WRITE 20 method for list, dictionary and tuples using examples to explain all methods

1. Method one: LIST

In [1]:

```
#1. Append(element): This add an element to the end of the list
#Example

my_list = [1, 2, 3]
my_list.append(4)
print(my_list)  # Output: [1, 2, 3, 4]
```

```
[1, 2, 3, 4]
```

In [3]:

```
#2 Extend(iterables) : This append element from an iterable to the end of the list
#Example

my_list = [1, 2, 3]
my_list.extend([4, 5,6])
print(my_list)  # Output: [1, 2, 3, 4, 5]
```

```
[1, 2, 3, 4, 5, 6]
```

In [4]:

```
#3 insert(index,element): This insert index at the specified element in the list

my_list = [1, 2, 3]
my_list.insert(1, 4)
print(my_list)  # Output: [1, 4, 2, 3]
```

```
[1, 4, 2, 3]
```

In [5]:

```
# 4 remove(element): Removes the first occurrence of the specified element from the list.

my_list = [1, 2, 3, 2]
my_list.remove(2)
print(my_list)  # Output: [1, 3, 2]
```

```
[1, 3, 2]
```

In [6]:

```
#5 pop(index):  Removes and returns the element at the specified index.
#If no index is provided, it removes and returns the last element.

my_list = [1, 2, 3]
popped_element = my_list.pop(1)
print(popped_element)  # Output: 2
print(my_list)         # Output: [1, 3]
```

```
2
[1, 3]
```

In [7]:

```

# 6 index(element, start, end): Returns the index of the first occurrence of the specified element in the list.

my_list = [1, 2, 3, 2]
index = my_list.index(2)
print(index)
```

```
1
```

In [11]:

```
# 7 count(element):  Returns the number of occurrences of the specified element in the list.

my_list = [1, 2, 3, 2]
count = my_list.count(2)
print(count)
```

```
2
```

In [12]:

```
# 8 sort(key=None, reverse=False): Sorts the elements of the list in ascending order.
#The reverse parameter can be set to True for descending order.

my_list = [3, 1, 4, 1, 5, 9, 2]
my_list.sort()
print(my_list)
```

```
[1, 1, 2, 3, 4, 5, 9]
```

In [13]:

```
# 9 reverse():  Reverses the order of elements in the list.

my_list = [1, 2, 3, 4]
my_list.reverse()
print(my_list)  # Output: [4, 3, 2, 1]
```

```
[4, 3, 2, 1]
```

In [14]:

```
# 10 copy():  Returns a shallow copy of the list.

my_list = [1, 2, 3]
copied_list = my_list.copy()
print(copied_list)
```

```
[1, 2, 3]
```

In [15]:

```
# 11 clear():  Removes all elements from the list.

my_list = [1, 2, 3]
my_list.clear()
print(my_list)
```

```
[]
```

In [16]:

```
#12 : count(element):  Returns the number of occurrences of the specified element in the list.

my_list = [1, 2, 3, 2]
count = my_list.count(2)
print(count)
```

```
2
```

In [17]:

```
#13 : index(element, start, end):  Returns the index of the first occurrence of the specified element in the list within

my_list = [1, 2, 3, 2]
index = my_list.index(2, 1, 3)
print(index)
```

```
1
```

In [18]:

```
#14 : max(element):  Calculates the maximum of all the elements of the List

my_list = [1, 2, 7,9, 5]
index = max(my_list)
print(index)
```

```
9
```

In [19]:

```
#15 : min(element):  Calculates the minimum of all the elements of the List

my_list = [1, 2, 7,9, 5]
index = min(my_list)
print(index)
```

```
1
```

In [23]:

```
#16 : sort(element):  Calculates the sort or arrange all the elements of the List

my_list = [1, 2, 7,9, 5,6,0,3,4]
my_list.sort()
print(my_list)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 9]
```

In [ ]:

Method two: Dictionaries

In [24]:

```
#1 get(key, default):  Returns the value associated with the specified key.
#If the key is not present, it returns the default value.

my_dict = {'a': 1, 'b': 2, 'c': 3}
value = my_dict.get('b', 0)
print(value)
```

```
2
```

In [25]:

```
# 2 keys():  Returns a list of all keys in the dictionary.

my_dict = {'a': 1, 'b': 2, 'c': 3}
keys = my_dict.keys()
print(keys)
```

```
dict_keys(['a', 'b', 'c'])
```

