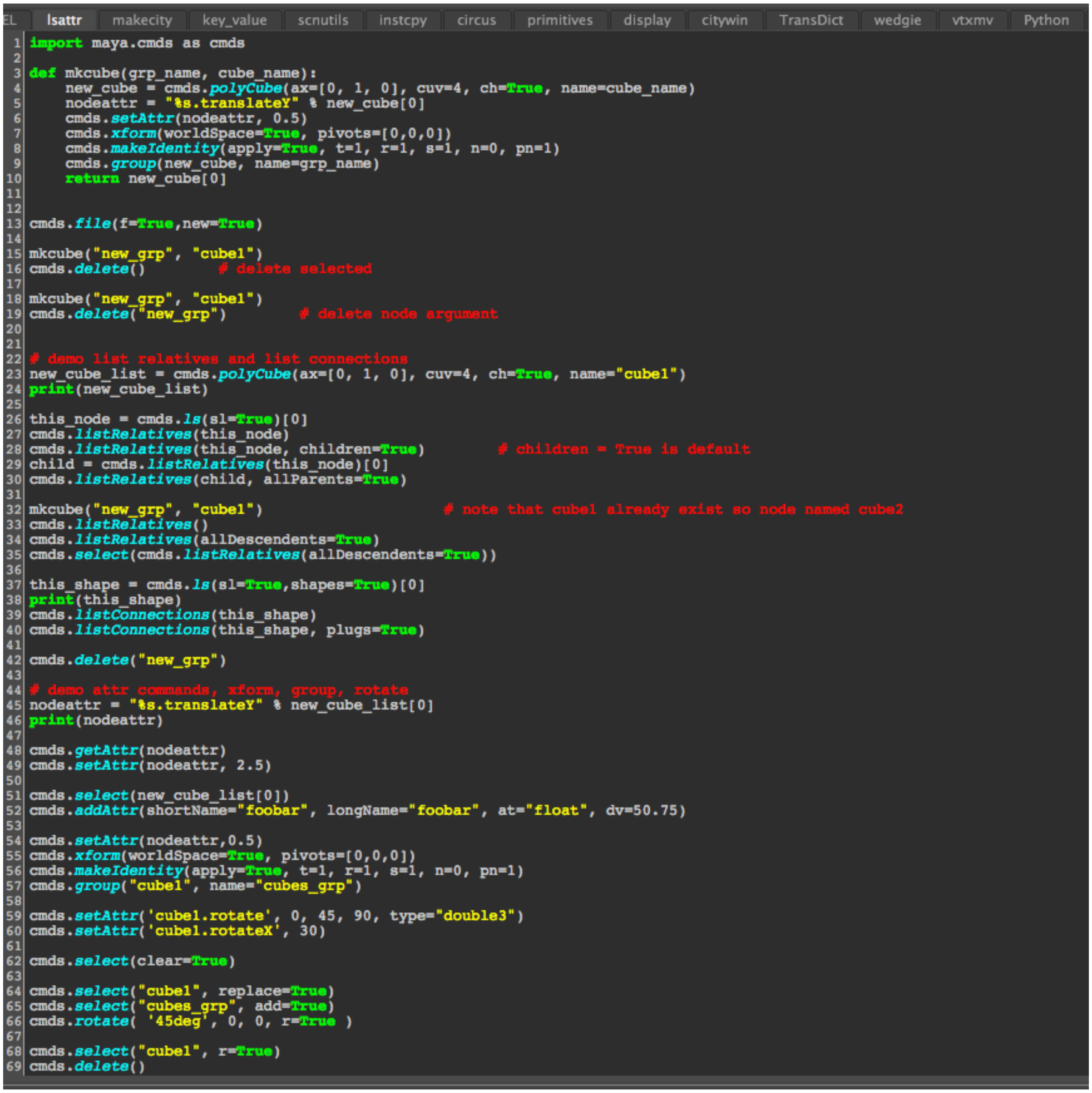
Maya Python Tabs in Script Editor

By: Suzanne Berger, VanArts Instructor: Intro To Python

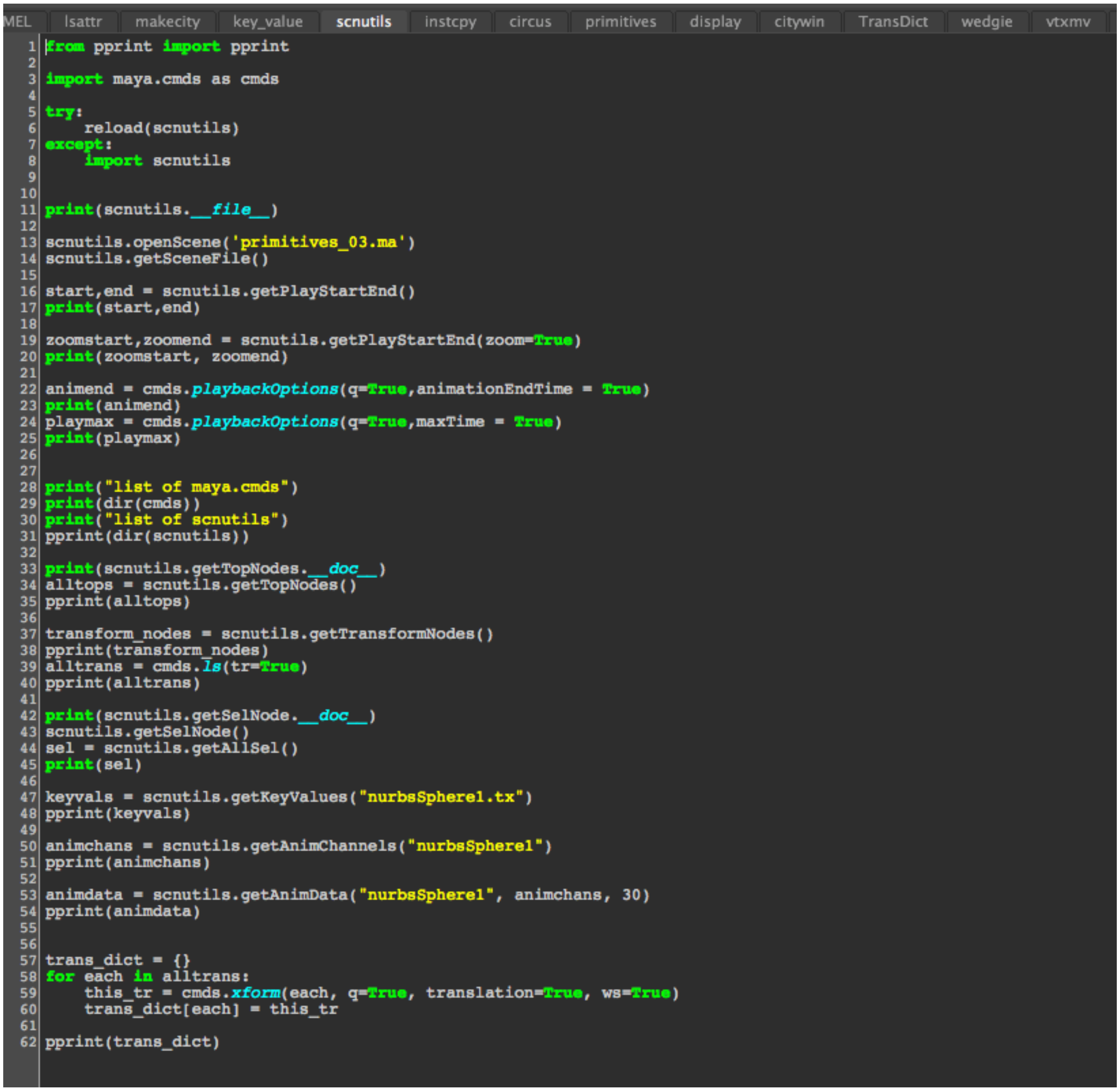
Date: October 12, 2020

LSATTR



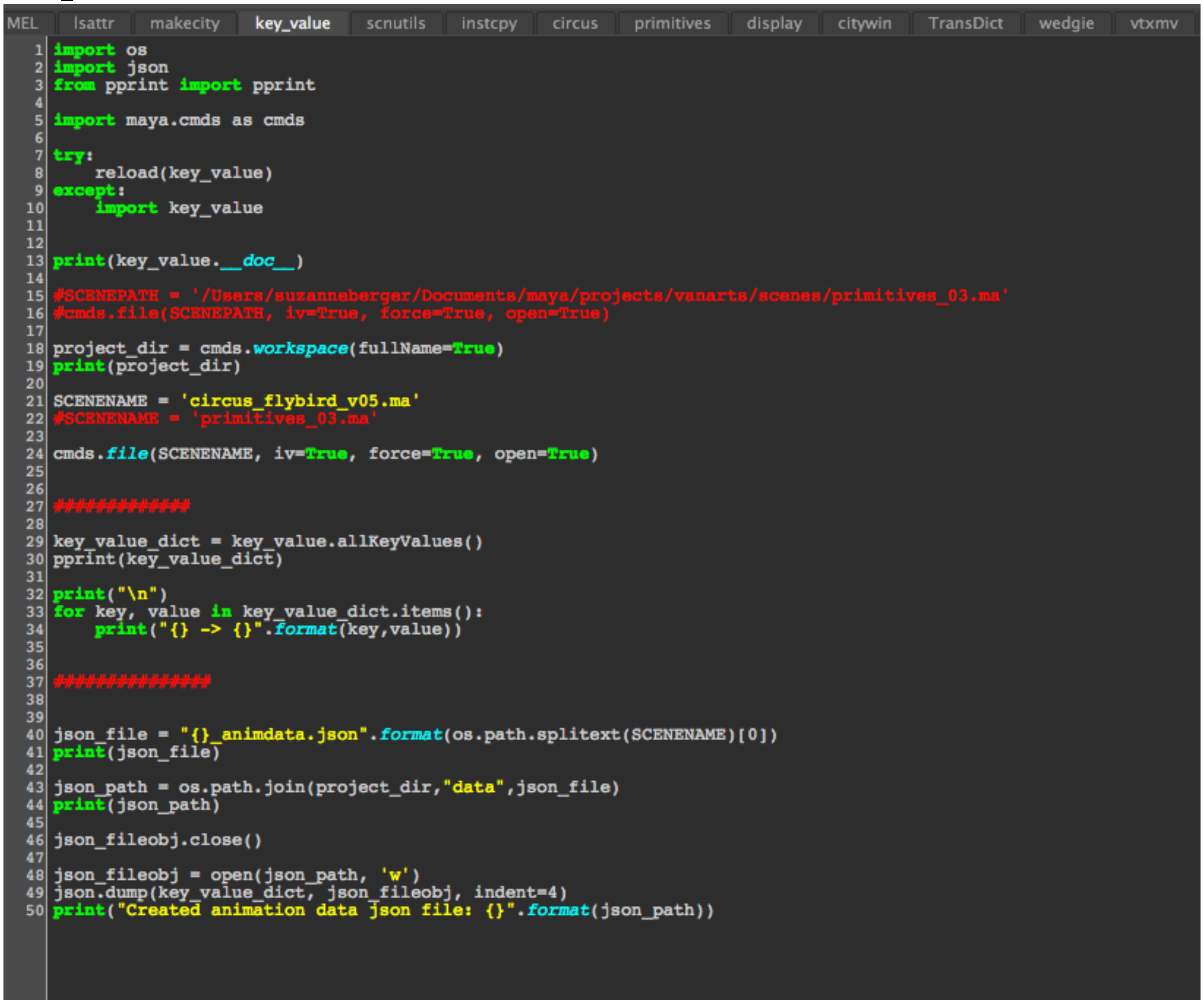
```python
import maya.cmds as cmds

def mkcube(grp_name, cube_name):
    new_cube = cmds.polyCube(ax=[0, 1, 0], cuv=4, ch=True, name=cube_name)
    nodeattr = "%s.translateY" % new_cube[0]
    cmds.setAttr(nodeattr, 0.5)
    cmds.xform(worldSpace=True, pivots=[0,0,0])
    cmds.makeIdentity(apply=True, t=1, r=1, s=1, n=0, pn=1)
    cmds.group(new_cube, name=grp_name)
    return new_cube[0]


cmds.file(f=True,new=True)

mkcube("new_grp", "cube1")
cmds.delete()        # delete selected

mkcube("new_grp", "cube1")
cmds.delete("new_grp")        # delete node argument


# demo list relatives and list connections
new_cube_list = cmds.polyCube(ax=[0, 1, 0], cuv=4, ch=True, name="cube1")
print(new_cube_list)

this_node = cmds.ls(sl=True)[0]
cmds.listRelatives(this_node)
cmds.listRelatives(this_node, children=True)        # children = True is default
child = cmds.listRelatives(this_node)[0]
