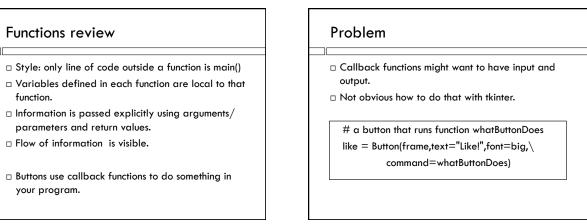
ECS 10 3/13

Announcements $\hfill\Box$ Final is Wds Mar 20, 1-3pm, in this room. □ Prog 6 due Sunday Mar 17 $\hfill\Box$ Practice final on SmartSite under resources, a few more programming examples to follow soon.

Functions review □ Style: only line of code outside a function is main() $\hfill\Box$ Variables defined in each function are local to that function. □ Information is passed explicitly using arguments/ parameters and return values. □ Flow of information is visible. $\hfill \square$ Buttons use callback functions to do something in your program.



Solution: Global variables		
□ Shared by all functions in the program	def usesX(): print(x)	
□ Here x is declared to be global in	def createsX(): global x x = 5	
setsX()	def main(): createsX() usesX()	
	main()	

Global variables		
□ All functions can see the value of a global	<pre>def usesX(): print(x) def createsX():</pre>	
variable. □ Only the ones	global x x = 5	
that declare it global can change its value .	def main(): createsX() usesX()	
	main()	



Metaphor – one-way glass

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

Style tips

- ☐ When using global variables, the flow of information might not be clear.
- □ Keep global variables to a minimum.

Button callbacks

- □ Have no parameters or return values, so no way to get data in and out.
- □ But need to be able to change variables in the program in order to do anything!
- □ Use global variables.

Tricky bit #1

 $\hfill\Box$ A variable defined in a function, not declared to be global, is local.

def f(): x = 3 # x is local def f():
global x
x = 3
x is global

Tricky bit #1

- ☐ This is true even if there is a global variable of the same
- $\hfill\Box$ Watch out for this!

def main():
 createX()
 localX()
 seesX()

def localX():
 x = 2
 print(x)

def seesX():
 print(x)

def createX():
 global x
 x = 5

Tricky bit #2

- □ Variables outside a function are always global. It's easy to forget this and get confused.
- ☐ Since we want 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.

Style review

- $\hfill\Box$ 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()
- $\hfill\Box$ Declare all global variables explicitly.

Another time globals are handy

- 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.