Lists in Python

February 10 2020

Administrative Notes

Homework #1 still due tonight before midnight

- If you’re having problems submitting let me know

Scalar variables

To date, all variables we've learned about in Python are ***scalar*** - they can hold exactly one value at a time.

Strings hold one set of characters (that can vary in length, but it's still a single thing)

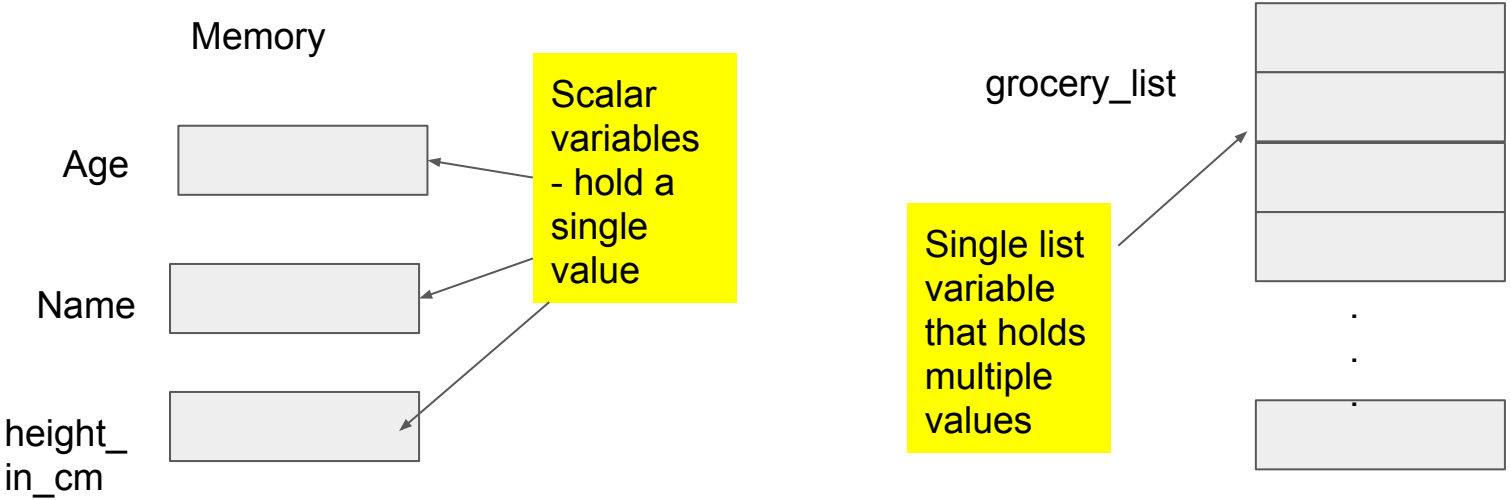
Ints can hold a single integer, floats can hold a single real number, and booleans can hold a single value of "true" or "false"

What if you need to manage more than one value?

Lists

Lists are the fundamental way that Python manages multi-valued variables

- Warning: You might get the impression from lecture and the book that a list is a one-dimensional array. Don't believe that. Arrays are different.



Example - my grocery list from this weekend

Milk
Eggs
Cereal
Coffee
Apples
Strawberries
Broccoli
Cucumber
Tomatoes
Green Onions
Half & Half

In this example, everything will be a string

Create a list variable in Python

1. Create an empty list so that we can later add items to it

Grocery_list = []

Square brackets mean “create an empty list”

2. Create a list with the values already inserted

Grocery_list = [“Milk”, “Eggs”, “Cereal”, “Coffee”, “Apples”, “Strawberries”, “Broccoli”, “Cucumber”, “Tomatoes”, “Green Onions”]

Square brackets mean “create a list.” Elements of the list are separated by commas. Double quotes mean the elements are strings.

We operate on lists using “methods”

“Method” is a term from object-oriented programming that defines an operation that can be peformed on a object/variable

Invoked by putting a dot, and then the method name, after the variable name.

Methods for lists:

append - add an element to the end of the list

remove - remove a designated element from the list

Insert - put an element into a designated space in the list

So building the list

```
grocery_list = []   #create an empty list

grocery_list.append(“Milk”)

grocery_list.append(“eggs”)

grocery_list.append(“cereal”)

…

grocery_list.append(“green onions”)
```

Indexing the list

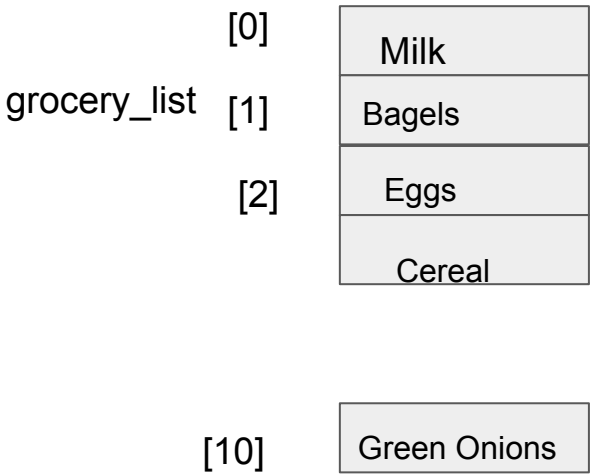
In computer science, we *almost* always start numbering from 0. The first element in any list in Python is [0] - e.g., grocery_list[0] contains “Milk”

Then go up by one. My grocery list has 10 items on it. So the last item, “green onions” is stored in grocery_list [9].

