'''

Glossary of string methods

'''

1. split() --> "this is a string".split(" ")

String to list of words, by default it splits on the basis of space.

2. join() --> " ".join(["this", "is", "a", "string"])

list of strings to a single string

3. replace() --> "this is a string".replace(" ", "\_")

Replaces all the occurrences of the first string by the second string

IMP - it creates a new string and returns it - doesn’t modify the original string

4. find() --> "this is a string".find("a")

Finds the exact sequence of char/string in the original string and returns the index of the first occurrence of it.

Returns -1 if the pattern is not present in the given string

5. count() --> "this is a string".count("a")

returns the count of the number of times the pattern is present in the given string

6. isdigit() --> "577834".isdigit()

Returns True if all the characters present in the string are digits

7. isalpha() --> "aaadfa".isalpha()

Returns True if all the characters present in the string are alphabets -- small or capitals

Spaces are not considered as alphabets. So, “aa bb cc”.isalpha() will return False

8. islower()

returns True if all the characters in the string are lowercase alphabets - else returns False

9. isupper()

returns True if all the characters in the string are uppercase alphabets - else returns False

10. isspace()

returns True if all the characters in the string are spaces- else returns False

11. lower() --> "HellO World".lower()

returns a new string with lowercase version of the given string

11. upper() --> "HellO World".upper()

returns a new string with uppercase version of the given string

12. ord("A") --> gives the ASCII value of the character

ord(‘A’) = 65

ord(‘a’) = 97

13. chr(val) --> returns the character corresponding to the ASCII value provided.

#Special mentions:

"amazing" in "you guys are amazing"

returns True if the pattern is present in the given string, else returns False

**Reverse a string without using slicing:**

str1 = “Scaler”

ans = “”

For c in str1:

ans = c+ ans

print(ans)