In [26]:

```
# 3 values():  Returns a list of all values in the dictionary.

my_dict = {'a': 1, 'b': 2, 'c': 3}
values = my_dict.values()
print(values)
```

```
dict_values([1, 2, 3])
```

In [27]:

```
#4 items(): Returns a list of key-value pairs as tuples.

my_dict = {'a': 1, 'b': 2, 'c': 3}
items = my_dict.items()
print(items)
```

```
dict_items([('a', 1), ('b', 2), ('c', 3)])
```

In [28]:

```
# 5 pop(key, default): Removes the key and returns its associated value.
#If the key is not present, it returns the default value.

my_dict = {'a': 1, 'b': 2, 'c': 3}
value = my_dict.pop('b', 0)
print(value)  # Output: 2
```

```
2
```

In [29]:

```
# 6 popitem(): Removes and returns the last key-value pair as a tuple.


my_dict = {'a': 1, 'b': 2, 'c': 3}
item = my_dict.popitem()
print(item)
```

```
('c', 3)
```

In [30]:

```
#7 update(dictionary): Updates the dictionary with elements from another dictionary or iterable.

my_dict = {'a': 1, 'b': 2}
new_dict = {'b': 3, 'c': 4}
my_dict.update(new_dict)
print(my_dict)  # Output: {'a': 1, 'b': 3, 'c': 4}
```

```
{'a': 1, 'b': 3, 'c': 4}
```

In [31]:

```
#8 clear(): Removes all elements from the dictionary.

my_dict = {'a': 1, 'b': 2, 'c': 3}
my_dict.clear()
print(my_dict)  # Output: {}
```

```
{}
```

In [32]:

```
#9 copy(): Returns a shallow copy of the dictionary.

my_dict = {'a': 1, 'b': 2, 'c': 3}
copied_dict = my_dict.copy()
print(copied_dict)  # Output: {'a': 1, 'b': 2, 'c': 3}
```

```
{'a': 1, 'b': 2, 'c': 3}
```

In [33]:

```
# 10 fromkeys(iterable, value): Creates a new dictionary with keys from the iterable and values set to a specified value

keys = ['a', 'b', 'c']
value = 0
new_dict = dict.fromkeys(keys, value)
print(new_dict)  # Output: {'a': 0, 'b': 0, 'c': 0}
```

```
{'a': 0, 'b': 0, 'c': 0}
```

In [34]:

```
#11 setdefault(key, default): Returns the value for the specified key.
#If the key is not present, inserts the key with the specified default value

my_dict = {'a': 1, 'b': 2}
value = my_dict.setdefault('c', 0)
print(value)  # Output: 0
print(my_dict)  # Output: {'a': 1, 'b': 2, 'c': 0}
```

```
0
{'a': 1, 'b': 2, 'c': 0}
```

In [35]:

```
#12 popitem(): Removes and returns a key-value pair as a tuple. Useful for removing an arbitrary item.

my_dict = {'a': 1, 'b': 2, 'c': 3}
item = my_dict.popitem()
print(item)  # Output: ('c', 3)
```

```
('c', 3)
```

In [36]:

```
#13 update(dictionary): Updates the dictionary with elements from another dictionary or iterable.

my_dict = {'a': 1, 'b': 2}
new_dict = {'b': 3, 'c': 4}
my_dict.update(new_dict)
print(my_dict)  # Output: {'a': 1, 'b': 3, 'c': 4}
```

```
{'a': 1, 'b': 3, 'c': 4}
```

In [37]:

```
#14: clear(): Removes all elements from the dictionary

my_dict = {'a': 1, 'b': 2, 'c': 3}
my_dict.clear()
print(my_dict)
```

```
{}
```

In [38]:

```
#15  copy(): Returns a shallow copy of the dictionary.

my_dict = {'a': 1, 'b': 2, 'c': 3}
copied_dict = my_dict.copy()
print(copied_dict)
```

```
{'a': 1, 'b': 2, 'c': 3}
```

In [ ]:

```
#16 fromkeys(iterable, value): Creates a new dictionary with keys from the iterable and values set to a specified value.

keys = ['a', 'b', 'c']
value = 0
new_dict = dict.fromkeys(keys, value)
print(new_dict)
```

