

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING,

PULCHOWK CAMPUS

A PROJECT ON

Python Built in datatype methods



Python Built-in Datatype Methods

STRING METHODS:

1. capitalize()

used to capitalize (uppercase) only the first letter of a sentence.

Example:

str = "python"

print("Initial string = ",str);

  res = str.capitalize()

  print("Updated string = ",res);

Output:

  Initial string =  python Updated string =  Python

2.casefold()

used to convert a string into lowercase.

Example:

str = "PYTHON"

print("Initial string = ",str);

  res = str.casefold()

  print("Updated string = ",res);

  Output:  Initial string =  PYTHON Updated string =  python

3.center()

used to align a string centrally. You can also use any character as a fill character for each size, but this is optional. The syntax is as follows:

  string.center(len, char)

Above, len is the length of the string, whereas, char is the character to be filled on the left and right sides.

Example:

str = "DEMO"

print("Initial string = ",str);

  res = str.center(10,"#")

print("Updated string = ",res);

Output:

Initial string =  DEMO Updated string =  ###DEMO###

4.count()

used to search for a specific value and its count of appearance in the string. The syntax is as follows

  string.count(str, begn, end )

Above, str is the string, begn is the position to start the search, and end is the position to end.

Example:

str = "This is one."

print("String = ",str);

  res = str.count("This",0, 15)

print("Count of specific value = ",res);

Output:

  String =  This is one. Count of specific value =  1

5.index()

used to find the index of the first occurrence of a specific value.

syntax:

string.index(str, begin, end)

Above, str is the value to be searched, begin is where the search begins and

end is where it ends.

example:

str = "This is first text."

print("String = ",str);

res = str.index("i")

print("The first occurrence of the specific value found at index =",res)

Output:

String = This is first text.

The first occurrence of the specific value found at index = 2

6.upper()

used to convert the lowercase letters in the string to uppercase.

example:

str = "The Walking Dead"

print("String = ",str)

res = str.upper()

print("Updated String with all letters in uppercase now =", res)

output:

String = The Walking Dead

Updated String with all letters in uppercase now = THE WALKING DEAD

7.lower()

used to convert the uppercase letters in the string to lowercase.

example:

str = "The Walking Dead"

print("String = ",str)

res = str.lower()

print("Updated String with all letters in lowercase =", res)

Output:

String = The Walking Dead

Updated String with all letters in lowercase = the walking dead

8. swapcase()

used swap the case i.e. convert lowercase to uppercase, and vice versa.

example:

str = "This is demo. This is another demo."

print("String = ",str)

res = str.swapcase()

print("Case Swapped =", res)

Example:

String = This is demo. This is another demo.

Case Swapped = tHIS IS DEMO. tHIS IS ANOTHER DEMO.

LIST METHODS:

1.append()

Adds an element at the end of the list

Example:

fruits = ['apple', 'banana', 'cherry']

fruits.append("orange")

print(fruits)

Output:

['apple', 'banana', 'cherry', , 'orange']

2.clear()

Removes all the elements from the list

Example:

fruits = ['apple', 'banana', 'cherry', 'orange']

fruits.clear()

print(fruits)

Output:

[]

3.copy()

Returns a copy of the list

Example:

fruits = ['apple', 'banana', 'cherry', 'orange']

x = fruits.copy()

print(x)

Output:

['apple', 'banana', 'cherry', 'orange']

4.count()

Returns the number of elements with the specified

Example:

fruits = ['apple', 'banana', 'cherry']

x = fruits.count("cherry")

print(x)

Output:

1

5.extend()

Add the elements of a list (or any iterable), to the end of the current list

Example:

fruits = ['apple', 'banana', 'cherry']

cars = ['Ford', 'BMW', 'Volvo']

fruits.extend(cars)

print(fruits)

Output:

['apple', 'banana', 'cherry', 'Ford', 'BMW', 'Volvo']

6.index()

Returns the index of the first element with the specified value

Example:

fruits = ['apple', 'banana', 'cherry']

x = fruits.index("cherry")

print(x)

Output:

2

7.insert()

Adds an element at the specified position

Example:

fruits = ['apple', 'banana', 'cherry']

fruits.insert(1, "orange")

print(fruits)

Output:

['apple', 'orange', 'banana', 'cherry']

8.pop()

Removes the element at the specified position

Example:

fruits = ['apple', 'banana', 'cherry']

fruits.pop(1)

print(fruits)

Output:

['apple', 'cherry']

9.remove()

Removes the first item with the specified value

Example:

fruits = ['apple', 'banana', 'cherry']

fruits.remove("banana")

print(fruits)

Output:

['apple', 'cherry']

10.reverse()

Reverses the order of the list

Example:

fruits = ['apple', 'banana', 'cherry']

fruits.reverse()

print(fruits)

Output:

['cherry', 'banana', 'apple']

11.sort()

Sorts the list

Example:

cars = ['Ford', 'BMW', 'Volvo']

cars.sort()

print(car)

Output:

['BMW','Ford','Volvo']

DICTIONARY METHODS:

1.clear()

Removes all the elements from the dictionary

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car.clear()

print(car)

Output:

{}

2.copy()

