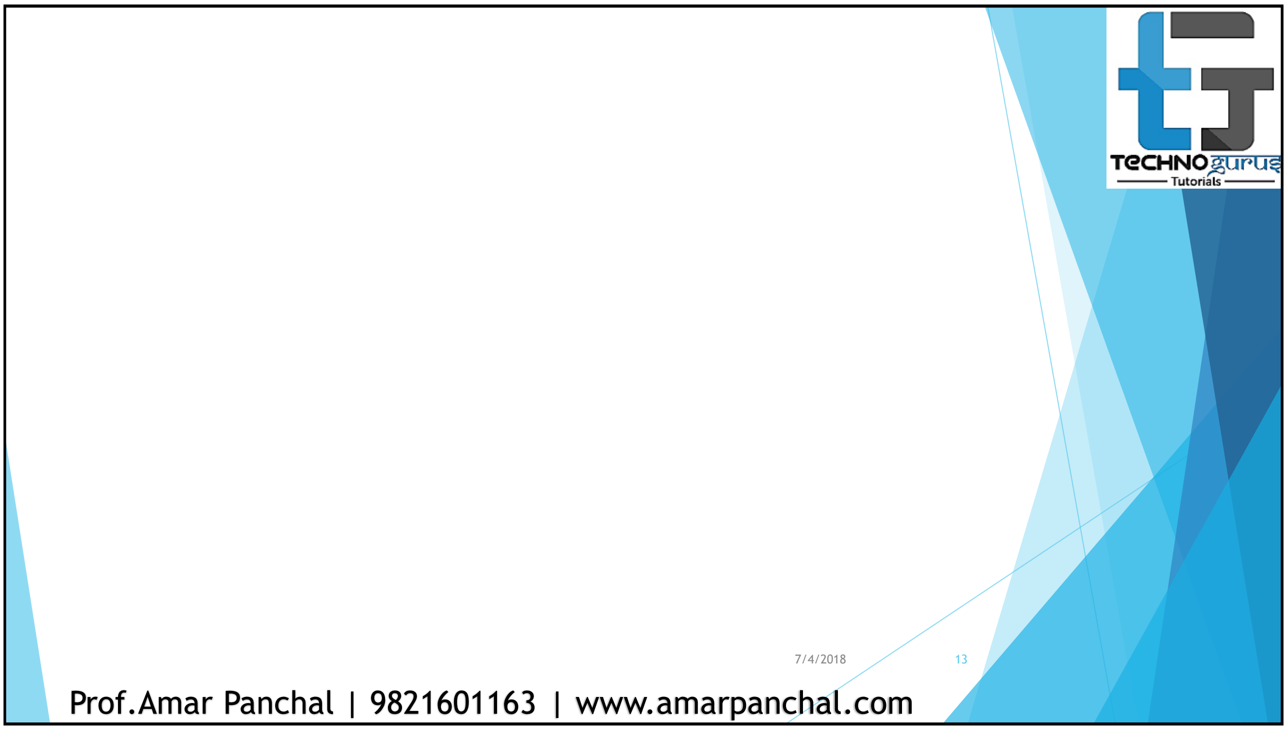
Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



Method	Description
clear()	Remove all items form the dictionary.
copy()	Return a shallow copy of the dictionary.
fromkeys(seq[, v])	Return a new dictionary with keys from seq and value equal to v(defaults to None).
get(key[,d])	Return the value of key. If key doesnot exit, return d (defaults to None).
items()	Return a new view of the dictionary's items (key, value).
keys()	Return a new view of the dictionary's keys.
pop(key[,d])	Remove the item with key and return its value or d if key is not found. If d is not provided and key is not found, raises KeyError.
popitem()	Remove and return an arbitary item (key, value). Raises KeyError if the dictionary is empty.
setdefault(key[,d])	If key is in the dictionary, return its value. If not, insert key with a value of d and return d (defaults to None).
update([other])	Update the dictionary with the key/value pairs from other, overwriting existing keys.
values()	Return a new view of the dictionary's values

