

Untitled-1

```
1  # %% [markdown]
2  # MUHAMMAD ABDULLAH
3  #
4  # B22F0577AI054
5  #
6  # SUBMITTED TO: MAM ANILA HABIB
7  #
8  # PROGRAMMING FOR AI LAB
9  #
10 # AI LAB:12
11
12 # %% [markdown]
13 # #Python Lists
14
15 # %% [markdown]
16 # Lists are ordered collection of data items. They store multiple items in a single variable.
17 # List items are separated by commas and enclosed within square brackets []. Lists are changeable
18 # meaning we can alter them after creation.
19
20 # %%
21 lst1 = [1,2,2,3,5,4,6]
22 lst2 = ["Red", "Green", "Blue"]
23
24 print(lst1)
25 print(lst2)
26
27 # %%
28 details = ["Abdullah", 18, "FYBScIT", 9.8]
29 print(details)
30
31 # %% [markdown]
32 # List Indexes
33 # Each item/element in a list has its own unique index. This index can be used to access any
34 # particular item from the list. The first item has index [0], second item has index [1], third
35 # item has index [2] and so on.
36
37 # %%
38 colors = ["Red", "Green", "Blue", "Yellow", "Green"]
39 #         [0]      [1]      [2]      [3]      [4]
40
41 # %% [markdown]
42 # Accessing list items:
43
44 # %% [markdown]
45 # Positive Indexing:
46 # As we have seen that list items have index, as such we can access items using these indexes.
47
48 # %%
49 colors = ["Red", "Green", "Blue", "Yellow", "Green"]
50 #         [0]      [1]      [2]      [3]      [4]
51 print(colors[2])
52 print(colors[4])
53 print(colors[0])
54
55 # %% [markdown]
56 # Negative Indexing:
```

```
52
53 # %% [markdown]
54 # Similar to positive indexing, negative indexing is also used to access items, but from the
    end of the list. The last item has index [-1], second last item has index [-2], third last item
    has index [-3] and so on
55
56 # %%
57 colors = ["Red", "Green", "Blue", "Yellow", "Green"]
58
59 #[-5] [-4] [-3] [-2] [-1]
60 print(colors[-1])
61 print(colors[-3])
62 print(colors[-4])
63
64 # %% [markdown]
65 # Check for item:
66 # We can check if a given item is present in the list. This is done using the in keyword.
67
68 # %%
69 colors = ["Red", "Green", "Blue", "Yellow", "Green"]
70 if "Yellow" in colors:
71     print("Yellow is present.")
72 else:
73     print("Yellow is absent.")
74
75 # %% [markdown]
76 # Add List Items
77
78 # %% [markdown]
79 # There are three methods to add items to list: append(), insert() and extend()
80 #
81 # append(): This method appends items to the end of the existing list.
82
83 # %%
84 colors = ["Red", "Green", "Blue", "Yellow", "Green"]
85 if "Orange" in colors:
86     print("Orange is present.")
87 else:
88     print("Orange is absent.")
89
90 # %% [markdown]
91 # What if you want to insert an item in the middle of the list? At a specific index?
92
93 # %% [markdown]
94 # insert():
95 # This method inserts an item at the given index. User has to specify index and the item to be
    inserted within the insert() method
96
97 # %%
98 animals = ["cat", "dog", "bat", "mouse", "pig", "horse", "donkey", "goat", "cow"]
99 print(animals[3:7]) #using positive indexes
100 print(animals[-7:-2]) #using negative indexes
101
102 # %% [markdown]
103 # What if you want to append an entire list or any other collection (set, tuple, dictionary) to
    the existing list?
104
```

```
105 # %% [markdown]
106 # extend():
107 # This method adds an entire list or any other collection datatype (set, tuple, dictionary) to
    the existing list.
108
109 # %%
110 colors = ["voilet", "indigo", "blue"]
111 colors.append("green")
112 print(colors)
113
114 # %%
115 colors = ["voilet", "indigo", "blue"]
116 #           [0]           [1]           [2]
117
118 colors.insert(1, "green") #inserts item at index 1
119 # updated list: colors = ["voilet", "green", "indigo", "blue"]
120 #           indexs           [0]           [1]           [2]           [3]
121
122 print(colors)
123
124 # %% [markdown]
125 # concatenate two lists:
126
127 # %% [markdown]
128 # you can simply concatenate two list to join two lists.
