listsss

April 21, 2024

list is the most common datatype in python list is mutable datatype —> can be changed after creation

[1]:
```
"1- define a list "
l = []
myl = list()
```

[2]:
```
"list can hold different data from different datatypes"
myl = [34,4, "iti", 344.32, "ahmed", True, ['kubernates', 'GCP', "ansible"],␣
 ↪"iti"]
```

[5]:
```
""" each element is assigned to an index --> index start from 0 """
print(myl[4])
print(myl[6][2])
```

```
ahmed
ansible
```

[6]:
```
"get len of list"
print(len(myl))
```

```
8
```

[7]:
```
"count element occurence"
print(myl.count("iti"))
```

```
2
```

[8]:
```
"""get index of element """
print(myl.index("iti"))
```

```
2
```

[9]:
```
"""slicing the list"""
print(myl[3:5])
```

```
[344.32, 'ahmed']
```

[11]:
```
print(myl[::-1])
```

```
['iti', ['kubernates', 'GCP', 'ansible'], True, 'ahmed', 344.32, 'iti', 4, 34]
```

[12]:
```
"""list is mutable datatype --> can be changed after creation"""
"""modify existing element"""
print(myl)
myl[1] = "updated"
print(myl)
```

```
[34, 4, 'iti', 344.32, 'ahmed', True, ['kubernates', 'GCP', 'ansible'], 'iti']
[34, 'updated', 'iti', 344.32, 'ahmed', True, ['kubernates', 'GCP', 'ansible'],
'iti']
```

[13]:
```
myl[100]='new value'
print(myl)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[13], line 1
----> 1 myl[100]='new value'
      2 print(myl)

IndexError: list assignment index out of range
```

“ “ ” add element to the list “ “ ”

[14]:
```
print(myl)
myl.append("new element")  # append element at the end of the list.
print(myl)
```

```
[34, 'updated', 'iti', 344.32, 'ahmed', True, ['kubernates', 'GCP', 'ansible'],
'iti']
[34, 'updated', 'iti', 344.32, 'ahmed', True, ['kubernates', 'GCP', 'ansible'],
'iti', 'new element']
```

[15]:
```
"insert in certain position"
myl.insert(2, "inserted element")
print(myl)
```

```
[34, 'updated', 'inserted element', 'iti', 344.32, 'ahmed', True, ['kubernates',
'GCP', 'ansible'], 'iti', 'new element']
```

[17]:
```
myl.insert(100 , "anyvalue")
print(myl)
```

```
[34, 'updated', 'inserted element', 'iti', 344.32, 'ahmed', True, ['kubernates',
'GCP', 'ansible'], 'iti', 'new element', 'anyvalue', 'anyvalue']
```

Remove Element from list

[18]:
```
""" remove element from list """
print(myl)
```

```
popped_value=myl.pop()
```

```
[34, 'updated', 'inserted element', 'iti', 344.32, 'ahmed', True, ['kubernates',
'GCP', 'ansible'], 'iti', 'new element', 'anyvalue', 'anyvalue']
```

[19]:
```
print(popped_value)
print(myl)
```

```
anyvalue
[34, 'updated', 'inserted element', 'iti', 344.32, 'ahmed', True, ['kubernates',
'GCP', 'ansible'], 'iti', 'new element', 'anyvalue']
```

[20]:
```
"""remove element at index --> """
popped_index_value=myl.pop(4)
```

[21]:
```
print(popped_index_value)
print(myl)
```

```
344.32
[34, 'updated', 'inserted element', 'iti', 'ahmed', True, ['kubernates', 'GCP',
'ansible'], 'iti', 'new element', 'anyvalue']
```

[22]:
```
"""remove element from the list"""
myl.remove("iti") # REMOVE FIRST OCCURRENCE OF THE ELEMENT FROM THE LIST.
print(myl)
```

```
[34, 'updated', 'inserted element', 'ahmed', True, ['kubernates', 'GCP',
'ansible'], 'iti', 'new element', 'anyvalue']
```