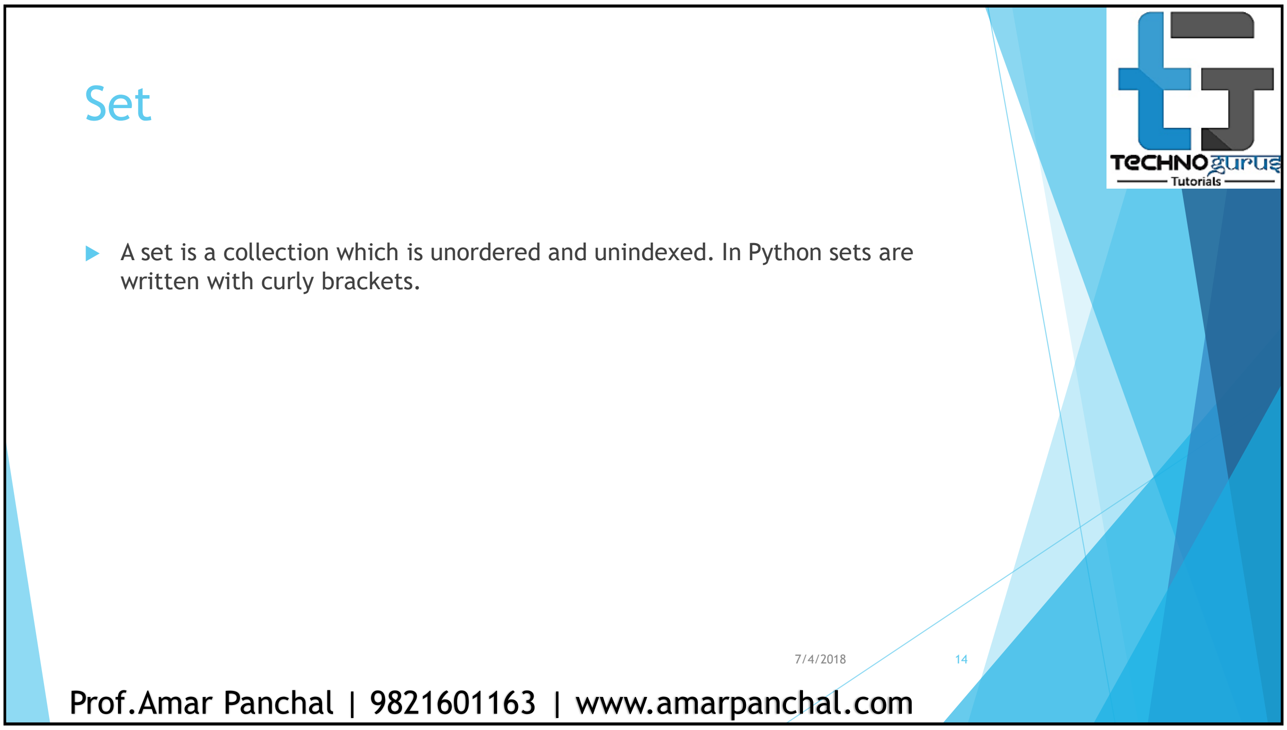
7/4/2018 12

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



7/4/2018 13

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



Set

- A set is a collection which is unordered and unindexed. In Python sets are written with curly brackets.

7/4/2018 14

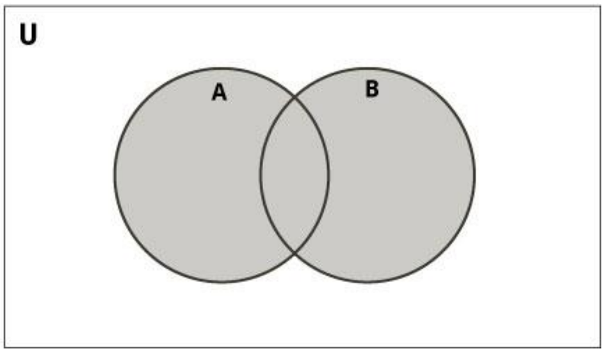
Prof.Amar Panchal | 9821601163 | www.amarpanchal.com

Python Set Methods

Method	Description
add()	Add an element to a set
clear()	Remove all elements form a set
copy()	Return a shallow copy of a set
difference()	Return the difference of two or more sets as a new set
difference_update()	Remove all elements of another set from this set
discard()	Remove an element from set if it is a member. (Do nothing if the element is not in set)
intersection()	Return the intersection of two sets as a new set
intersection_update()	Update the set with the intersection of itself and another
isdisjoint()	Return True if two sets have a null intersection
issubset()	Return True if another set contains this set
issuperset()	Return True if this set contains another set
pop()	Remove and return an arbitary set element. Raise KeyErrorif the set is empty
remove()	Remove an element from a set. If the element is not a member, raise a KeyError
symmetric_difference()	Return the symmetric difference of two sets as a new set
symmetric_difference_update()	Update a set with the symmetric difference of itself and another
union()	Return the union of sets in a new set
update()	Update a set with the union of itself and others

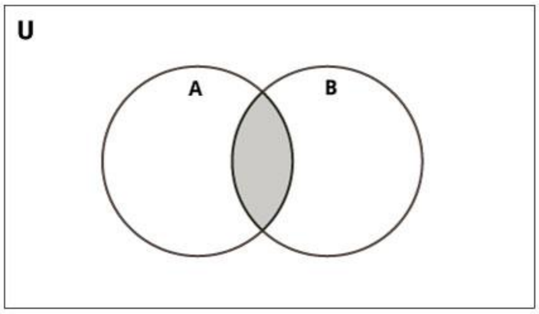
Set Union

```
# initialize A and B
A = {1, 2, 3, 4, 5}
B = {4, 5, 6, 7, 8}
# use | operator
print(A | B)
```