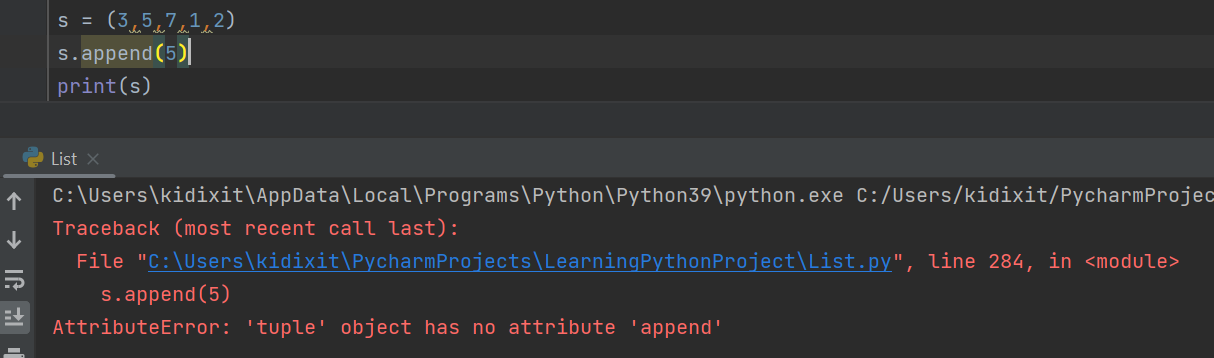
**List Operations:**

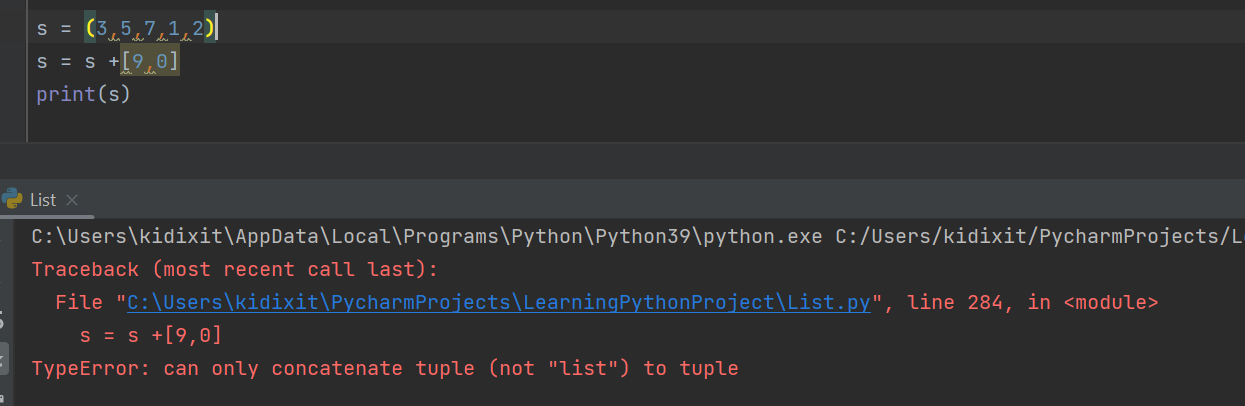
|  |  |
| --- | --- |
| **Method** | **Description** |
| [append()](https://www.w3schools.com/python/ref_list_append.asp) | Adds an element at the end of the list. Even dictionary can be appended to a list. |
| [clear()](https://www.w3schools.com/python/ref_list_clear.asp) | Removes all the elements from the list |
| [copy()](https://www.w3schools.com/python/ref_list_copy.asp) | Returns a copy of the list- if any change is made to the copy, original list  Does not get modified. |
| [count()](https://www.w3schools.com/python/ref_list_count.asp) | Returns the number of elements with the specified value |
| [extend()](https://www.w3schools.com/python/ref_list_extend.asp) | Add the elements of a list (or any iterable), to the end of the current list    Only single parameter can be passed to extend(). Error code with multiple parameters below: |
| [index()](https://www.w3schools.com/python/ref_list_index.asp) | Returns the index of the first element with the specified value. |
| [insert()](https://www.w3schools.com/python/ref_list_insert.asp) | Adds an element at the specified position |
| [pop()](https://www.w3schools.com/python/ref_list_pop.asp) | Removes the element at the specified position. By default, it’ll remove the last element if  index is not provided. Pop return the value, which is removed, so we can store  what is getting removed. |
| [remove()](https://www.w3schools.com/python/ref_list_remove.asp) | Removes the first item with the specified value. It does not return any value. \*Since the  value is removed already which was passed, so need to return it.\* |
| [reverse()](https://www.w3schools.com/python/ref_list_reverse.asp) | Reverses the order of the list. This is ‘InPlace’ reverse.    **Reversed():** |
| [sort()](https://www.w3schools.com/python/ref_list_sort.asp) | Sorts the list    Sorting cannot be done if list has values with different datatypes. |

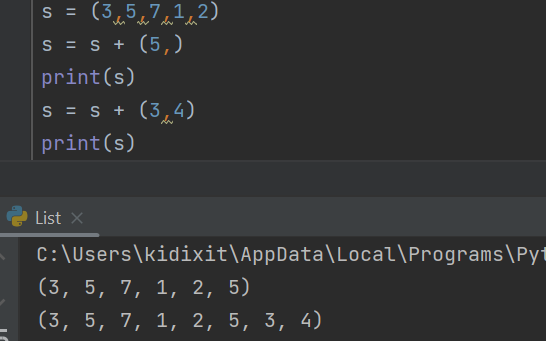
**Tuple Functions:**

|  |  |
| --- | --- |
| **Method** | **Description** |
| [count()](https://www.w3schools.com/python/ref_tuple_count.asp) | Returns the number of times a specified value occurs in a tuple |
| [index()](https://www.w3schools.com/python/ref_tuple_index.asp) | Searches the tuple for a specified value and returns the position of where it was found |

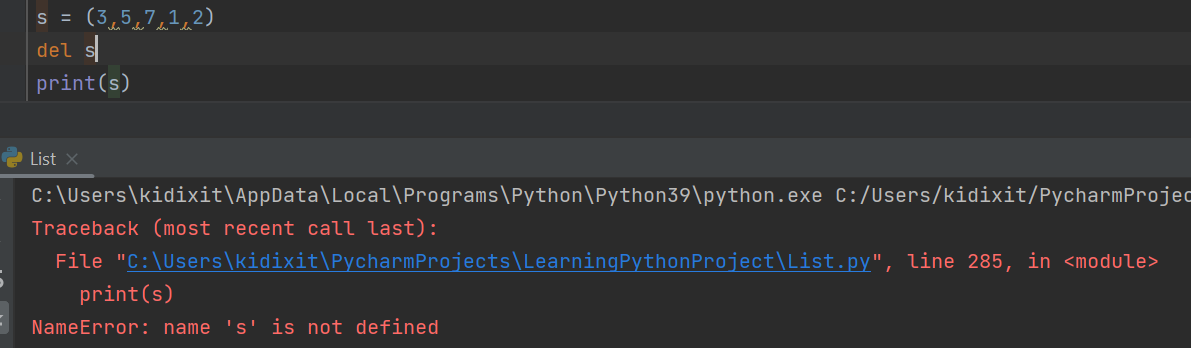
**Adding an element in Tuple: ‘+’ can be used for adding. Only tuple can be added to tuple, since tuples are immutable, we cannot modify tuple**