cmds.listRelatives(child, allParents=True)

mkcube("new_grp", "cube1")                          # note that cube1 already exist so node named cube2
cmds.listRelatives()
cmds.listRelatives(allDescendents=True)
cmds.select(cmds.listRelatives(allDescendents=True))

this_shape = cmds.ls(sl=True,shapes=True)[0]
print(this_shape)
cmds.listConnections(this_shape)
cmds.listConnections(this_shape, plugs=True)

cmds.delete("new_grp")

# demo attr commands, xform, group, rotate
nodeattr = "%s.translateY" % new_cube_list[0]
print(nodeattr)

cmds.getAttr(nodeattr)
cmds.setAttr(nodeattr, 2.5)

cmds.select(new_cube_list[0])
cmds.addAttr(shortName="foobar", longName="foobar", at="float", dv=50.75)

cmds.setAttr(nodeattr,0.5)
cmds.xform(worldSpace=True, pivots=[0,0,0])
cmds.makeIdentity(apply=True, t=1, r=1, s=1, n=0, pn=1)
cmds.group("cube1", name="cubes_grp")

cmds.setAttr('cube1.rotate', 0, 45, 90, type="double3")
cmds.setAttr('cube1.rotateX', 30)

cmds.select(clear=True)

cmds.select("cube1", replace=True)
cmds.select("cubes_grp", add=True)
cmds.rotate( '45deg', 0, 0, r=True )

cmds.select("cube1", r=True)
cmds.delete()
```