Returns a copy of the dictionary

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.copy()

print(x)

Output:

{

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

3.fromkeys()

Returns a dictionary with the specified keys and value

Example:

x = ('key1', 'key2', 'key3')

y = 0

thisdict = dict.fromkeys(x, y)

print(thisdict)

Output:

{'key1':0,'key2':0,'key3':0}

4.get()

Returns the value of the specified key

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.get("model")

print(x)

Output:

Mustang

5.items()

Returns a list containing a tuple for each key value pair

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.items()

print(x)

Output:

dict\_items([('brand',Ford'),('model',Mustang'),( 'year',1964)])

6.keys()

Returns a list containing the dictionary's keys

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.keys()

print(x)

Output:

dict\_keys(['brand', 'model', 'year'])

7.pop()

Removes the element with the specified key

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car.pop("model")

print(car)

Output:

{'brand': 'Ford', 'year': 1964}

8.popitem()

Removes the last inserted key-value pair

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car.popitem()

print(car)

Output:

{'brand': 'Ford', 'model': 'Mustang'}

9.setdefault()

Returns the value of the specified key. If the key does not exist: insert the key, with the specified value

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.setdefault("model", "Bronco")

print(x)

Output:

Mustang

10.update()

Updates the dictionary with the specified key-value pairs

Example:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car.update({"color": "White"})

print(car)

Output:

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'White'}

11.values()

Returns a list of all the values in the dictionary

Examples:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.values()

print(x)

Output:

dict\_values(['Ford', 'Mustang', 1964])

TUPLES METHODS:

1.count()

Returns the number of times a specified value occurs in a tuple

Example:

thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = thistuple.count(5)

print(x)

Output:

2

2.index()

Searches the tuple for a specified value and returns the position of where it was found

Example:

thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = thistuple.index(8)

print(x)

Output:

3

SET METHODS:

1.add()

Adds an element to the set

Example:

fruits = {"apple", "banana", "cherry"}

fruits.add("orange")

print(fruits)

Output:

{"apple", "banana", "cherry”,”orange”}

2.clear()

Removes all the elements from the set

Example:

fruits = {"apple", "banana", "cherry"}

fruits.clear()

print(fruits)

Output:

{}

3.copy()

Returns a copy of the set

Example:

fruits = {"apple", "banana", "cherry"}

x = fruits.copy()

print(x)

Output:

{"apple", "banana", "cherry"}

4.difference()

Returns a set containing the difference between two or more sets

Example:

x = {"apple", "banana", "cherry"}

y = {"google", "microsoft", "apple"}

z = x.difference(y)

print(z)

Output:

{'cherry', 'banana'}

5.difference\_update()

Removes the items in this set that are also included in another, specified set

Example:

x = {"apple", "banana", "cherry"}

y = {"google", "microsoft", "apple"}

x.difference\_update(y)

print(x)

Output:

{'cherry', 'banana'}

6.discard()

Remove the specified item

Example:

fruits = {"apple", "banana", "cherry"}

fruits.discard("banana")

print(fruits)

Output:

{"apple", "cherry"}

7.intersection()

Returns a set, that is the intersection of two other sets

Example:

x = {"apple", "banana", "cherry"}

y = {"google", "microsoft", "apple"}

z = x.intersection(y)

print(z)

Output:

apple

8.intersection\_update()

Removes the items in this set that are not present in other, specified set(s)

Example:

x = {"apple", "banana", "cherry"}

y = {"google", "microsoft", "apple"}

x.intersection\_update(y)

print(x)

Output:

{‘apple’}

9.isdisjoint()

Returns whether two sets have a intersection or not

Example:

x = {"apple", "banana", "cherry"}

y = {"google", "microsoft", "facebook"}

z = x.isdisjoint(y)

print(z)

Output:

True

10.issubset()

<= Returns whether another set contains this set or not

< Returns whether all items in this set is present in other, specified set(s)

Example:

x = {"a", "b", "c"}

y = {"f", "e", "d", "c", "b", "a"}

z = x.issubset(y)

print(z)

Output:

True

11.issuperset()

>= Returns whether this set contains another set or not

> Returns whether all items in other, specified set(s) is present in this set

Example:

x = {"f", "e", "d", "c", "b", "a"}

y = {"a", "b", "c"}

z = x.issuperset(y)

print(z)

Output:

True

12.pop()

Removes an element from the set

Example:

fruits = {"apple", "banana", "cherry"}

fruits.pop()

print(fruits)

Output:

{}

13.remove()

Removes the specified element

Example:

fruits = {"apple", "banana", "cherry"}

fruits.remove("banana")

print(fruits)

Output:

{"apple", "cherry"}

14.union()

Return a set containing the union of sets

Example:

x = {"apple", "banana", "cherry"}

y = {"google", "microsoft", "apple"}

z = x.union(y)

print(z)

Output:

{"apple", "banana", "cherry","google", "microsoft"}

15.update()

Update the set with the union of this set and others

Example:

x = {"apple", "banana", "cherry"}

y = {"google", "microsoft", "apple"}

x.update(y)

print(x)

Output:

{'google', 'apple', 'cherry', 'banana', 'microsoft'}