CS 209
Data Structures and Mathematical Foundations

01 / 25 / 2024

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- more Python review
  - For loops
  - Calling functions
  - print vs. return
  - Defining our own functions

python review

- Calling functions
  - To be able to call functions correctly we
  - Must pay attention to the required parameters and their expected types
  - Must pay attention to whether or not the function **return**s anything

python review

- Let's look at the parameters and what these return:
- append, sort, sum, len, random.randint, input, print

python review

- Let's look at the parameters and what these return:
- append, sort, sum, len, random.randint, input, print
- append, sort and print each return None
- sum, len, random.randint return a number
- input returns a str

python review

- Because append, sort and print return None we should not call them within print, nor have them on the righthand side of an assignment or other place where we are expecting a value
  - Instead functions that return None should be called on a line by themselves (not within print and without operators like =)
- Because sum, len, random.randint return a number we should call them either on the right hand side of an assignment statement or other place where we are expecting a value
- Similarly with input which returns a str.

python review

- Most functions do not print anything.
  - Examples we used recently:
    - append, sort, sum, len, random.randint
- There were two built-in functions that did print something. What were they?

python review

- Most functions do not print anything.
  - Examples we used recently:
    - append, sort, sum, len, random.randint
- There were two built-in functions that did print something. What were they?
  - input and print
- When we write our own functions we will tend NOT to have them print anything. Most of the time they are to do something (other than print) and/or return an answer.

python review

- Appropriate calls to these functions:

```
ages = [22,64,31,19]
ages.append(25)
ages.sort()


total = sum(ages)
numages = len(ages)
random_num = random.randint(1,10)
```

python review

- Appropriate calls to these functions:

```
firstname = input('what is your first name?')
print('hello', firstname)
```

For loop

- 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 code above would print each element of the list
- item becomes the value 5 1st iteration, becomes 10 the next iteration and so on

For loop

- Example:

```
for char in ‘Skidmore’:
    print(char)
```

- The code above would print each character in the string ‘Skidmore’
- char becomes the value ‘S’ in the first iteration, ‘k’ the 2nd iteration and so on

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

python review

- If I ask us to write a function that returns something, I mean we write a def and make that function return something.
- I am not asking you to just write code (outside of a function) to accomplish that task.
- Also remember that we generally avoid printing an answer in a function and prefer to **return** the answer instead.

python review

- Let's write a program that asks the user how many numbers he/she will enter and then prompts the user to enter that many numbers and then:
- prints out the list of those numbers
- prints out the list of those numbers sorted from low to high
- prints the average of the numbers (we'll implement an average function)
  - The average function will take in a list of numbers as a parameter and **return** the average of those numbers

python review

- Revisions: Let's write our own selection sort function and call it instead of python3's built-in sort function
  - Selectionsort
    - will NOT return anything
    - will sort the list passed in to the parameter