SCNUTILS



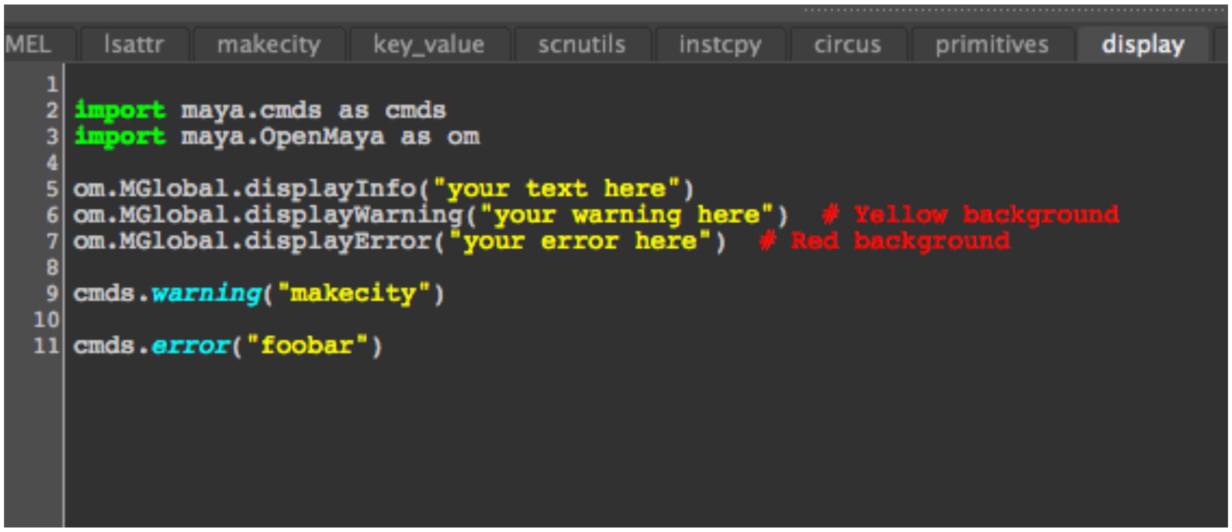
```python
from pprint import pprint

import maya.cmds as cmds

try:
    reload(scnutils)
except:
    import scnutils


print(scnutils.__file__)

scnutils.openScene('primitives_03.ma')
scnutils.getSceneFile()

start,end = scnutils.getPlayStartEnd()
print(start,end)

zoomstart,zoomend = scnutils.getPlayStartEnd(zoom=True)
print(zoomstart, zoomend)

animend = cmds.playbackOptions(q=True,animationEndTime = True)
print(animend)
playmax = cmds.playbackOptions(q=True,maxTime = True)
print(playmax)


print("list of maya.cmds")
print(dir(cmds))
print("list of scnutils")
pprint(dir(scnutils))

print(scnutils.getTopNodes.__doc__)
alltops = scnutils.getTopNodes()
pprint(alltops)

transform_nodes = scnutils.getTransformNodes()
pprint(transform_nodes)
alltrans = cmds.ls(tr=True)
pprint(alltrans)

print(scnutils.getSelNode.__doc__)
scnutils.getSelNode()
sel = scnutils.getAllSel()
print(sel)

keyvals = scnutils.getKeyValues("nurbsSphere1.tx")
pprint(keyvals)

animchans = scnutils.getAnimChannels("nurbsSphere1")
pprint(animchans)

animdata = scnutils.getAnimData("nurbsSphere1", animchans, 30)
pprint(animdata)


trans_dict = {}
for each in alltrans:
    this_tr = cmds.xform(each, q=True, translation=True, ws=True)
    trans_dict[each] = this_tr

pprint(trans_dict)
```