[ ]:
```
""" loop over the list """
```

[23]:
```
for element in myl:
    print(f"element = {element}")
```

```
element = 34
element = updated
element = inserted element
element = ahmed
element = True
element = ['kubernates', 'GCP', 'ansible']
element = iti
element = new element
element = anyvalue
```

[24]:
```
"""check element in list or not """
print("iti" in myl)
```

```
True
```

[26]:
```
"""---> list concat"""
l1 = ["python", "mongo", "javascript", "sql"]
l2= ["jenkins", "GCP", "Ansible", "Kubernates", "Docker"]
```

[27]:
```
lst_of_courses = l1  + l2 # NEW LIST
print(lst_of_courses)
```

```
['python', 'mongo', 'javascript', 'sql', 'jenkins', 'GCP', 'Ansible',
'Kubernates', 'Docker']
```

[28]:
```
"""extend a list """
l1.extend(l2)
print(l1)
print(l2)
```

```
['python', 'mongo', 'javascript', 'sql', 'jenkins', 'GCP', 'Ansible',
'Kubernates', 'Docker']
['jenkins', 'GCP', 'Ansible', 'Kubernates', 'Docker']
```

Sort list –> comparing —> list –> elements from the same type

[29]:
```
print(lst_of_courses)
lst_of_courses.sort()   # SORT LIST ELEMENTS IN THE SAME LIST
print(lst_of_courses)
```

```
['python', 'mongo', 'javascript', 'sql', 'jenkins', 'GCP', 'Ansible',
'Kubernates', 'Docker']
['Ansible', 'Docker', 'GCP', 'Kubernates', 'javascript', 'jenkins', 'mongo',
'python', 'sql']
```

[30]:
```
lst_of_courses.sort(reverse=True)
print(lst_of_courses)
```

```
['sql', 'python', 'mongo', 'jenkins', 'javascript', 'Kubernates', 'GCP',
'Docker', 'Ansible']
```

“ “ ” reverse the list ? “ “ ”

[31]:
```
myl.reverse() # DOESN'T ENFORCE LIST ITEMS TO BE FROM THE SAME TYPE.
print(myl)
```

```
['anyvalue', 'new element', 'iti', ['kubernates', 'GCP', 'ansible'], True,
'ahmed', 'inserted element', 'updated', 34]
```

Casting string to a list.

[32]:
```
name = 'noha'
name = list(name)
print(name)
```

```
['n', 'o', 'h', 'a']
```

[33]:
```
"""split string to a list """
message = 'we support Ghaza'
newl = message.split(" ") # LIST
print(newl)
```

```
['we', 'support', 'Ghaza']
```

min-max

[34]:
```
print(min([3,454,23,323,12,21]))
print(max([3,454,23,323,12,21]))
```

```
3
454
```

for loop and index

[36]:
```
myl.append("iti")
index  = 0
for item in myl:
    print(f"{item},{myl.index(item)}, {index}")
    index += 1
```

```
anyvalue,0, 0
new element,1, 1
iti,2, 2
['kubernates', 'GCP', 'ansible'],3, 3
True,4, 4
ahmed,5, 5
inserted element,6, 6
updated,7, 7
34,8, 8
iti,2, 9
iti,2, 10
```

enum

[39]:
```
myl_enum = enumerate(myl)
print(myl_enum)
# print(list(myl_enum))
```

```
<enumerate object at 0x753406a30270>
```

use it with loops

[40]:
```
for i , value in myl_enum:
    print(f"i={i}, item = {value}")
```

```
i=0, item = anyvalue
i=1, item = new element
i=2, item = iti
i=3, item = ['kubernates', 'GCP', 'ansible']
```

```
i=4, item = True
i=5, item = ahmed
i=6, item = inserted element
i=7, item = updated
i=8, item = 34
i=9, item = iti
i=10, item = iti
```

[ ]: