

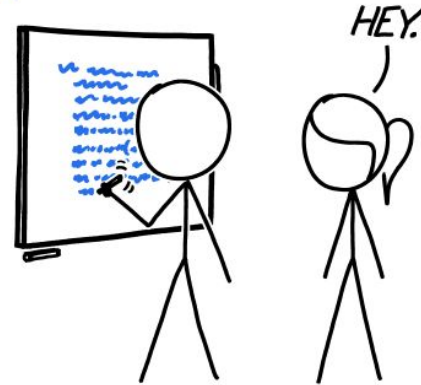
CSc 110

Lists

Adriana Picoral (she/her/hers)

<https://xkcd.com/2483/>

```
define traverseLinkedList(headPointer):
    myID = "1111111111111111"
    authToken = "1111111111111111"
    museumAddress = "1111111111111111"
    client = mailRestClient(myID, authToken)
    client.messages.send(to=museumAddress,
        subj="Item donation?", body="Thought you
        might be interested: "+str(headPointer))
    return
```



CODING INTERVIEW TIP: INTERVIEWERS GET REALLY MAD WHEN YOU TRY TO DONATE THEIR LINKED LISTS TO A TECHNOLOGY MUSEUM.

Announcements

- Exam
 - Next week
 - Study guide 2 is up on the website
 - Review Session Tuesday 5-7pm
- Motion Parallax PA due tomorrow
- Next PA due Oct 25

Lists

```
empty_list = []
```

```
numbers = [1, 5, 2, 10, 7]
```

```
names = ['ron', 'joe', 'kyle']
```

```
values = [1, 1.15, 7, 1.75, 'those']
```

```
values[0]
```

What will this print out?

```
names = 'Alex Jamison, Janette Kirk, Karina Paul'  
names_list = names.split(' ')
```

```
print(names_list)
```

```
i = 0  
while i < len(names_list):  
    print(names_list[i])  
    i += 2
```

What will this print out?

```
names = 'Alex Jamison, Janette Kirk, Karina Paul'  
names_list = names.split(',')
```

```
i = 0  
while i < len(names_list):  
    print(names_list[i])  
    i += 1
```

Changing values in a list

```
ages = [100, 25, 18, 30, 25, 25]
```

```
ages[0] = 27
```

```
ages[3] += 2
```

```
ages[4] -= 1
```

```
ages[1] = ages[2]
```

```
print(ages)
```

Making numbers even

```
weights = [150, 137, 187, 175, 170, 150, 129]  
make_even(weights)  
print(weights)
```

Output would be:

```
[150, 138, 188, 176, 170, 150, 130]
```

Making numbers even

```
def make_even(numbers):  
    i = 0  
    while i < len(numbers):  
        if numbers[i] % 2 == 1:  
            numbers[i] += 1  
        i += 1
```

```
weights = [150, 137, 187, 175, 170, 150, 129]  
make_even(weights)  
print(weights)
```


What will
this print?

```
import random

numbers = [5] * 5

i = 0
while i < len(numbers):
    r = random.randint(4, 10)
    numbers[i] += r
    i += 1

i = 0
while i < len(numbers):
    if numbers[i] <= 8:
        print(numbers[i])
    i += 1
```

Appending to a list

```
ages = []  
ages.append(17)  
ages.append(25)  
ages.append(37)  
print(ages)
```

What will this print?

```
numbers = [10, 12, 17, 20, 7, 21, 8, 7, 25, 27, 50, 70]
others = []
i = 0
while i < len(numbers):
    if numbers[i] % 5 == 0:
        others.append(numbers[i])
    i += 1
i = 0
while i < len(others):
    print(others[i])
    i += 2
```

Print the longest string

```
strings = input('Enter strings separated by spaces: ')\ns1 = strings.split(' ')\n\nprint_longest_name(s1)
```

Example input:

James Ron Richard Rand

Example output:

Richard

Flash Cards

- An app for quizzing someone with simple, text-based "flash cards"
- User provides a number of cards to create, and then the word + definition for each card
- Then, quiz the user!



Flash Cards

Enter number of flashcards to create: 3

Word for card 1: **parameter**

Definition for card 1: **variable assigned the value of an argument when a function is called**

Word for card 2: **boolean**

Definition for card 2: **binary variable: true or false**

Word for card 3: **list**

Definition for card 3: **data structure that allows the storage of multiple values**

. . .

+-----+

| parameter |

+-----+

Press enter to continue

+-----+

| variable assigned the value of an argument when a function is called |

+-----+

Press enter to continue

+-----+

| boolean |

+-----+

Press enter to continue

+-----+

| binary variable: true or false |

+-----+

. . .

```
import random
```

```
def print_card(content):
```

```
    ''' Print a card with the provided content '''
```

```
def quiz(words, definitions):
```

```
    ''' Quiz the user on a single random flashcard '''
```

```
def get_card(words, definitions, card_num):
```

```
    ''' Get a card from the user '''
```

```
def main():
```

```
    ''' What goes here? '''
```

```
main()
```



```
def main():  
    words = []  
    definitions = []  
    num_cards = int(input('Enter number of flashcards to create: '))  
  
    i = 1  
    while i <= num_cards:  
        get_card(words, definitions, i)  
        i += 1  
  
    while True:  
        quiz(words, definitions)
```

Implement get_card

```
def get_card(words, definitions, card_num):  
    '''
```

This function should ask the user for two inputs:
a word and the definition for that word.
Then, it should add the word to the words
list and the definition to the definitions
list.

Can be done in 4 lines of code!
'''

Implement get_card

```
def get_card(words, definitions, card_num):  
    word = input('Word for card ' + str(card_num) + ': '  
    definition = input('Definition for card ' + str(card_num) + ': '  
    words.append(word)  
    definitions.append(definition)
```

```
def get_card(words, definitions, card_num):  
    word = input('Word for card ' + str(card_num) + ': '  
    definition = input('Definition for card ' + str(card_num) + ': '  
    words.append(word)  
    definitions.append(definition)
```

```
def main():  
    words = []  
    definitions = []  
    num_cards = int(input('Enter number of flashcards to create: '))  
    i = 1  
    while i <= num_cards:  
        get_card(words, definitions, i)  
        i += 1  
    while True:  
        quiz(words, definitions)
```

```
main()
```

Implement print_card

```
def print_card(content):
```

```
    '''
```

The function should print out a text card, with the provided content as the text on the card.

For example, if content = "Hi There", then this function should print out:

```
+-----+
```

```
|  Hi There  |
```

```
+-----+
```

```
    '''
```

Implement print_card

```
def print_card(content):  
    length = len(content)  
    print('+--' + '-' * length + '--+')  
    print('|' + content + '|')  
    print('+--' + '-' * length + '--+')
```