KEY_VAL



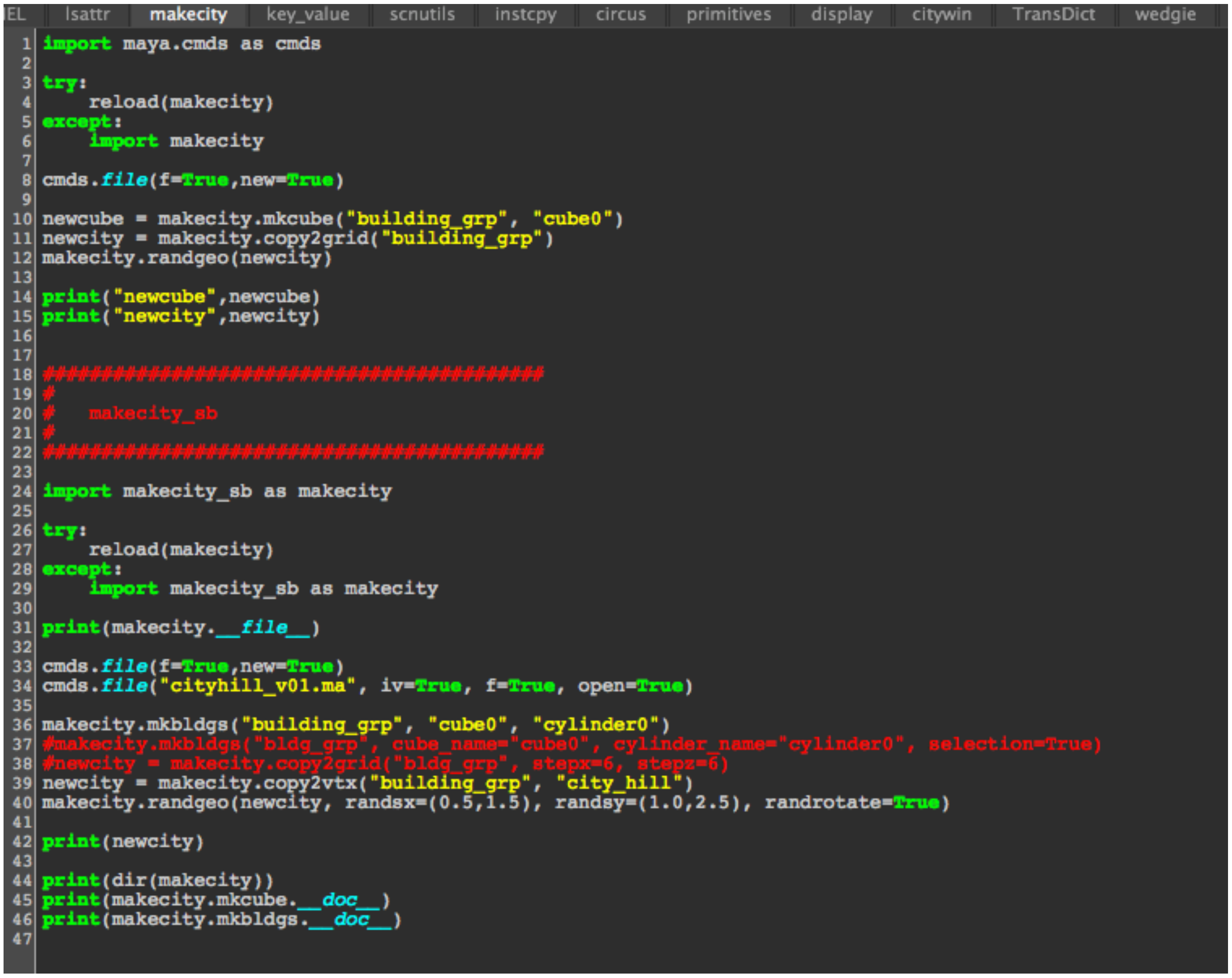
```python
import os
import json
from pprint import pprint

import maya.cmds as cmds

try:
    reload(key_value)
except:
    import key_value


print(key_value.__doc__)

#SCENEPATH = '/Users/suzanneberger/Documents/maya/projects/vanarts/scenes/primitives_03.ma'
#cmds.file(SCENEPATH, iv=True, force=True, open=True)

project_dir = cmds.workspace(fullName=True)
print(project_dir)

SCENENAME = 'circus_flybird_v05.ma'
#SCENENAME = 'primitives_03.ma'

cmds.file(SCENENAME, iv=True, force=True, open=True)


#############

key_value_dict = key_value.allKeyValues()
pprint(key_value_dict)

print("\n")
for key, value in key_value_dict.items():
    print("{} -> {}".format(key,value))


###############


json_file = "{}_animdata.json".format(os.path.splitext(SCENENAME)[0])
print(json_file)

json_path = os.path.join(project_dir,"data",json_file)
print(json_path)

json_fileobj.close()

json_fileobj = open(json_path, 'w')
json.dump(key_value_dict, json_fileobj, indent=4)
print("Created animation data json file: {}".format(json_path))
```

DISPLAY



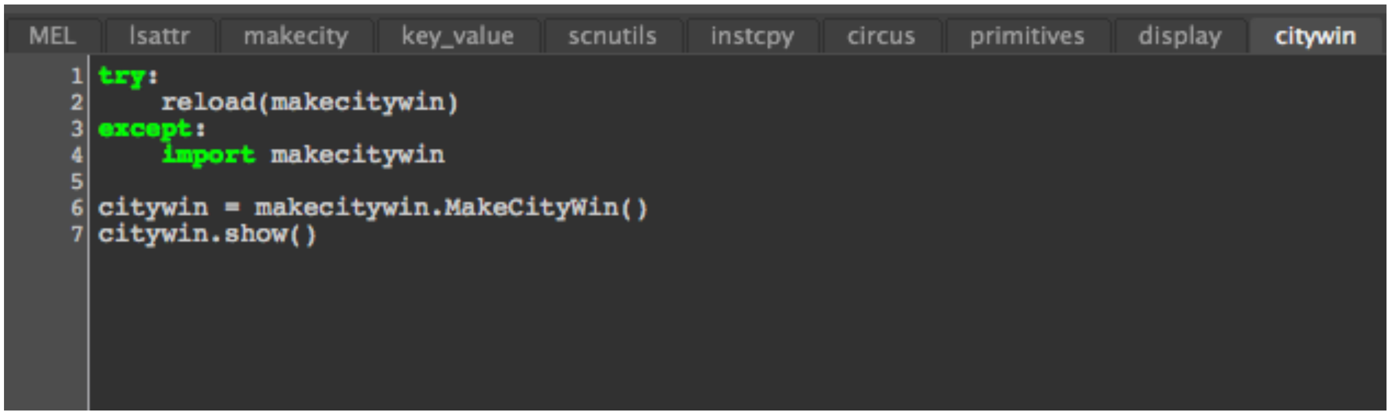
```python

import maya.cmds as cmds
import maya.OpenMaya as om

om.MGlobal.displayInfo("your text here")
om.MGlobal.displayWarning("your warning here")  # Yellow background
om.MGlobal.displayError("your error here")  # Red background

cmds.warning("makecity")

cmds.error("foobar")
```

MAKECITY



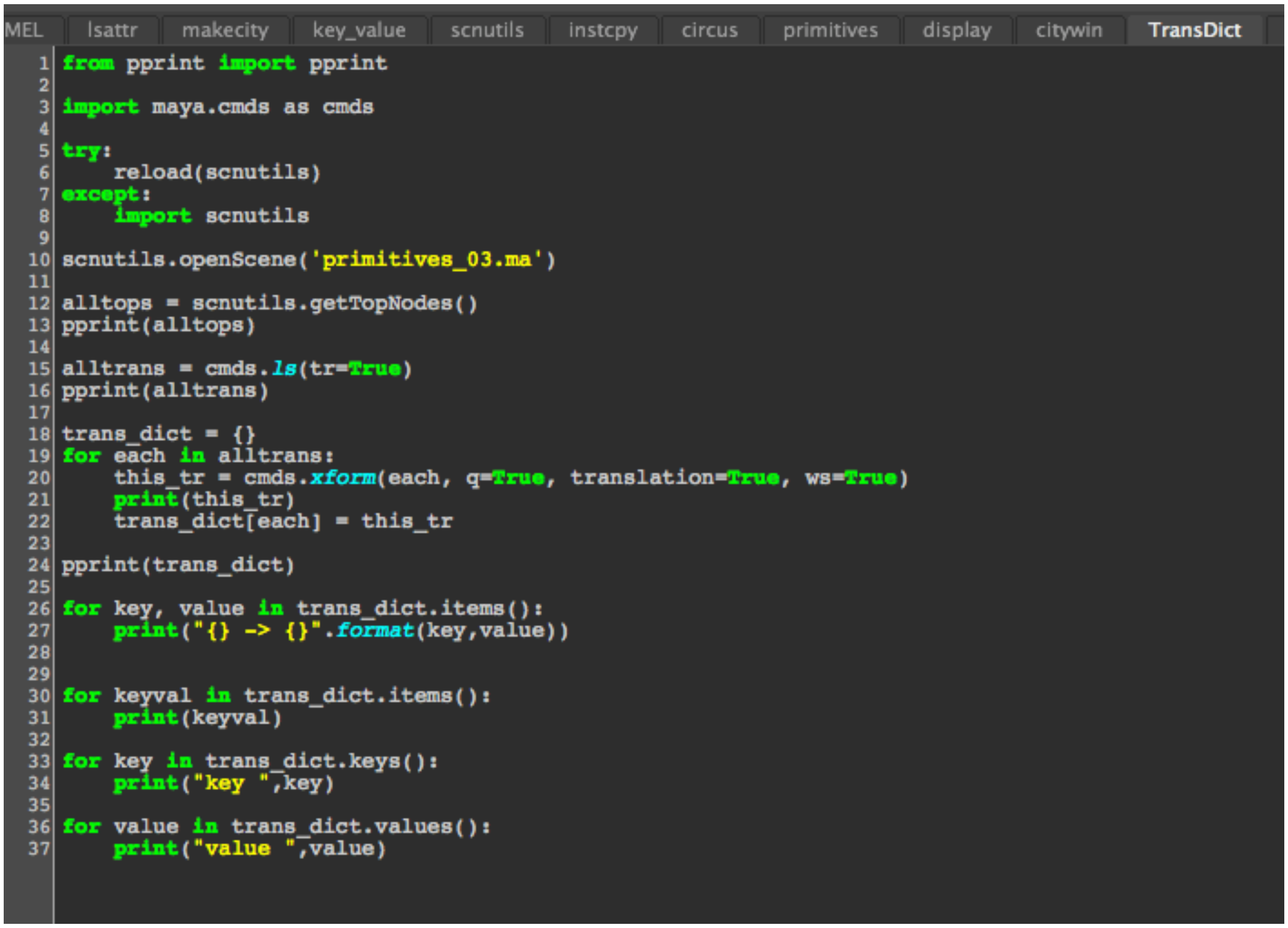
```python
import maya.cmds as cmds

try:
    reload(makecity)
except:
    import makecity

cmds.file(f=True,new=True)

newcube = makecity.mkcube("building_grp", "cube0")
newcity = makecity.copy2grid("building_grp")
makecity.randgeo(newcity)

print("newcube",newcube)
print("newcity",newcity)


############################################
#
#   makecity_sb
#
############################################

import makecity_sb as makecity

try:
    reload(makecity)
except:
    import makecity_sb as makecity

print(makecity.__file__)

cmds.file(f=True,new=True)
cmds.file("cityhill_v01.ma", iv=True, f=True, open=True)

makecity.mkbldgs("building_grp", "cube0", "cylinder0")
#makecity.mkbldgs("bldg_grp", cube_name="cube0", cylinder_name="cylinder0", selection=True)
#newcity = makecity.copy2grid("bldg_grp", stepx=6, stepz=6)
newcity = makecity.copy2vtx("building_grp", "city_hill")
makecity.randgeo(newcity, randsx=(0.5,1.5), randsy=(1.0,2.5), randrotate=True)

print(newcity)

print(dir(makecity))
print(makecity.mkcube.__doc__)
print(makecity.mkbldgs.__doc__)
```

CITYWIN



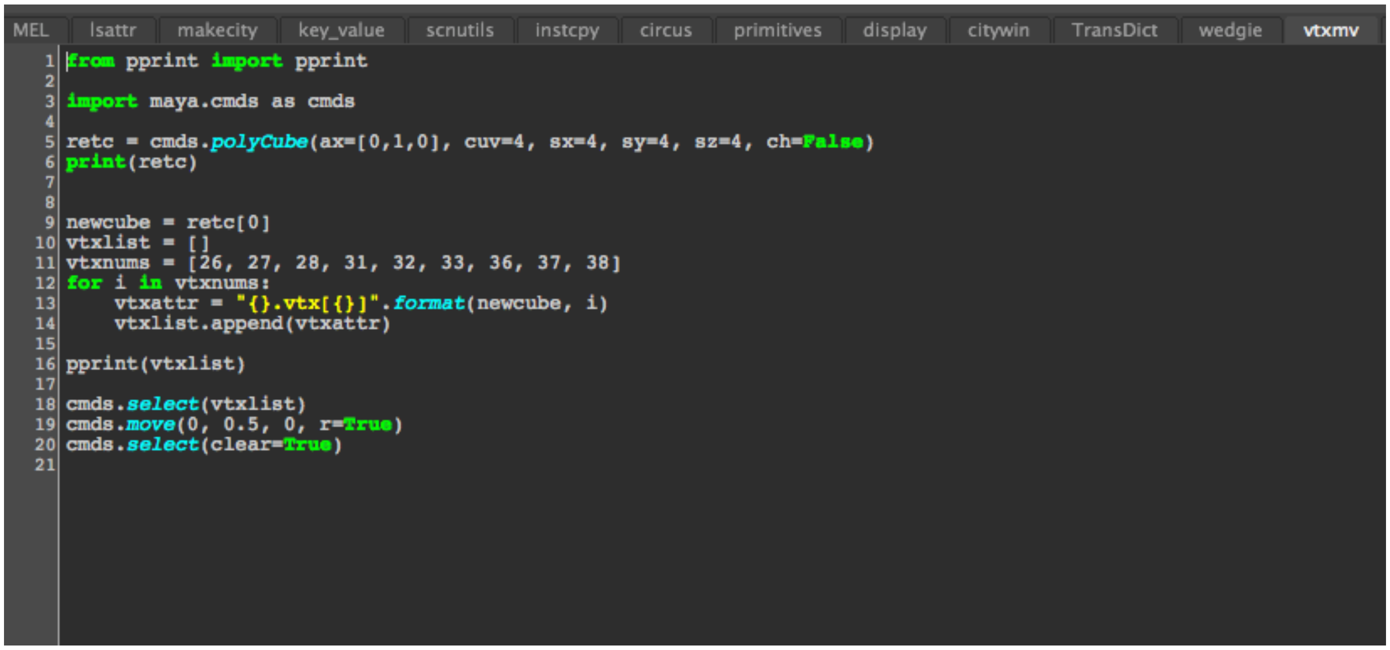
```python
try:
    reload(makecitywin)
except:
    import makecitywin

citywin = makecitywin.MakeCityWin()
citywin.show()
```

INSTCOPY



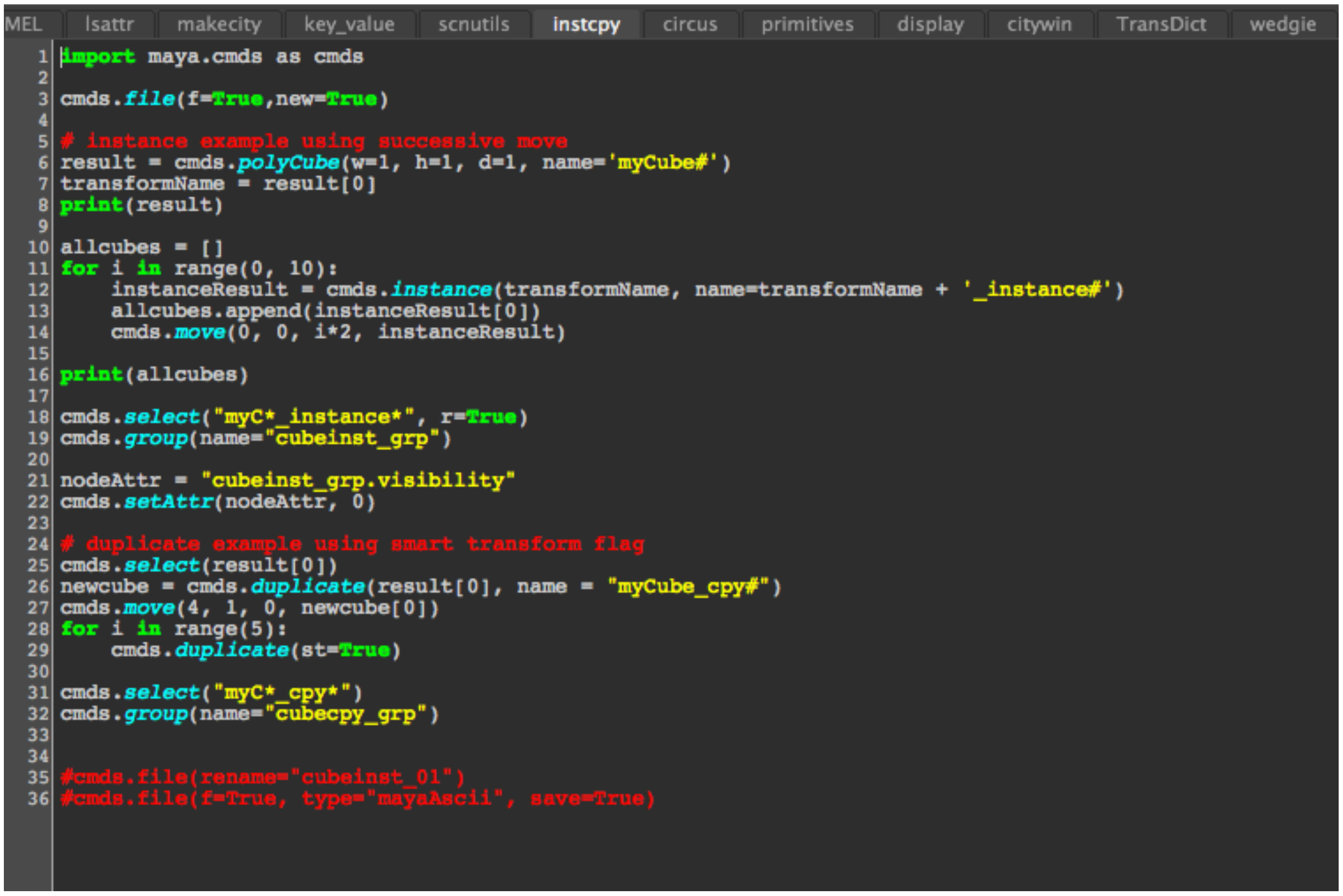
```
from pprint import pprint

import maya.cmds as cmds

try:
    reload(scnutils)
except:
    import scnutils

scnutils.openScene('primitives_03.ma')

alltops = scnutils.getTopNodes()
pprint(alltops)

alltrans = cmds.ls(tr=True)
pprint(alltrans)

trans_dict = {}
for each in alltrans:
    this_tr = cmds.xform(each, q=True, translation=True, ws=True)
    print(this_tr)
    trans_dict[each] = this_tr

pprint(trans_dict)

for key, value in trans_dict.items():
    print("{} -> {}".format(key,value))


for keyval in trans_dict.items():
    print(keyval)

for key in trans_dict.keys():
    print("key ",key)

for value in trans_dict.values():
    print("value ",value)
```

