

CS 106

Introduction to Computer Science I

07 / 07 / 2026

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- Review: Lists and For loops, range
- Exercises

lists

- Lists are mutable (can change values, add or remove them, etc.) unlike strings which are immutable
- An example list: [45, 26, 89, 32]
- Each list element has an associated index. Indices start at 0 and end at length of the list – 1
- The list: [45, 26, 89, 32] has 45 in the 0 index, 26 in the 1 index, 89 in the 2 index and 32 in the 3 index and the length of that list is 4. Notice the valid indices go 0 .. (4-1)=3
- List terminology: list, element, index

lists

indices within [] (and negative indices)

slices using :

len function

Some useful list methods:

append

remove

insert

clear

lists

`mylist = [ 45, 26, 89, 32]` # creates a list named mylist

`mylist[0]` # refers to the first item in the list which has the value 45

`mylist[3]` # refers to the last item in the list which has the value 32

`mylist[-1]` # another way to specify last item in a list, use -1 index

show on the board the positive indices and negative indices of # the elements

`len(mylist)` # has value 4 because there are 4 elements in the list

lists

```
mylist = [ 45, 26, 89, 32]
```

```
# slices using :
```

```
# mylist[a:b]  # a and b are indices, sublist from index # a  
to b (exclusive)
```

```
# if either a and/or b are left out, 0 and len of the list  
# are assumed, respectively
```

```
mylist[1:3]    # [26,89]
```

```
mylist[:2]     # [45,26]
```

```
mylist[1:]     # [26,89,32]
```

lists

```
mylist = []
```

```
mylist.append('hey') # mylist contains: ['hey']
```

```
mylist.append('now') # mylist contains ['hey', 'now']
```

```
# this inserts 'you' before index 1.
```

```
mylist.insert(1, 'you') # mylist contains ['hey', 'you', 'now']
```

```
mylist.remove('hey') # mylist contains ['you', 'now']
```

```
mylist.clear() # mylist is now []
```

For loop

- Like the while loop we learned recently, a for loop is code that repeats lines of code (iterates over the body of a loop) 0 or more times.
- The syntax is

```
for ____ in ____:  
    # body of the loop  
    # 2nd line in body
```

- The second ____ is a sequence (e.g. a list or a string). The first ____ is a variable name that gets set to an element of the sequence each iteration of the loop.

For loop

- Example:

```
for item in [5,10,15,20]:  
    print(item)
```

- The loop above would allow item to be each element of the list one per iteration
- and so the code will print each element of the list

For loop

- Example:

```
for char in 'Skidmore':  
    print(char)
```

- char would be 'S' the first iteration
- char would be 'k' the next iteration
- and so on
- The code above would print each character in the string 'Skidmore'

For loop with range

- Range returns an object that produces a sequence of integers.
- Range can be called in 3 ways:

`range(stop)`

`range(start, stop)`

`range(start, stop, step)`

- Each of these produces a sequence of ints starting at start and ending just before stop (that is, stop is not inclusive)
- In first call above, start defaults to 0
- step in third call changes the default of 1 to what is passed in

`for i in range(10):`

`print(i)`

- 0,1,2,3,4,5,6,7,8, and 9 will print

For loop with range

```
for i in range(5,10):  
    print(i)
```

5,6,7,8, and 9 will print

```
for i in range(1,20,3):  
    print(i)
```

1, 4, 7, 10, 13, 16, 19 prints

For loops

If you just need to access (not change) the elements of a list, use this style of for loop

```
alist = [5,10,15,20]
```

```
for item in alist:
```

```
    # do whatever you like with item here,
```

```
    # except change its value
```

For loops

If you just need to change the elements of a list, or the order of the elements of a list use this style of for loop

```
alist = [5,10,15,20]
```

```
for i in range(len(alist)):
```

```
    # do whatever you like with alist[i] here
```

```
    # including possibly changing its value
```

Programs to write together

- 0. Write a program that asks the user to repeatedly enter words until quit. Store them in a list. Print out each element of the list using a while loop (and a for loop). Display a random word from the list.
- 1. Ask user to repeatedly enter numbers, q to quit, and show smallest, largest and average of the numbers
 - Write it saving the numbers to a list this time
- 2. determine the product of the odd integers from 1 to 15 inclusive
- 3. modify the last one to ask user for upper limit and whether to do multiply the odds or evens

Programs to write together

- 4. Write a program to count up how many times a value appears in a list. Also show count method
- 5. Show use of type
- 6. Show use of indices to get every pair of characters in a string or elements of a list
- 7. Show difference of using indices in a list vs. with for item in loop