

```

import matplotlib as mpl

from math import pi
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import matplotlib.patches as mpatches
from matplotlib.collections import PatchCollection
import numpy as np
from numpy.random import rand
from pylab import pcolor, show, colorbar, xticks, yticks, pcolormesh, imshow
from random import gauss
from matplotlib.ticker import PercentFormatter
from mpl_toolkits.mplot3d import Axes3D

from mpl_toolkits.mplot3d.art3d import Poly3DCollection, Line3DCollection
import itertools

#In: 2D-arraylike "V" of coins for your patches
#      (each line is a coin and as three column)
#      2D-arraylike "Surfaces" coins defining surfaces, each lines
#      contains coins indices that define a surface
#      (indices start from 0 and are the same order
#      as are the coins in the list )
#      A list of color or none (default), default implies all blue,
#      the list must be 1D list with a color per surface
#Out: Axes and figure objects, a figure of the patches will also be
#      displayed at the next plt.show()
def patches3d(V, Surfaces, fc=None):
    if not fc:
        fc='tab:blue'
    V=np.array(V)
    Surfaces=np.array(Surfaces, dtype=object)
    fig_patch = plt.figure()
    ax_patch = fig_patch.add_subplot(111, projection='3d')
    ax_patch.scatter3D(V[:, 0], V[:, 1], V[:, 2])
    verts=[V[surf] for surf in Surfaces]
    ax_patch.add_collection3d(Poly3DCollection(verts, facecolors=fc,
                                              linewidths=1, edgecolors='r', alpha=.25))
    return ax_patch, fig_patch

#e.g. pyramid (square base)
#definition of coins
v1=[0,0,1]
v2=[1,1,-1]
v3=[1,-1,-1]
v4=[-1,-1,-1]
v5=[-1,1,-1]

v =[v1,v2,v3,v4,v5]

S=[[0,1,2,], [0,2,3], [0,3,4], [0,1,4], [1,2,3,4]] #surfaces definition

# plot sides
ax, fig= patches3d(v, S, fc=['b', 'y', 'g', 'orange', 'violet'])
ax.set_xlim(-1,1)
ax.set_ylim(-1,1)
ax.set_zlim(-1,1)

```

```
plt.show()
```