VTXMV



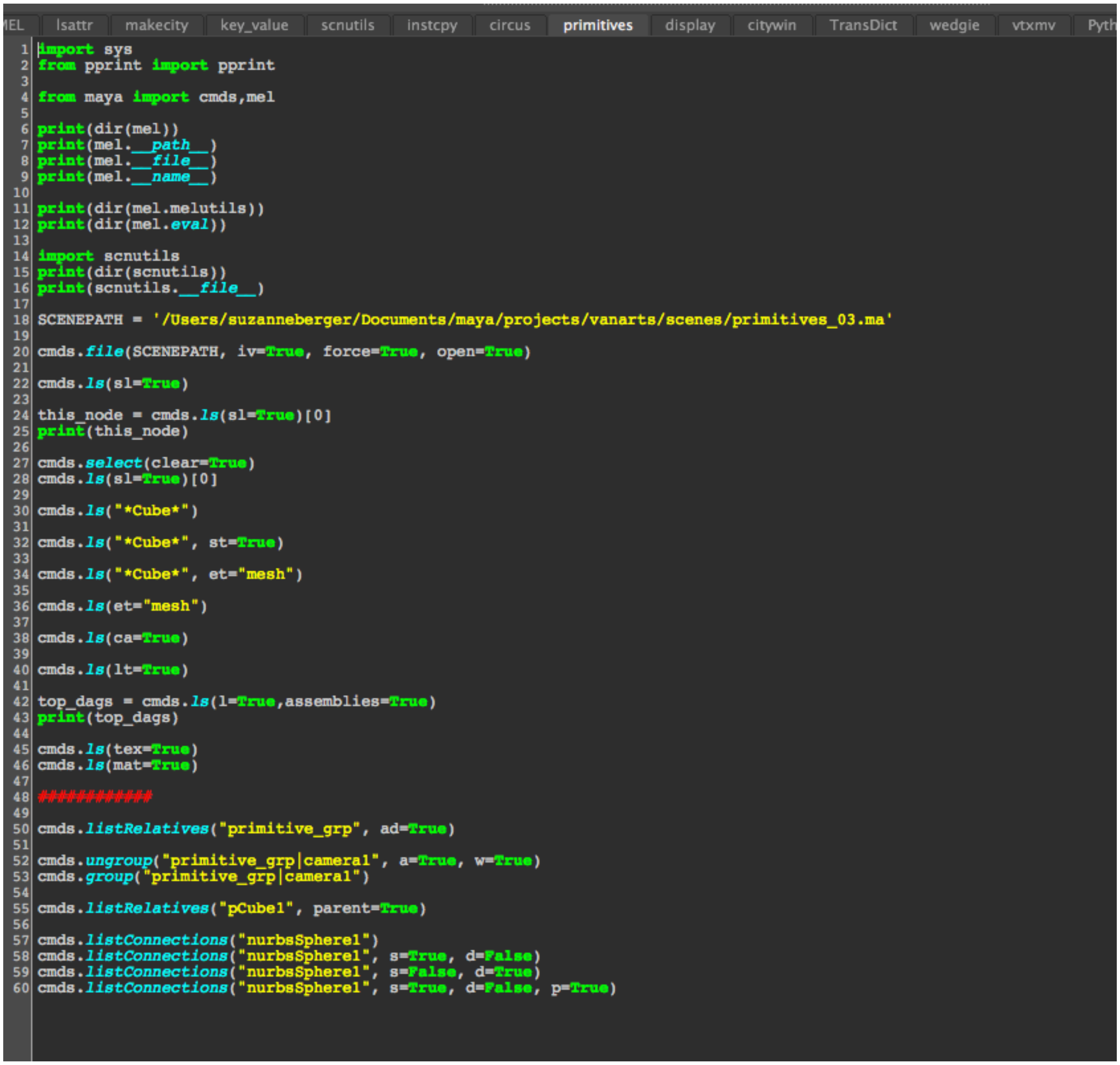
```
from pprint import pprint

import maya.cmds as cmds

retc = cmds.polyCube(ax=[0,1,0], cuv=4, sx=4, sy=4, sz=4, ch=False)
print(retc)


newcube = retc[0]
vtxlist = []
vtxnums = [26, 27, 28, 31, 32, 33, 36, 37, 38]
for i in vtxnums:
    vtxattr = "{}.vtx[{}]".format(newcube, i)
    vtxlist.append(vtxattr)

pprint(vtxlist)

cmds.select(vtxlist)
cmds.move(0, 0.5, 0, r=True)
cmds.select(clear=True)