In [39]:

```
#17 setdefault(key, default): Returns the value for the specified key. If the key is not present, inserts the key with th

my_dict = {'a': 1, 'b': 2}
value = my_dict.setdefault('c', 0)
print(value)
print(my_dict)
```

```
0
{'a': 1, 'b': 2, 'c': 0}
```

In [40]:

```
# 18: items():Returns a list of key-value pairs as tuples.

my_dict = {'a': 1, 'b': 2, 'c': 3}
items = my_dict.items()
print(items)
```

```
dict_items([('a', 1), ('b', 2), ('c', 3)])
```

In [ ]:

Method Three: Tuples

In [41]:

```
#1 count(element): Returns the number of occurrences of the specified element in the tuple.

my_tuple = (1, 2, 3, 2, 4, 2, 5)
count = my_tuple.count(2)
print(count)
```

```
3
```

In [42]:

```
#2 index(element): Returns the index of the first occurrence of the specified element in the tuple.

my_tuple = (10, 20, 30, 20, 40)
index = my_tuple.index(20)
print(index)
```

```
1
```

In [43]:

```
#3 len(): Returns the number of elements in the tuple.

my_tuple = (1, 2, 3, 4, 5)
length = len(my_tuple)
print(length)  # Output: 5
```

```
5
```

In [44]:

```
#4 max(): Returns the maximum element of the tuple

my_tuple = (10, 5, 20, 15)
maximum = max(my_tuple)
print(maximum)
```

```
20
```

In [47]:

```
#5 min(): Returns the minimum element of the tuple.

my_tuple = (10, 5, 20, 15)
minimum = min(my_tuple)
print(minimum)
```

```
5
```

In [48]:

```
#6 sorted(): Returns a new sorted list from the elements of the tuple.

my_tuple = (3, 1, 4, 1, 5, 9, 2)
sorted_list = sorted(my_tuple)
print(sorted_list)
```

```
[1, 1, 2, 3, 4, 5, 9]
```

In [49]:

```
#7 sum(): Returns the sum of all elements in the tuple.

my_tuple = (3, 1, 4, 1, 5, 9, 2)
my_tuple = (1, 2, 3, 4, 5)
total = sum(my_tuple)
print(total)
```

```
15
```

In [50]:

```
#8 any(): Returns True if at least one element of the tuple is True, otherwise False

my_tuple = (False, False, True, False)
any_true = any(my_tuple)
print(any_true)
```

```
True
```

In [51]:

```
#9 all(): Returns True if all elements of the tuple are True, otherwise False.

my_tuple = (True, True, True, False)
all_true = all(my_tuple)
print(all_true)
```

```
False
```

In [52]:

```
#10: reversed(): Returns a reversed iterator of the tuple's elements.

my_tuple = (1, 2, 3, 4, 5)
reversed_tuple = tuple(reversed(my_tuple))
print(reversed_tuple)
```

```
(5, 4, 3, 2, 1)
```

In [53]:

```
#11 len(): Returns the number of elements in the tuple.

my_tuple = (1, 2, 3, 4, 5)
length = len(my_tuple)
print(length)
```

```
5
```

In [54]:

```
#12: max(): Returns the maximum element of the tuple.

my_tuple = (10, 5, 20, 15)
maximum = max(my_tuple)
print(maximum)
```

```
20
```

In [55]:

```
#13: min(): Returns the minimum element of the tuple.

my_tuple = (10, 5, 20, 15)
minimum = min(my_tuple)
print(minimum)
```

```
5
```

In [56]:

```
#14 sorted(): Returns a new sorted list from the elements of the tuple.

my_tuple = (3, 1, 4, 1, 5, 9, 2)
sorted_list = sorted(my_tuple)
print(sorted_list)
```

```
[1, 1, 2, 3, 4, 5, 9]
```

In [58]:

```
#15 reversed(): Returns a reversed iterator of the tuple's elements.

my_tuple = (1, 2, 3, 4, 5)
reversed_tuple = tuple(reversed(my_tuple))
print(reversed_tuple)
```

```
(5, 4, 3, 2, 1)
```

In [57]:

```
#16 index(element, start, end): Returns the index of the first occurrence of the specified element in
#the tuple within the given start and end indices.

my_tuple = (1, 2, 3, 2, 4, 2, 5)
index = my_tuple.index(2, 2, 6)
print(index)
```

```
3
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: