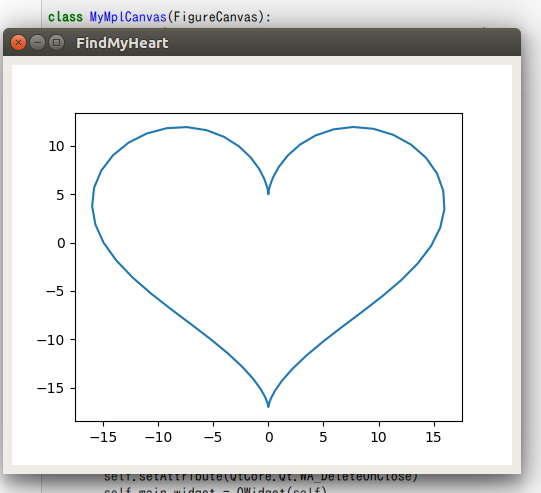
**FindMyHeart**

กำหนดโค้ดโปรแกรมสร้างกราฟมาให้ โดยโค้ดดังกล่าวเป็นโค้ดที่วาดกราฟด้วยการกำหนดค่าในแกน X, Y และช่วงที่จะวาดภาพ โดยโค้ดนี้เมื่อรันดูจะแสดงภาพเป็นวงกลม งานของคุณคือการทำโค้ดนี้ให้เปลี่ยนจากกราฟวงกลมเป็นรูปหัวใจให้ได้ โดยการปรับแก้โค้ดส่วน MyStaticMplCanvas ให้มีการสร้างกราฟเป็นดังตัวอย่าง



โดยวิธีการสร้างกราฟมีสามขั้นตอนดังนี้

1. กำหนดช่วงระยะขอบเขตการสร้างกราฟผ่านทางคำสั่ง range (อ่านวิธีใช้งานจาก <https://docs.scipy.org/doc/numpy/reference/generated/numpy.arange.html>)
2. กำหนดค่า x, y ที่ใช้ในสมการ
3. สร้างกราฟผ่านทางคำสั่ง axes.plot (อ่านวิธีใช้งานจาก <https://matplotlib.org/api/_as_gen/matplotlib.axes.Axes.plot.html#matplotlib.axes.Axes.plot>)

ส่วนสมการสร้างหัวใจสามารถหาได้ตามเว็บกูเกิ้ล โดยอาจใช้คำค้นหาให้เหมาะสมเช่น “heart function” เป็นต้น

**กำหนดให้การส่งนั้น ต้องใช้สมการที่ห้ามซ้ำกันกับผู้ส่งก่อนหน้าจึงจะยอมรับว่าผ่าน**

**โค้ด**

|  |
| --- |
| import sys import matplotlib matplotlib.use("Qt5Agg") from PyQt5 import QtCore from PyQt5.QtWidgets import QApplication, QMainWindow, QSizePolicy, QVBoxLayout, QWidget from numpy import arange, sin, cos,pi from matplotlib.backends.backend\_qt5agg import FigureCanvasQTAgg as FigureCanvas from matplotlib.figure import Figure  #ส่วนกำหนดข้อมูลสร้างภาพกราฟ class MyMplCanvas(FigureCanvas):  def \_\_init\_\_(self, parent=None, width=5, height=4, dpi=100):  fig = Figure(figsize=(width, height), dpi=dpi)  self.axes = fig.add\_subplot(111)  self.compute\_initial\_figure()  FigureCanvas.\_\_init\_\_(self, fig)  self.setParent(parent)  FigureCanvas.setSizePolicy(self,  QSizePolicy.Expanding,  QSizePolicy.Expanding)  FigureCanvas.updateGeometry(self)   def compute\_initial\_figure(self):  pass  #ส่วนที่ต้องเขียนโค้ด (นิยามสมการหัวใจและวาดมัน) class MyStaticMplCanvas(MyMplCanvas):  def compute\_initial\_figure(self): #แก้ให้เป็นรูปหัวใจ  t = arange(0,2\*pi+0.1, 0.1) #ช่วงในการสร้างวงกลมตั้งแต่ 0 ถึง 2Pi โดยวาดทุก ๆ 0.1 เรเดียน  x = cos(t)  y = sin(t)  self.axes.plot(x,y)   #ส่วนสร้างหน้าต่าง class MainWindow(QMainWindow):  def \_\_init\_\_(self):  QMainWindow.\_\_init\_\_(self)  self.setAttribute(QtCore.Qt.WA\_DeleteOnClose)  self.main\_widget = QWidget(self)  l = QVBoxLayout(self.main\_widget)  Heart = MyStaticMplCanvas(self.main\_widget, width=5, height=5, dpi=100) #กำหนดขนาดกราฟ  l.addWidget(Heart)  self.main\_widget.setFocus()  self.setCentralWidget(self.main\_widget)  #ส่วนเริ่มโปรแกรม if \_\_name\_\_ == '\_\_main\_\_':  app = QApplication(sys.argv)  main\_window = MainWindow()  main\_window.setWindowTitle("FindMyHeart")  main\_window.show()  app.exec\_() |

ในงานนี้ใช้ความรู้เพิ่มเติมจากไพธอนพื้นฐาน 3 อย่าง ได้แก่

PyQT ซึ่งเป็น package ที่รองรับการสร้างหน้าจอ

numpy เป็น package ที่บรรจุชุดคำสั่งที่เกี่ยวข้องกับการคำนวนสมการ

matplotlib เป็น package ที่บรรจุชุดคำสั่งสำหรับการเขียนกราฟ

โดยหากผู้ที่ต้องการศึกษาเพิ่มเติมด้วยตนเองสามารถเข้าไปอ่านข้อมูลเบื่องต้นได้จากที่ดังต่อไปนี้

PyQT5 : <http://zetcode.com/gui/pyqt5/>

numpy : <http://cs231n.github.io/python-numpy-tutorial/>

matplotlib : <https://www.labri.fr/perso/nrougier/teaching/matplotlib/>

-------------------------------------------------------------------------------------------------------------------------------