Adding to the list - My wife texts. Can I please pick up a half-dozen bagels, too?

grocery_list.append("bagels")

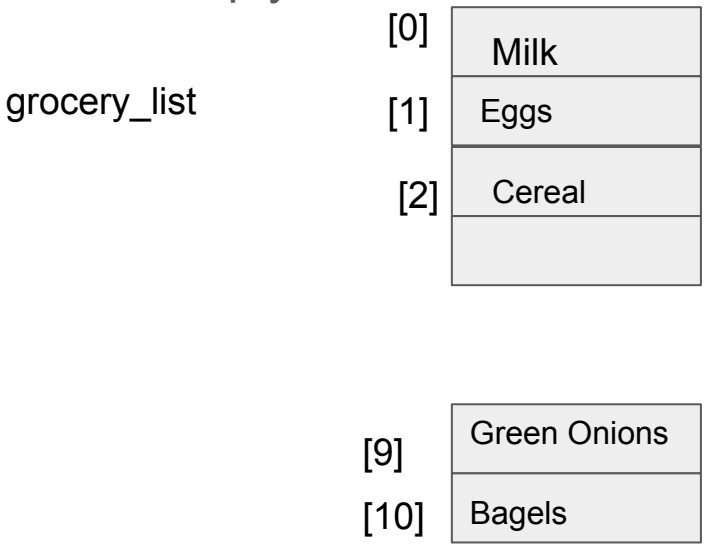
grocery_list.insert(1. "bagels")

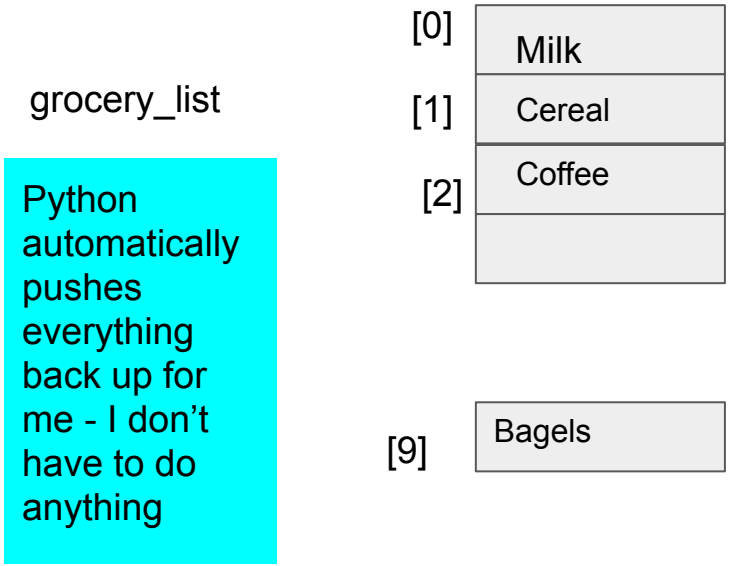


Python automatically pushes everything down for me - I don't have to do anything

Removing an item (by its value)

Suppose I want to remove each item from my list as I put it in the cart. When the list is empty I'm done. Use the .remove method: grocery_list.remove("Eggs")





Removing an element by its index uses a different method - pop

Useful tools to manipulate lists

How long is it? The “len” function.

len(grocery_list)

Note - this gives the total number of elements in the list. So it’s always equal to one more than the last index. Or, the last index is len(grocery_list) - 1.

What happens if you try

grocery_list[len(grocery_list)]

What about grocery_list[len(grocery_list) - 1]

Is a particular value in a list?

The reserved word “in” is useful for this

“Eggs” in grocery_list returns a boolean value

Lots more list operations available

But we'll get to them later in the semester. For the rest of class we'll work on some coding.

Questions about lists

Do all elements of a list have to be the same type (all ints, all floats, all strings?)

- NO!! Python can sort the types out and manage them. But you'll generally make all of your list elements the same type because otherwise you get into really bad design, really fast

When should I use a list?

- When you have a collection of data elements of the same type that logically go together - *items on my grocery list*!!

When should I *not* use a list?

- When you have multiple data elements that really don't belong together - *name, birthdate, age, major, height, weight of a student*
    - You would use an array for this, and lists are not arrays!!