Global Variables

Scope of Variables

Using the value of a global

- You may use the value of a global variable anywhere in your program without any problems

```
myPup = “Trixie”
def main():
        print(“My pup’s name is”, myPup)
        return
main()
```

Changing the value of a global variable

- If you try to change the value of a global variable like in the following example, it doesn’t work

```
myPup = “Trixie”
def main():
        myPup = “Trooper”
        print(“My pup’s name is”, Trooper)      #Prints Trooper
main()
print(“My pup’s name is”, myPup)      #Prints Trixie
```

Why?

- Because of the way that Python works
- When there is an assignment statement, Python checks to see if there is a local variable, and if so, reassigns the local variable to the new value
- If there isn’t a local variable of that name, then it creates one.
- See the problem?

How to change global variables

- Tell your program that you are using the global variable in your function before you try to do anything with it
- Keyword global

```
myPup = “Trixie”
def main():
        global myPup
        myPup = “Trooper”
        print(“My pup’s name is”, Trooper)        #Prints Trooper
main()
print(“My pup’s name is”, Trooper)        #Prints Trooper
```