ECS 10

11/28

Announcements
- Program 6 due Tues night, Makeup due Thurs night.
- Wednesday course evaluations.
- Review for final on Friday in class.
- Final is this room, Monday Dec 5, 8AM.
- Last year’s final is on SmartSite.
- We’ll schedule some office hours for Sunday.

Organizing big programs
- Put conceptually different parts of program in different functions.
- Variables defined in each function are local to that function.
- If the only line in the program not in a function is main() (and imports), then all variables are local variables.
- Information is passed explicitly using arguments/parameters and return values.
- Flow of information is visible.

Sometimes difficult to do
- Example: a function needs to remember info from last time it was called.
- Could remember the info in main(), but this defeats the purpose of separating conceptually different parts of the program from each other.

Global variables
- Shared by all functions in the program
- Here x is declared to be global in setsX()

```python
def seesX():
    print(x)

def setsX():
global x
x = 5

def main():
    setsX()
    seesX()

main()
```

Global variables
- All functions can see the value of a global variable.
- Only the ones that declare it can change its value.

```python
def seesX():
    print(x)

def setsX():
global x
x = 5

def main():
    setsX()
    seesX()

main()
```
Function can see globals, but not change them (unless it declares them).

**Metaphor – one-way glass**

- When using global variables, the flow of information might not be clear.
- Keep global variables to a minimum.
- Calling a function should not have side effects; break this rule at your own peril.

**Button callbacks**

- Have no parameters, so no way to get data in.
- But need to be able to change variables containing GUI widgets.
- Make those widgets global.

**Tricky bit #1**

- A variable defined in a function, not declared to be global, is local.

```python
def f():
    x = 3
    print(x)

def g():
    x = 5
    print(x)

def seesX():
    print(x)

def setsX():
    global x
    x = 5
```

**Tricky bit #2**

- Variables outside a function are always global. It’s easy to forget this.
- Forgetting a variable is global introduces all kinds of bugs.
- Since we want to be very aware of all global variables,
  - don’t put code outside functions, (except imports and the call to main()),
  - watch out for assignments with global variables on the left; they have to be declared global in that function.

```python
def main():
    setsX()
    localX()
    seesX()
```
Style review

- Order of stuff in your program:
  1. Imports. Import everything in the first lines.
  2. All function definitions, with main() last. Functions get run only when they are called.
  3. One statement outside of function definitions: main()
- Declare all global variables explicitly.