Lecture 8: Lists

CS 51P

February 12, 2026

Review: Sequences

- sequences are ordered sets of values
    - ranges are sequences of integers
    - strings are sequences of characters
    - tuples are sequences of arbitrary (possibly mixed) types

- we can perform operations on sequences
    - indexing (e.g., "hello"[0])
    - slicing (e.g., "hello"[1:3])
    - looping (with for loop) (e.g., for char in “hello”:)
    - check membership (e.g., char in "abcd")

Review: Immutability

- strings and tuples are immutable (you can make new sequences, but you can’t change an existing one in place)

```
tup = (3, 6, 2, 1)
tup[0] = 4
```

- TypeError: ’tuple’ object does not support item assignment

Lists

- like a tuple, a list is a sequence or an *ordered collection* of arbitrary items
- Lists start/end with brackets with elements separated by commas.

```
a_list = [3, 6, 2, 1]
float_list = [5.1, 6.2, 0.23]
str_list = [‘this’, ‘is’, ‘a’, ‘list’]
mix_list = [3, 5.1, ‘is’, True]
empty_list = []
```

- like any sequence, we can index into, slice, loop over, check for membership

Key Difference: Lists are mutable

```
a_list = [3.5, 6, (1, 2), "abc"]
```

- ways of modifying an existing list:

1. modify element at index

```
a_list[3] = 3.33333
```

Example 1

- Define a function `double_last` that takes a list of ints `lst` and doubles the last element of the list.

```
my_list = [-1, 2, -3]
x = double_last(my_list)
print(my_list)
print(x)
```

```
[-1, 2, -6]
None
```

Exercise 1

- Define a function `square_neg` that takes a list of ints `lst` and squares all the negative numbers. It should return None.

```
my_list = [-1, 2, -3]
x = square_neg(my_list)
print(my_list)
print(x)
```

```
[1, 2, 9]
None
```

Key Difference: Lists are mutable

```
a_list = [3.5, 6, (1, 2), "abc"]
```

- ways of modifying an existing list:

1. modify element at index

```
a_list[3] = 3.33333
```

2. add element to end

```
a_list.append("c")
```

3. remove element from end

```
x = a_list.pop()
```

Example 2

```
my_list = []
x = my_list.append(-1)
print(x)
print(my_list)

my_list.append(2)
my_list.append(-3)
print(my_list)

x = my_list.pop()
print(x)
print(my_list)

my_list.pop()
print(x)
print(my_list)
```

```
None
[-1]
[-1, 2, -3]
-3
[-1, 2]
-3
[-1]
```

Exercise 2

- Define a function `square_neg` that takes a list of ints `lst` and returns a new list in which the positive numbers are squared.

```
my_list = [-1, -2, 3, 4, -5]
x = square_neg(my_list)
print(my_list)
print(x)
```

```
[-1, -2, 3, 4, -5]
[1, 4, 3, 4, 25]
```

More List Functionality

creating a list

- explicit definition
  - `l = []`
  - `l = [-1, (1,2), "a”]`
- cast from other sequence type
  - `l = list("abc")`

modifying an existing list

- direct assignment
  - `a_list[0] = 2`
  - `a_list[:2] = ["a", "b"]`

adding to an existing list

- a_list.extend(*list*)
- a_list.append(*elem*)
  - Different than extend – e.g. [5, 1]
- a_list.insert(*index*, *elem*)
  - Insert elem at index, shifts down

removing from an existing list

- a_list.remove(*elem*)
  - removes 1st instance of elem
  - error if *elem* not in a_list
- del a_list[*slice*])
  - removes the slice from the list based on the given index
- a_list.pop()
  - returns (and removes) a_list[-1]
- a_list.pop(*index*)
  - returns (and removes) a_list[index]

other occasionally useful stuff

- a_list.copy()
- a_list.reverse()
- a_list.sort()

Exercise 3

```
a_list = [3.5, 6, (1, 2), "abc"]
a_list[3] = list(range(0,5,2))
a_list[:2] = ["a", "b"]
a_list.extend([5,3,1])

print(len(a_list))
for elem in a_list:
     print(str(elem) + ":" + str(type(elem)))

del(a_list[3:5])
a_list.remove("a")
a_list.pop()
print(a_list)
```