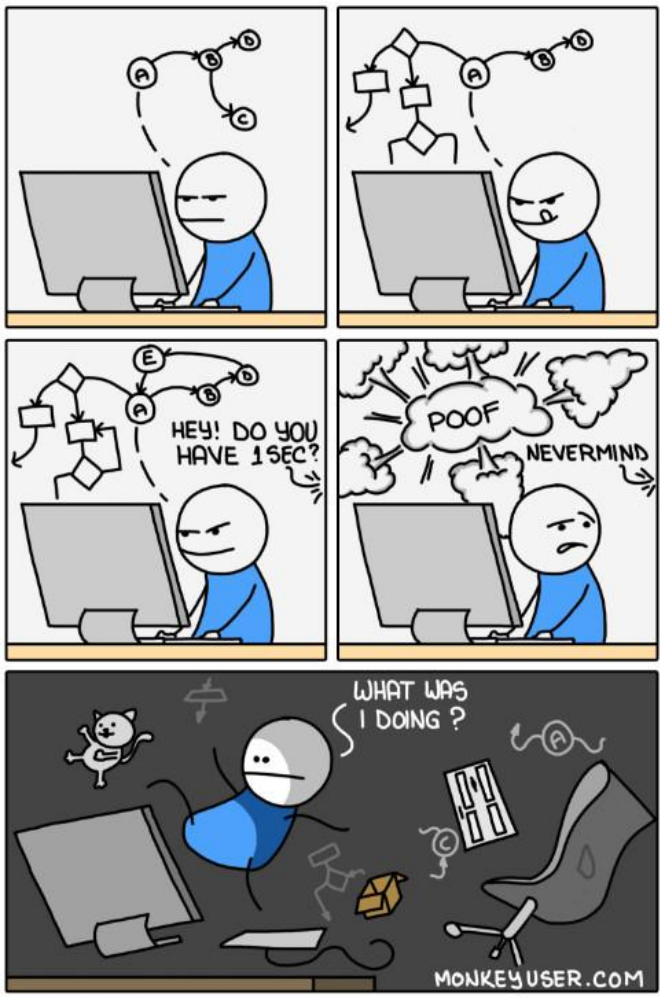

CSc 110
Lists

Benjamin Dicken

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: Grace
Definition for card 1: Favor; good will; kindness
Word for card 2: Faith
Definition for card 2: The assent of the mind to the truth
of a proposition advanced by another
Word for card 3: Peace
Definition for card 3: In a general sense, a state of quiet
or tranquillity
```

```
. . .
+---------+
|  Faith  |
+---------+
Press enter to continue
+-------------------------------------------------------------------------+
|  The assent of the mind to the truth of a proposition advanced by another  |
+-------------------------------------------------------------------------+
Press enter to continue
+---------+
|  Peace  |
+---------+
Press enter to continue
+-----------------------------------------------------+
|  In a general sense, a state of quiet or tranquillity  |
+-----------------------------------------------------+
. . .
```

Flash Cards

How would you go about implementing this, given what has been covered in class previously?

What types have been covered in class so far?

What types have been covered in class so far?

integer **float** **boolean** **string**

Lists

- A **list** is another type
- Lists are **sequences** of zero or more values
- A list is a type of **Data Structure**

Data Structure

“In computer science, a **data structure** is a data organization, management and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.”

https://en.wikipedia.org/wiki/Data_structure

Creating Lists

```
no_numbers = []

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

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

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

print(type(values))
```

What would this print?

```
values = ['the', 'bear', 'in', 'the', 'tree']
print(values[3] + values[4] + values[1])
```

Print each value in the list using **while** loop

```
values = ['those', '123', '5i', 'those', '4']
```

Should Print:

```
those
123
5i
those
4
```

Print each value in the list using **while** loop

```
values = ['those', '123', '5i', 'those', '4']

i = 0
while i < 5:
    print(values[i])
    i += 1
```

Print each value in the list using **while** loop

```
values = ['those', '123', '5i', 'those', '4']

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

Check the word lengths

```
values = ['those', 'hi', 'incomprehensibilities', 'yo', 'instrumental']
```

Should Print:

```
3 regular word(s)
2 large word(s)
```

Check if each value is numeric

```
values = ['those', 'hi', 'incomprehensibilities', 'yo', 'instrumental']

i = 0
regular = 0
large = 0
while i < len(values):
    word = values[i]
    if len(word) > 7:
        large += 1
    else:
        regular += 1
    i += 1

print(regular, 'regular word(s)')
print(large, 'large word(s)')
```

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 += 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)
```

Appending to a list

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

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
```

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: ')
sl = strings.split(' ')

print_longest_name(sl)
```

Example input:

```
James Ron Richard Rand
```

Example output:

```
Richard
```