```
# Library support for 2D LED strip arrays
# By Brygg Ullmer, Sida Dai, and Mitali Bhosekar, Clemson University
# Begun 2022-04-13

import os

try: #https://github.com/v923z/micropython-ulab
    from ulab import numpy as np
    from ulab import scipy as sp

except ImportError:
    import numpy as np
    import scipy.special

#a = numpy.full((3,3), '.', dtype=numpy.char)

############## numpy|ulab 2D character array #############

class np2DCharArr: #
  arr         = None
  shape       = None
  defaultChar = None
  verbose     = False
  colormap    = {'O': (255, 128, 0), 'P': (128, 0, 128),
    'R': (255, 0, 0), 'W': (255, 255, 255), 'G': (0, 255, 0),
    'B': (0, 0, 255), '.': (0, 0, 0)}

  ############## constructor ##############

  def __init__(self, shape=(3,3), defaultChar = '.'):
    self.defaultChar = defaultChar; self.shape = shape
    dcint = ord(defaultChar)
    self.arr = np.full(shape, dcint, dtype=np.uint8)
    # self.buildColormap()

  def getShape(self): return self.shape

  ############## get idx color ##############

  def getIdxColor(self, posIdx):
    i, j = posIdx
    idxColInt = self.arr[i][j]
    idxCol = '%c' % idxColInt
    #print("getIdxColor:", posIdx, idxCol)
    if idxCol in self.colormap:
      color = self.colormap[idxCol]; return color
    return None

  ############## fill row ##############

  def fillRow(self, whichRow, whichChar):
    wcint = ord(whichChar)
    maxIdx = self.shape[0]
    for i in range(maxIdx): self.arr[whichRow][i] = wcint

  ############## fill col ##############

  def fillCol(self, whichCol, whichChar):
    wcint = ord(whichChar)
    maxIdx = self.shape[1]
    for i in range(maxIdx): self.arr[i][whichCol] = wcint

  ############## print ##############

  def print(self):
    #https://numpy.org/doc/stable/reference/generated/numpy.array2string.html
    s = np.array2string(self.arr, separator='', \
        formatter={'int':lambda x: chr(x)})
    print(s)

  ############## row to string ##############

  def row2str(self, whichRow):
    result = ''
    numCols = self.shape[0]
    for j in range(numCols):
      idxColInt = self.arr[whichRow][j]
      idxCol = '%c' % idxColInt
      result += idxCol
    return result

  ############## printCSV ##############

  def genCSV(self):
    resultL = []
    numRows = self.shape[1]
    for j in range(numRows):
      resultL.append(self.row2str(j))
    result = ','.join(resultL)
    return result

  ############## genColWeave ##############

  def genColWeave(self): #generate column-wise "weave," initially for LED chain
    numCols = self.shape[0]
    numRows = self.shape[1]

    result = ''
    for j in range(numCols):
      for i in range(numRows):
        if j % 2 == 0: i2 = i
        else:          i2 = numRows-i-1
        if self.verbose: print("%i,%i" % (i2, j))

        val = self.arr[i2][j]
        c = '%c' % val
        result += c
    return result

  ############## map color string to integer ##############

  def mapColor(self, colorCh): #convert color character to RGB triple
    if colorCh in self.colormap:
      result = self.colormap[colorCh]; return result
    print("np2DCharArr mapColor: no code for %c found in colormap" % colorCh)
    return None

  ############## map color string to integer ##############

  def mapColorStr2Int(self, targStr): #convert string of color codes to list of RGB triples
    tslen  = len(targStr)
    result = []

    for i in range(tslen):
      ch = targStr[i]
      co = self.mapColor(ch)
```

```
      result.append(co)

    return result

  ############## map color string to integer ##############

  def clearScr(self):
    try: os.clear() #Linux-flavors
    except: pass

    try: os.cls() #Windows-flavors
    except: pass

############## main #############

if __name__ == '__main__':
  na = np2DCharArr((4,4))
  na.fillRow(1, 'P')
  na.fillCol(1, 'O')
  na.fillRow(3, 'P')
  na.print()
  weaveCh = na.genColWeave()
  weaveCo = na.mapColorStr2Int(weaveCh)
  print("\n" + weaveCh)
  print("\n" + str(weaveCo))

### end ###
```

```
# Library support for 2D LED strip arrays
# By Brygg Ullmer, Sida Dai, and Mitali Bhosekar, Clemson University
# Begun 2022-04-13

from ledArray import *

############## numpy|ulab 2D character array #############

class ledArrayViz:
  ledArrayHandle   = None
  rectList         = None
  vertList         = None
  rectDim          = (3,3)
  basePos          = (50, 50)
  dx, dy           = 30, 30
  defaultRectColor = (50, 50, 50)
  vert2idx, idx2vert, idx2rect = [None]*3

  ############## constructor ##############

  def __init__(self, ledArray):
    self.ledArrayHandle = ledArray
    self.constructGridViz()

  ############## construct rects ##############

  def constructGridViz(self):
    print("ledArrayViz constructor called")

    rows, cols = self.ledArrayHandle.getShape()
    self.vertList = []; self.vert2idx = {}; self.idx2vert = {}

    cy = self.basePos[1]

    for i in range(rows):
      cx = self.basePos[0]
      for j in range(cols):
        vert = (cx, cy)
        self.vertList.append(vert)
        idx = (i, j)
        self.vert2idx[vert] = idx
        self.idx2vert[idx]  = vert
        cx += self.dx
      cy += self.dy

    self.constructRects()

  ############## construct rects ##############

  def constructRects(self):
    self.rectList = []
    #print("constructRects:", self.vertList)
    self.idx2rect = {}

    for vert in self.vertList:
      rect = Rect(vert, self.rectDim)
      idx  = self.vert2idx[vert]
      #print(idx, vert)
      self.idx2rect[idx] = rect

  ############## draw rects ##############

  def drawRects(self, screen):
    #print("drawRects:", self.vertList)
    for vert in self.vertList:
      idx  = self.vert2idx[vert]
      #print("dr1:", vert, idx)
      rect = self.idx2rect[idx]
      color = self.ledArrayHandle.getIdxColor(idx)
      #print("dr2:", idx, color)
      if color == (0, 0, 0) or color == None:
        color = self.defaultRectColor

      screen.draw.filled_rect(rect, color)

  ############## pgzero draw #############

  def draw(self, screen):
    self.drawRects(screen)

### end ###
```

```
# Library support for 2D LED strip arrays
# By Brygg Ullmer, Sida Dai, and Mitali Bhosekar, Clemson University
# Begun 2022-04-13

import os
from ledArray import *

na = np2DCharArr((13,13))

for i in range(4):
  na.fillRow(i*4, 'P')
  na.fillCol(i*4, 'P')

na.clearScr()
na.print()
weaveCh = na.genColWeave()
weaveCo = na.mapColorStr2Int(weaveCh)
print("\n" + weaveCh)
print("\n" + str(weaveCo))

### end ###
```

```
# Library support for 2D LED strip arrays
# By Brygg Ullmer, Sida Dai, and Mitali Bhosekar, Clemson University
# Begun 2022-04-13

from ledArray    import *
from ledArrayViz import *

print("main called")

na = np2DCharArr((8,8))
na.fillRow(1, 'P')
na.fillCol(1, 'O')
na.fillRow(3, 'P')
na.print()
print(na.genCSV())

global lav
lav = ledArrayViz(na)


def draw():
   global screen, lav
   try: lav.draw(screen)
   except: pass

### end ###
```