PYTHON LIST METHODS

Here are some of the python list methods below.

**1.Append()**

* **Description:** This method is use to add an item to the end of the list.
* **Syntax:** list.append(item)
  + - * **Item:** The element to be added to the list.
* **Return Type:** None
* **Example:** my\_list:int= [1 , 2 , 3 , 4]

my\_list.append(5)

print(my\_list) *#Output = [1 , 2 , 3 , 4 , 5]*

**2.Extend()**

* **Description:** This method is use to extend the list by appending it from some iterable(like another list) to the end.
* **Syntax:** list.extend(iterable)

**Iterable:** The iterable(list , tuple , set , etc) whose element is added to the lsit

* **Return Type:** None
* **Example:** **:** my\_list:int = [1 , 2 , 3 , 4]

my\_list.extend([4 , 5])

print(my\_list) *#Output = [1 , 2 , 3 , 4 , 5]*

**3.Insert()**

* + - * **Description:** This method is used to insert an item to the given position of the lsit.
      * **Syntax:** list.insert(index , item)
  + **Index:** The position where the item should be inserted.
  + **Item:** The element to be inserted.
    - **Return Type:** None
* **Example:** my\_list:int = [1 , 2 , 3 , 5]

my\_list.insert(3 , 4)

print(my\_list) *#Output = [1 , 2 , 3 , 4 , 5]*

**4**.**Remove()**

* **Description:** This method is use to remove the first occurrence of the item of the specified item from the list.
* **Syntax:** list.remove( item)
  + **Item:** The element to be removed.
* **Return:** None
* **Example:** fruits:str = [“Apple” , “Banana” , “Grapes”]

fruits.remove(“Banana”)

print(fruits) *#Output = [“Apple” , “Grapes”]*

**5.Pop()**

* + - **Description:**This method is use to remove and returns the item at the specified position in the list.If no item is specified it will remove and return the last element of the list.
    - **Syntax:**list.pop(index)
  + **Index:**Optional.The position of the item to remove and return.Default is -1 (last element).
* **Return:**The removed item
* **Example:** fruits:str = [“Apple” , “Banana” , “Grapes”]

item: str = fruits.pop(0)

print(item) *#Output = [“Apple”]*

print(fruits) *#Output = [“Apple” , “Grapes”]*

**6**.**Clear()**

* **Description:** This method is used to clear all elements from the list leaving it empty.
* **Syntax:**list.clear()
* **Return Type:**None
* **Example:** fruits:str = [“Apple” , “Banana” , “Grapes”]

fruits.clear()

print(fruits) *#Output = []*

**7**.**Index()**

* **Description:** This method is used to return the index of the first occurrence of a specified element of the list.
* **Syntax:**list.index(item , start=0 , end=len(list)
  + - * **Item:** The element to search for.
      * **Start:** Optional.The position to start the search.Default is 0.
      * **End:** Optional.The position to end the list.Default is len(list).
* **Return Type:** int(the index of the item)
* **Example:** fruits:str = [“Apple” , “Banana” , “Grapes”]

index = fruits.index(“Banana”)

print(index) *#Output = 1*

**8.Count()**

* **Description:** Return the number of times of specified item appear in the list.
* **Syntax:** list.count( item)
  + **Item:** The element to be count.
* **Return:** int(the count of the element)
* **Example:** fruits:str = [“Apple” , “Apple” , “Banana” , “Grapes”]

count = my\_list.count(“Apple”)

print(count) *#Output = 2*

**9.Sort()**

* **Description:** Sort the items of the liat by default in Ascending order.It can also sort in Descending order and allow custom sorting criteria.
* **Syntax:**list.sort( key=None , reverse= False)
  + - * **Key:** Optional.A function to execute for each list item brfore sorting.Default is none.
      * **Reverse:** If ‘True’, the list elements are sorted as if each comparison was reversed. Default is False.
* **Return Type:** None
* **Example:** my\_list:int = [3 , 4 , 5 , 2 , 1 ]

my\_list.sort()

print(my\_list) *#Output = [1 , 2 . 3 . 4 . 5]*

* Sort in Descending order.

my\_list.sort(reverse=True)

print(my\_list) *#Output = [5 , 4 , 3 , 2 , 1]*

**10.Reverse()**

* **Description:**This method is used to reverse the elements of the lis
* **Syntax:** list.reverse()
* **Return Type:** None
* **Example:** my\_list:int = [1 , 2 , 3 , 4 ]

my\_list.reverse()

print(my\_list) *#Output = [1, 2 , 3 , 4]*

**11.Copy()**

* **Description:** This method returns the shallow copy of the list.
* **Syntax:**list.copy()
* **Return Type:** A new list that is the shallow copy of original list.
* **Example:** fruits:str = [“Apple” , “Banana” , “Grapes”]

new\_list = fruits.copy()

print(new\_list) *#Output = [“Apple” , “Banana” , “Grapes”]*

**12.Len()**

* **Description:** This is not a method but a function that returns the number of item in the list.
* **Syntax:**len(list)
* **Return Type:** int
* **Example:** fruits:str = [“Apple” , “Banana” , “Grapes”]

len(fruits) *#Output = 3*

## ❤**(´｡•◡•｡`)♡ ThankYo u ♡(´｡•◡•｡`) ❤**