```

INSTCPY



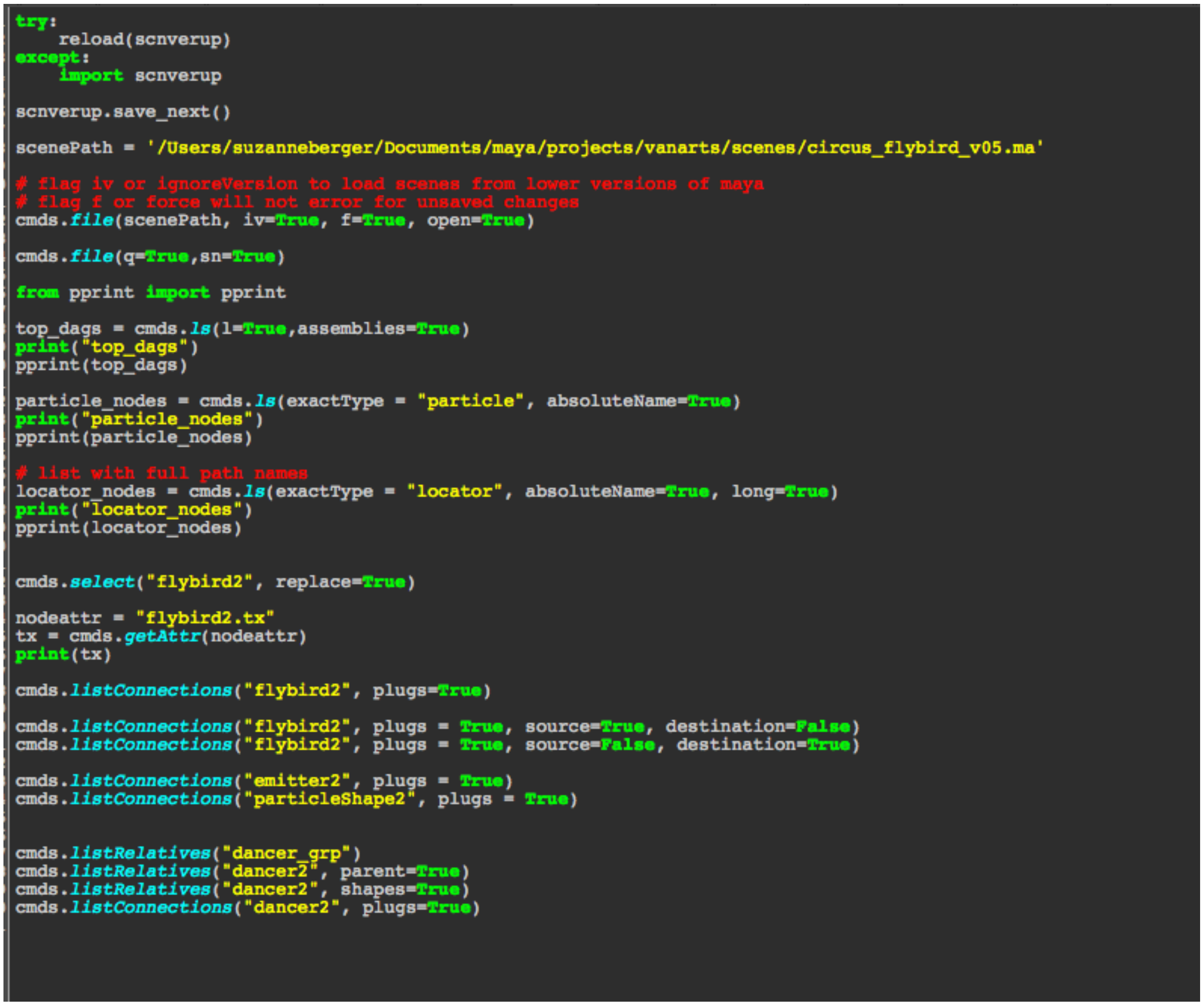
```python
import maya.cmds as cmds

cmds.file(f=True,new=True)

# instance example using successive move
result = cmds.polyCube(w=1, h=1, d=1, name='myCube#')
transformName = result[0]
print(result)

allcubes = []
for i in range(0, 10):
    instanceResult = cmds.instance(transformName, name=transformName + '_instance#')
    allcubes.append(instanceResult[0])
    cmds.move(0, 0, i*2, instanceResult)

print(allcubes)

cmds.select("myC*_instance*", r=True)
cmds.group(name="cubeinst_grp")

nodeAttr = "cubeinst_grp.visibility"
cmds.setAttr(nodeAttr, 0)

# duplicate example using smart transform flag
cmds.select(result[0])
newcube = cmds.duplicate(result[0], name = "myCube_cpy#")
cmds.move(4, 1, 0, newcube[0])
for i in range(5):
    cmds.duplicate(st=True)

cmds.select("myC*_cpy*")
cmds.group(name="cubecpy_grp")


#cmds.file(rename="cubeinst_01")
#cmds.file(f=True, type="mayaAscii", save=True)
```

PRIMITIVES



```python
import sys
from pprint import pprint

from maya import cmds,mel

print(dir(mel))
print(mel.__path__)
print(mel.__file__)
print(mel.__name__)

print(dir(mel.melutils))
print(dir(mel.eval))

import scnutils
print(dir(scnutils))
print(scnutils.__file__)

SCENEPATH = '/Users/suzanneberger/Documents/maya/projects/vanarts/scenes/primitives_03.ma'

cmds.file(SCENEPATH, iv=True, force=True, open=True)

cmds.ls(sl=True)

this_node = cmds.ls(sl=True)[0]
print(this_node)

cmds.select(clear=True)
cmds.ls(sl=True)[0]

cmds.ls("*Cube*")

cmds.ls("*Cube*", st=True)

cmds.ls("*Cube*", et="mesh")

cmds.ls(et="mesh")

cmds.ls(ca=True)

cmds.ls(lt=True)

top_dags = cmds.ls(l=True,assemblies=True)
print(top_dags)

cmds.ls(tex=True)
cmds.ls(mat=True)

############

cmds.listRelatives("primitive_grp", ad=True)

cmds.ungroup("primitive_grp|camera1", a=True, w=True)
cmds.group("primitive_grp|camera1")

cmds.listRelatives("pCube1", parent=True)

cmds.listConnections("nurbsSphere1")
cmds.listConnections("nurbsSphere1", s=True, d=False)
cmds.listConnections("nurbsSphere1", s=False, d=True)
cmds.listConnections("nurbsSphere1", s=True, d=False, p=True)
```

CIRCUS



```
try:
    reload(scnverup)
except:
    import scnverup

scnverup.save_next()

scenePath = '/Users/suzanneberger/Documents/maya/projects/vanarts/scenes/circus_flybird_v05.ma'

# flag iv or ignoreVersion to load scenes from lower versions of maya
# flag f or force will not error for unsaved changes
cmds.file(scenePath, iv=True, f=True, open=True)

cmds.file(q=True,sn=True)

from pprint import pprint

top_dags = cmds.ls(l=True,assemblies=True)
print("top_dags")
pprint(top_dags)

particle_nodes = cmds.ls(exactType = "particle", absoluteName=True)
print("particle_nodes")
pprint(particle_nodes)

# list with full path names
locator_nodes = cmds.ls(exactType = "locator", absoluteName=True, long=True)
print("locator_nodes")
pprint(locator_nodes)


cmds.select("flybird2", replace=True)

nodeattr = "flybird2.tx"
tx = cmds.getAttr(nodeattr)
print(tx)

cmds.listConnections("flybird2", plugs=True)

cmds.listConnections("flybird2", plugs = True, source=True, destination=False)
cmds.listConnections("flybird2", plugs = True, source=False, destination=True)

cmds.listConnections("emitter2", plugs = True)
cmds.listConnections("particleShape2", plugs = True)


cmds.listRelatives("dancer_grp")
cmds.listRelatives("dancer2", parent=True)
cmds.listRelatives("dancer2", shapes=True)
cmds.listConnections("dancer2", plugs=True)
```