Announcements

- Python lab hours next week:
  - Mon 9-11, review of files and string manipulation. Get started on Prog. 5 if you have not already.
  - Tues 10-12, using dictionaries. Also relevant to Program 5.
  - Wds 11-1, normal lab hours.
  - Thurs 4-6, normal lab hours.

Start early!!

Variable in functions are local
def addTwo(x):
def main():
main()

After this runs, what's x? What's y?

The Rule

- If you're using functions, no code outside the main() function.

def main():
def addTwo(x):
def main():
def addTwo(x):
def main():

Why?

- This makes the behavior of variables pretty simple; every function has its own local variables.
- The x in one function is a different variable from the x in another function.
- This is how the book does it, and the reason it always puts programs in main().

Order of stuff in program

- Imports first.
- All function definitions, with main() last.
- At the very end, just one statement outside a function: main().

One way in, one way out

def addTwo(z):
def main():
def addTwo(z):
def main():

Input to a function is through its parameters (can be lots).

main() passes data to addTwo().
x is the argument of addTwo() in main()
z is the parameter of addTwo()

Output is only through its return value (only one).
z is the return value of addTwo()

Benefits of The Rule

- Variables in a function are local to that function.
- Information is passed between functions using arguments/parameters and return values.
- The flow of information is clear.
- The entire function is reading/writing to the same set of the parameters.

But Change! Variables

- When you pass a mutable data structure to another function, it's best practice to pass a copy by value.
- This makes the function's changes not affect the original.
- Mutation (in this case dictionaries) can be avoided.

Why? Lists and dictionaries

- One variable for whole filing cabinet full of data.
- Don't want to make a whole new filing cabinet every time you change one item (mutable).
- Don't want to make a whole new filing cabinet every time you use a function.

One way in, one way out

Inside Python - “pass by value”
main() inc
x
1
y

Inside Python - “pass by reference”
main() inc
x
The entire mutable data structure is passed.
L
[2]

Why? Lists and dictionaries

- Lists and dictionaries can be passed as parameters.
- Remember that changes to mutable objects are to the single copy of that object…
- …so even if they are not returned as return values, they might be altered by the function so data gets back to the calling function.
- Changes to immutable (normal) variables are to the local copies within each function.