Set Intersection

```
# initialize A and B
A = {1, 2, 3, 4, 5}
B = {4, 5, 6, 7, 8}
# use & operator#
print(A & B)
```

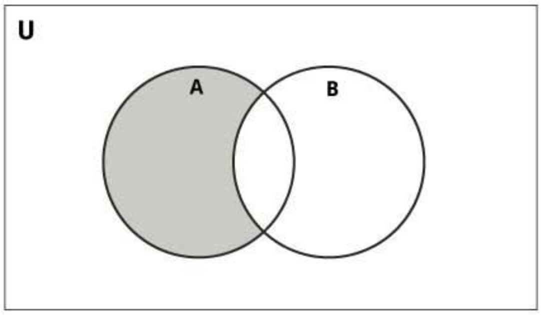


7/4/2018 17

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com

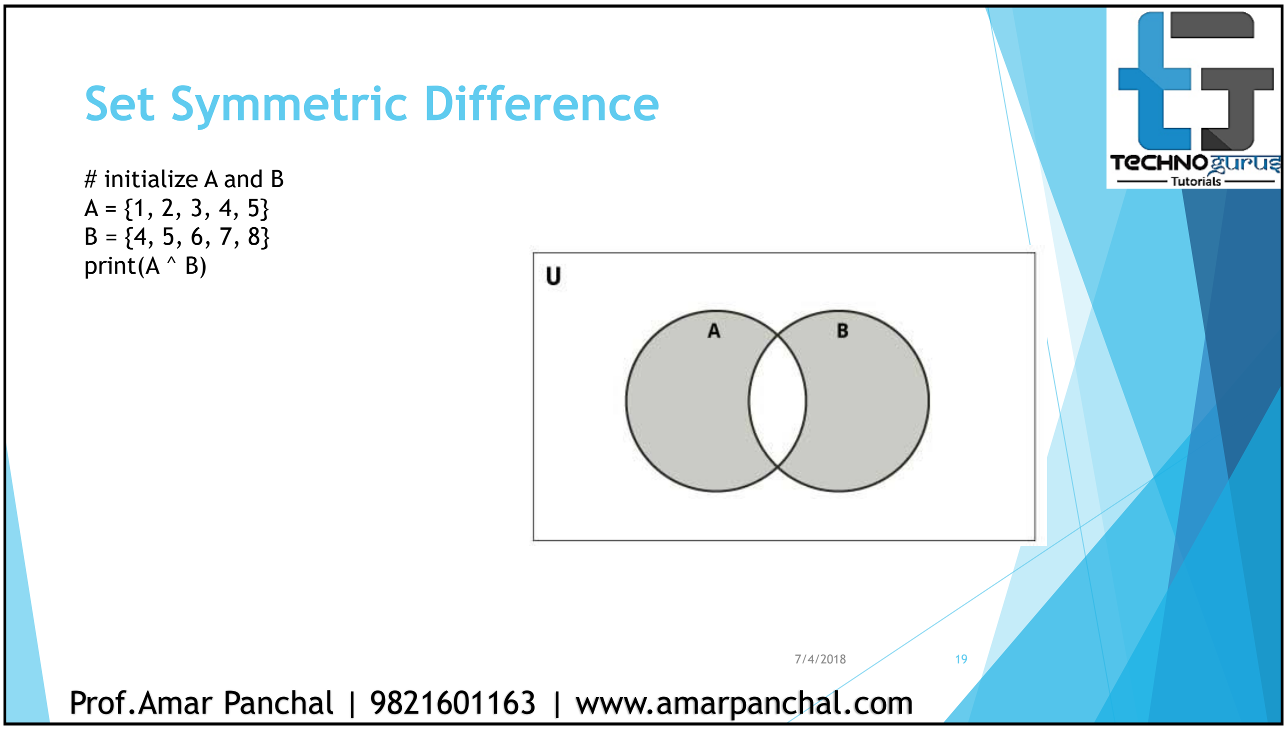
Set Difference

```
# initialize A and B
A = {1, 2, 3, 4, 5}
B = {4, 5, 6, 7, 8}
print(A - B)
```



7/4/2018 18

Prof.Amar Panchal | 9821601163 | www.amarpanchal.com



Set Symmetric Difference

```
# initialize A and B
A = {1, 2, 3, 4, 5}
B = {4, 5, 6, 7, 8}
print(A ^ B)
```