

More about Recursion

November 14

Reminders

Homework # 6 due tomorrow night

Remember the list of permitted and forbidden magicks - python methods and functions that you can or cannot use on specific lessons

- `list.extend()` is permitted on HW6

A clarification of `list.extend()`

`list.extend(seq)` takes the elements of `seq` and adds them to the end of `list`, one element at a time

“`seq`” MUST be a list or dict - an iterable type. You cannot “extend” a list with, say, an integer

This differs from `list.append(var)` which will put “`var`” at the end of the list, no matter what type “`var`” is.

Appending a list to a list

list.extend()

I didn't make that clear enough Tuesday

What I said about `append()` adding everything at once while `extend()` cycles through and adds one element at a time is true and important - appending a list to a list produces very different behavior than extending a list by a list

Don't try to assign the result of an `extend()` in the same statement

HW #6 part 2

Part 2: base case: the lists are zero or one element long

(we'll presume that you've already done error checking and made sure that the lists are the same length)

- What do you do?

Recursive case: if you have lists with more than one element, chop off an element and make the recursive call.

- Which one do you get rid of?

HW#6 part 3

NO string methods other than len() and splicing

- So what can you do?

This is recursion. What's the base case?

And what's the recursive case? How do you make the problem shorter or smaller?

On to some more examples of recursion

String permutations

Euclid's Greatest Common Divisor (GCD) algorithm