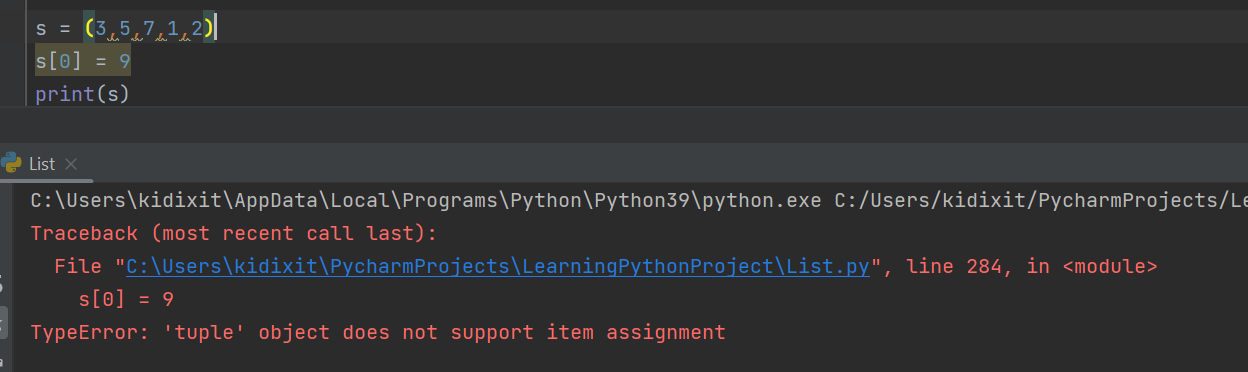




**Removing an element from a tuple: Tuples are immutable, we cannot remove any particular element. However, the entire tuple can be deleted.**



**Assigning any new value to existing index: Tuples are immutable, values cannot be modified.**



**Set Functions:**

|  |  |
| --- | --- |
| **Method** | **Description** |
| [add()](https://www.w3schools.com/python/ref_set_add.asp) | Adds an element to the set |
| [clear()](https://www.w3schools.com/python/ref_set_clear.asp) | Removes all the elements from the set |
| [copy()](https://www.w3schools.com/python/ref_set_copy.asp) | Returns a copy of the set |
| [difference()](https://www.w3schools.com/python/ref_set_difference.asp) | Returns a set containing the difference between two or more sets |
| [difference\_update()](https://www.w3schools.com/python/ref_set_difference_update.asp) | Removes the items in this set that are also included in another, specified set |
| [discard()](https://www.w3schools.com/python/ref_set_discard.asp) | Remove the specified item |
| [intersection()](https://www.w3schools.com/python/ref_set_intersection.asp) | Returns a set, that is the intersection of two or more sets |
| [intersection\_update()](https://www.w3schools.com/python/ref_set_intersection_update.asp) | Removes the items in this set that are not present in other, specified set(s) |
| [isdisjoint()](https://www.w3schools.com/python/ref_set_isdisjoint.asp) | Returns whether two sets have a intersection or not |
| [issubset()](https://www.w3schools.com/python/ref_set_issubset.asp) | Returns whether another set contains this set or not |
| [issuperset()](https://www.w3schools.com/python/ref_set_issuperset.asp) | Returns whether this set contains another set or not |
| [pop()](https://www.w3schools.com/python/ref_set_pop.asp) | Removes an element from the set |
| [remove()](https://www.w3schools.com/python/ref_set_remove.asp) | Removes the specified element |
| [symmetric\_difference()](https://www.w3schools.com/python/ref_set_symmetric_difference.asp) | Returns a set with the symmetric differences of two sets |
| [symmetric\_difference\_update()](https://www.w3schools.com/python/ref_set_symmetric_difference_update.asp) | inserts the symmetric differences from this set and another |
| [union()](https://www.w3schools.com/python/ref_set_union.asp) | Return a set containing the union of sets |
| [update()](https://www.w3schools.com/python/ref_set_update.asp) | Update the set with another set, or any other iterable |