

Exercise

1. Create a list of numbers 100 elements in length that counts by 3s - i.e., [3,6,9,12,...]

```
In [4]: list = [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54,
              57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105,
              108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147,
              150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189,
              192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231,
              234, 237, 240, 243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273,
              276, 279, 282, 285, 288, 291, 294, 297, 300, 303]
print(list)
a = range(3, 301, 3);
```

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 6
3, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114, 11
7, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147, 150, 153, 156, 159, 162, 1
65, 168, 171, 174, 177, 180, 183, 186, 189, 192, 195, 198, 201, 204, 207, 210,
213, 216, 219, 222, 225, 228, 231, 234, 237, 240, 243, 246, 249, 252, 255, 258,
261, 264, 267, 270, 273, 276, 279, 282, 285, 288, 291, 294, 297, 300, 303]
```

2. Using the list from question 1, create a second list whose elements are the same values converted to strings. hint: use a for loop and the function str().

```
In [5]: string_list = [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54,
                      57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105,
                      108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147,
                      150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189,
                      192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231,
                      234, 237, 240, 243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273,
                      276, 279, 282, 285, 288, 291, 294, 297, 300, 303]
print(string_list)
```

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 6
3, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114, 11
7, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147, 150, 153, 156, 159, 162, 1
65, 168, 171, 174, 177, 180, 183, 186, 189, 192, 195, 198, 201, 204, 207, 210,
213, 216, 219, 222, 225, 228, 231, 234, 237, 240, 243, 246, 249, 252, 255, 258,
261, 264, 267, 270, 273, 276, 279, 282, 285, 288, 291, 294, 297, 300, 303]
```

3. Using the list from question 2, create a variable that concatenates each of the elements in order of index (Hint: result should be like "36912...").

```
In [6]: list1 = [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54,
57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105,
108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147,
150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189,
192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231,
234, 237, 240, 243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273,
276, 279, 282, 285, 288, 291, 294, 297, 300, 303]

list2 = [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54,
57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105,
108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147,
150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189,
192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231,
234, 237, 240, 243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273,
276, 279, 282, 285, 288, 291, 294, 297, 300, 303]

for i in list2:
    list1.append(i)

print ("Concatenated list using method : " + str(list1))
```

```
Concatenated list using method : [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36,
39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96,
99, 102, 105, 108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144,
147, 150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189, 192,
195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231, 234, 237, 240,
243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273, 276, 279, 282, 285, 288,
291, 294, 297, 300, 303, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45,
48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102,
105, 108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147, 150,
153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189, 192, 195, 198,
201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231, 234, 237, 240, 243, 246,
249, 252, 255, 258, 261, 264, 267, 270, 273, 276, 279, 282, 285, 288, 291, 294,
297, 300, 303]
```

4. Using .pop() and .append(), create a list whose values are the same as the list from question 1 but in reverse order. (Hint: .pop() removes the last element from a list. The value can be save, i.e., x = lst.pop().)

```
In [7]: r = range(3, 301, 3)
ls = []
for i in r:
    ls.append(i)

new_ls = []
for i in range(len(ls)):
    new_ls.append(ls.pop(-1))
print(new_ls)
```

```
[300, 297, 294, 291, 288, 285, 282, 279, 276, 273, 270, 267, 264, 261, 258, 255, 252, 249, 246, 243, 240, 237, 234, 231, 228, 225, 222, 219, 216, 213, 210, 207, 204, 201, 198, 195, 192, 189, 186, 183, 180, 177, 174, 171, 168, 165, 162, 159, 156, 153, 150, 147, 144, 141, 138, 135, 132, 129, 126, 123, 120, 117, 114, 111, 108, 105, 102, 99, 96, 93, 90, 87, 84, 81, 78, 75, 72, 69, 66, 63, 60, 57, 54, 51, 48, 45, 42, 39, 36, 33, 30, 27, 24, 21, 18, 15, 12, 9, 6, 3]
```

5. Using len(), calculate the midpoint of the list from question 1. Pass this midpoint to slice the list so that the resultant copy includes only the second half of the list from question 1.

```
In [8]: r = range(3, 301, 3)
ls = []
for i in r:
    ls.append(i)
mid = int(len(ls)/2)
half_ls = ls[mid:].copy()
print(half_ls)
```

```
[153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189, 192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231, 234, 237, 240, 243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273, 276, 279, 282, 285, 288, 291, 294, 297, 300]
```

6. Create a string that includes only every other element, starting from the 0th element, from the string in question 3 while maintaining the order of these elements (Hint: this can be done by using a for loop whose values are descending).

```
In [10]: list1 = [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57,
list2 = [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57,

for i in list2:
    list1.append(i)

print("Concatenated list using method : " + str(list1))

string = str()
for i in range(len(list1)):
    if i%2==0:
        string = string+str(list1[i])
```

```
Concatenated list using method : [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36,
39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96,
99, 102, 105, 108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144,
147, 150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189, 192,
195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231, 234, 237, 240,
243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273, 276, 279, 282, 285, 288,
291, 294, 297, 300, 303, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 4
5, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102,
105, 108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147, 150,
153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189, 192, 195, 198,
201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231, 234, 237, 240, 243, 246,
249, 252, 255, 258, 261, 264, 267, 270, 273, 276, 279, 282, 285, 288, 291, 294,
297, 300, 303]
```

7. Explain the difference between a dynamic list in Python (usually referred to as a list) and a tuple.

Ans: List is mutable, and tuple is immutable. For example, if a list already exists, particularelements of it can be reassigned. Along with this, the entire list can be reassigned. Elements andslices of elements can be deleted from the list. On the other hand, particular elements on the tuplecannot be reassigned or deleted, but it is possible to slice it, and even reassign and delete thewhole tuple. Because tuples are immutable, they cannot be copied. They are also different inmemory size wise. In Python, tuples are allocated large blocks of memory with lower overhead,since they are immutable; whereas for lists, small memory blocks are allocated. Between the two,tuples have smaller memory. This helps in making tuples faster than lists when there are a largenumber of elements.

Exploration

1. Use a generator to create a list of the first 100 prime numbers. Include a paragraph explaining how a generator works.

```
In [1]: def prime_generator(end):  
        for n in range(3, end):  
            for x in range(2, n):  
                if n % x == 0:  
                    break  
            else:  
                yield n  
  
list1 = prime_generator(100)  
g = list(list1)  
print('List of prime nummbers ',g)
```

```
List of prime nummbers [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 5  
3, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Include a paragraph explaining how a generator works.

Generator functions in Python allow programmers to declare a function that operates like an iterator, allowing them to create an iterator quickly, easily, and cleanly. Iterators are objects that can be looped or iterated on. It is used to abstract a data container so that it can be utilized as an iterable object. Lists, dictionaries, and strings are examples of iterable objects that are extensively used.

2.Using a for loop and the pop function, create a list of the values from the list of prime numbers whose values are descending from highest to lowest.

```
In [2]: list2 = list()  
        for j in range(0,24):  
            list2.append(g.pop())  
        print (list2)
```

```
[97, 89, 83, 79, 73, 71, 67, 61, 59, 53, 47, 43, 41, 37, 31, 29, 23, 19, 17, 1  
3, 11, 7, 5, 3]
```

3. Using either of prime numbers, create another list that includes the same numbers but is randomly ordered. Do this without shuffling the initial list (Hint: you will need to import random and import copy).

```
In [3]: import random

for k in range(24):
    list3 = list()
    list3 = random.choice(list2)
    print(list3)
```

```
41
5
41
89
67
41
5
61
53
29
5
7
73
71
13
23
41
73
43
97
23
3
71
41
```