

# Python! Refactoring Redundant Code

Contributed by Kevin Bell  
23, Jul. 2008  
Last Updated 01, Aug. 2008

If you ever need to do repetitive work then you can either code each operation one after the next which can make your script really long winded, or you can make your code more efficient by utilizing functions. The code below has a dictionary containing a whereclause for each desired output layer. I could code something like this:

- make feature layer1
- kernel density from lyr1
- make feature layer2
- kernel density from lyr2
- make feature layer3
- kernel density from lyr3
- etc
- etc
- etc...

A "cleaner" approach is use something like the code below, which passes each key and value from a dictionary that consist of the output hotspot rasters-to-be's name, and the whereclause that subsets the data from a master Feature Class.

```
import os
import arcgisscripting
gp = arcgisscripting.create()
gp.OverwriteOutput = 1
gp.workspace = "C:\\StagingDBs\\slc\\dot.gdb"
try:
    gp.CheckOutExtension("Spatial")
except:
    print "can't acquire Spatial... exiting..."
    os.sys.exit()

allAcc = "C:\\StagingDBs\\slc\\dot.gdb\\accidents\\allAccidents"

rasterDict = {
'allAccidents': "\"Score\" >70 AND \"Status\" <> 'U'",
'allAccidentsDark': "\"Score\" >70 AND \"Status\" <> 'U' AND \"IsDark\" = 1",
'peds': "\"Score\" >70 AND \"Status\" <> 'U' AND \"ped_invol\" > '0'",
'pedsDark': "\"Score\" >70 AND \"Status\" <> 'U' AND \"ped_invol\" > '0' AND \"IsDark\" = 1",
'bikes': "\"Score\" >70 AND \"Status\" <> 'U' AND \"bicyclist_\" > '0'",
'bikesDark': "\"Score\" >70 AND \"Status\" <> 'U' AND \"bicyclist_\" > '0' AND \"IsDark\" = 1",
'amPeak': "\"Score\" >70 AND \"Status\" <> 'U' AND \"occ_time\" > 700 AND \"occ_time\" < 900",
'pmPeak': "\"Score\" >70 AND \"Status\" <> 'U' AND \"occ_time\" > 1600 AND \"occ_time\" < 1800"
}

def makeDensity(gp, rasterToBe, where_clause):
    layername = os.path.basename(rasterToBe) + "_lyr"
    gp.MakeFeatureLayer_management(allAcc, layername, where_clause)
    gp.KernelDensity_sa(layername, "NONE", rasterToBe, "50", "350", "SQUARE_MAP_UNITS")
    print "...finished Density surface for " + rasterToBe

for k, v in rasterDict.iteritems():
    makeDensity(gp, k, v)
```