Sequence Types - Lists

January 24, 2023

1 Lists in Python

- Ordered Collection of Elements (of same type or of different types)
- Every list element is associated with an index value
- Indexes start from 0

[9]:

```
lst = [10, 20, 30, 40]
#      0   1   2   3
print(lst)
print(type(lst))
```

```
[10, 20, 30, 40]
<class 'list'>
```

[11]:

```
lst = [10, 20, 30, 40]
#      0   1   2   3
print(lst[1])
print(lst[3])
```

```
20
40
```

[12]:

```
lst = [10, 20, 30, 40]
#      0   1   2   3
print(lst[4])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3008\320047938.py in <cell line: 3>()
      1 lst = [10, 20, 30, 40]
      2 #      0   1   2   3
----> 3 print(lst[4])

IndexError: list index out of range
```

1.1 Creating a list from another iterable using list()

- list() –> Pass another iterable object as an argument

[14]:

```
lst = list(range(1, 21))
print(lst)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

[15]:

```
string = 'this is python'
lst = list(string)
print(lst)
```

```
['t', 'h', 'i', 's', ' ', 'i', 's', ' ', 'p', 'y', 't', 'h', 'o', 'n']
```

[17]:

```
a = map(int, input().split()) # return a map object
for i in a:
    print(i)
```

```
10 20 30
10
20
30
```

[19]:

```
# Reading a list of integers from the user
a = list(map(int, input().split()))
print(a)
```

```
10 20 30 40 50 60 70 80 90
[10, 20, 30, 40, 50, 60, 70, 80, 90]
```

[ ]:

```
5
10 20 30 40 50 # list(map(int, input().split()))
```

[18]:

```
n = 12345
lst = list(n)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3008\4084999993.py in <cell line: 2>()
      1 n = 12345
----> 2 lst = list(n)

TypeError: 'int' object is not iterable
```

[21]:

```
n = int(input())
lst = list(map(int, input().split()))
print(lst)
```

```
5
10 20 30 40 50 60
[10, 20, 30, 40, 50, 60]
```

[22]:

```
n = int(input())
lst = list(map(int, input().split()))[:n]
print(lst)
```

```
5
10 20 30 40 50 60 70 80
[10, 20, 30, 40, 50]
```

[23]:

```
# using negative indexes
  #    0   1   2   3
lst = [10, 20, 30, 40]
  #    -4  -3  -2  -1
print(lst[0], lst[-4])
print(lst[1], lst[-3])
```

```
10 10
20 20
```

[24]:

```
lst = [10, 20, 40, 50]
print(lst[-1])
```

```
50
```

[25]:

```
# Traversing through a list
# Index based traversal
# Element based traversal
lst = [10, 20, 30, 40, 50]
for element in lst:
    print(element)
```

```
10
20
30
40
50
```

[26]:

```
lst = [10, 20, 30, 40, 50]
for i in lst:
    print(i ** 2)
```

```
100
400
900
1600
2500
```

[28]:

```
# Index Based Traversal
lst = [10, 20, 30, 40, 50]
# ind  0   1   2   3   4
for i in range(len(lst)): # 0 1 2 3 4
```

```
    print(lst[i])
```

```
10
20
30
40
50
```

[38]:

```
# Find out the number of even numbers in the list
# that are present between 2 odd numbers
lst = [10, 11, 14, 15, 16, 17, 18, 19, 21, 27, 23, 24]
# ind  0   1   2   3   4   5   6   7   8   9   10  11
for i in range(1, len(lst) - 1):
    if lst[i]%2 == 0 and lst[i - 1]%2 != 0 and lst[i + 1]%2 != 0:
        print(lst[i], end = ' ')
```

```
14 16 18
```

[41]:

```
# How many elements we have to read beforehand
lst = [] # lst = list()
n = int(input())
for i in range(n):
    val = int(input())
    lst.append(val)
print(lst)
```

```
5
10
20
30
40
50
[10, 20, 30, 40, 50]
```

[ ]:

```
10
20
30
40
50
```

[40]:

```
lst = [10, 20]
lst.append(30)
print(lst)
```

```
[10, 20, 30]
```

1.2 Slicing

- Getting parts of a list
- [:] –> Single colon slicing

- [::] –> double colon slicing
- [start_index:end_index:index_jump]
- Defaults
  - start_index = 0
  - end_index = len(list)
  - index_jump = 1

[42]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
new_list = lst[0:3:]
print(new_list)
```

```
[10, 20, 30]
```

[44]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
print(lst[::]) # 0 10
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

[45]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
print(lst[2::]) # 2 10
```

```
[30, 40, 50, 60, 70, 80, 90, 100]
```

[47]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
print(lst[:7:]) # 0 6
```

```
[10, 20, 30, 40, 50, 60, 70]
```

[48]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
print(lst[2:7:]) # 0 6
```

```
[30, 40, 50, 60, 70]
```

[49]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
print(lst[::2]) # 0 10
```

```
[10, 30, 50, 70, 90]
```

[50]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
print(lst[::7]) # 0 10
```

```
[10, 80]
```

[52]:

```
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# ind  0   1   2   3   4   5   6   7   8   9
print(lst[2:8:3]) # 0 10
```

```
[30, 60]
```

[54]:

```
# ind -10  -9  -8  -7  -6  -5  -4  -3  -2  -1
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
print(lst[-3:-6:-1]) # default index = 1
```

```
[80, 70, 60]
```

[57]:

```
# Reversing a list elements
# Using a second list
# In-place reverse
lst = [10, 20, 30, 40] # len(lst) = 4 - 1 = 3
# ind  0   1   2   3
lst2 = [] # [40, 30, 20, 10]
for i in range(len(lst) - 1, -1, -1):
    lst2.append(lst[i])
print(lst2)
```

```
[40, 30, 20, 10]
```

[58]:

```
# In-place reverse
lst = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
i = 0
j = len(lst) - 1
while i < j:
    lst[i], lst[j] = lst[j], lst[i]
    i += 1
    j -= 1
print(lst)
```

```
[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]
```