129
130 # %%
131 #add a list to a list
132 colors = ["voilet", "indigo", "blue"]
133 rainbow = ["green", "yellow", "orange", "red"]
134 colors.extend(rainbow)
135 print(colors)
136
137 # %% [markdown]
138 # Remove List Items
139 #
140 # There are various methods to remove items from the list: pop(), remove(), del(), clear()
141 #
142 # pop(): This method removes the last item of the list if no index is provided. If an index is
    provided, then it removes item at that specified index.
143
144 # %%
145 #add a tuple to a list
146 cars = ["Hyundai", "Tata", "Mahindra"]
147 cars2 = ("Mercedes", "Volkswagen", "BMW")
148 cars.extend(cars2)
149 print(cars)
150
151 # %%
152 colors = ["voilet", "indigo", "blue", "green"]
153 colors2 = ["yellow", "orange", "red"]
154 print(colors + colors2)
155
156 # %% [markdown]
157 # remove():
158 # This method removes specific item from the list.
159
```

```
160 # %%
161 colors = ["Red", "Green", "Blue", "Yellow", "Green"]
162 colors.pop()          #removes the last item of the list
163 print(colors)
164
165 # %% [markdown]
166 # del:
167 # del is not a method, rather it is a keyword which deletes item at specific from the list, or
168 # deletes the list entirely.
169
170 # %%
171 colors = ["Red", "Green", "Blue", "Yellow", "Green"]
172 colors.pop(1)          #removes item at index 1
173 print(colors)
174
175 # %% [markdown]
176 # What if we don't want to delete the entire list, we just want to delete all items within that
177 # list?
178
179 # %% [markdown]
180 # clear():
181 # This method clears all items in the list and prints an empty list.
182
183 # %%
184 colors = ["voilet", "indigo", "blue", "green", "yellow"]
185 colors.remove("blue")
186 print(colors)
187
188 # %% [markdown]
189 # Change List Items
190
191 # %% [markdown]
192 # Changing items from list is easier, you just have to specify the index where you want to
193 # replace the item with existing item.
194
195 # %%
196 colors = ["voilet", "indigo", "blue", "green", "yellow"]
197 del colors[3]
198 print(colors)
199
200 # %%
201 colors = ["voilet", "indigo", "blue", "green", "yellow"]
202 colors.clear()
203 print(colors)
204
205 # %% [markdown]
206 # List Comprehension
207
208 # %% [markdown]
209 # List comprehensions are used for creating new lists from other iterables like lists, tuples,
210 # dictionaries, sets, and even in arrays and strings.
211
212 #
213 # Syntax:
214 #
215 # List = [expression(item) for item in iterable if condition]
216 #
217 # expression: it is the item which is being iterated.
```

```
213 #
214 # iterable: it can be list, tuples, dictionaries, sets, and even in arrays and strings.
215 #
216 # condition: condition checks if the item should be added to the new list or not.
217
218 # %% [markdown]
219 # Example 1: accepts items with the small letter "o" in the new list
220
221 # %%
222 names = ["Harry", "Sarah", "Nadia", "Oleg", "Steve"]
223 names[2] = "Millie"
224 print(names)
225
226 # %%
227 names = ["Harry", "Sarah", "Nadia", "Oleg", "Steve"]
228 names[2:4] = ["juan", "Anastasia"]
229 print(names)
230
231 # %% [markdown]
232 # List Methods
233
234 # %% [markdown]
235 # We have discussed methods like append(), clear(), extend(), insert(), pop(), remove() before.
236 # Now we will learn about some more list methods:
237
238 # %% [markdown]
239 # sort(): This method sorts the list in ascending order
240
241 # %%
242 colors = ["voilet", "indigo", "blue", "green"]
243 colors.sort()
244 print(colors)
245
246 num = [4,2,5,3,6,1,2,1,2,8,9,7]
247 num.sort()
248 print(num)
249
250 # %%
251 colors = ["voilet", "indigo", "blue", "green"]
252 colors.sort(reverse=True)
253 print(colors)
254
255 num = [4,2,5,3,6,1,2,1,2,8,9,7]
256 num.sort(reverse=True)
257 print(num)
258
259 # %% [markdown]
260 # reverse(): This method reverses the order of the list.
261
262 # %%
263 colors = ["voilet", "indigo", "blue", "green"]
264 colors.reverse()
265 print(colors)
266
267 num = [4,2,5,3,6,1,2,1,2,8,9,7]
268 num.reverse()
```

```
268 print(num)
269
270 # %% [markdown]
271 # copy(): Returns copy of the list. This can be done to perform operations on the list without
    modifying the original list.
272
273 # %%
274 colors = ["voilet", "green", "indigo", "blue"]
275 newlist = colors.copy()
276 print(colors)
277 print(newlist)
```