

```

# ===== Challenge 1: Can a computer create Art?

import matplotlib.pyplot as plt
import numpy as np

# 1. Plot a circle.
t = np.linspace(0,2*np.pi) # linearly space t between 0 and 2*pi for a full
circle.
x = np.cos(t)
y = np.sin(t)
plt.plot(x,y)
plt.gca().set_aspect('equal') # make the aspect ratio equal so it appears 'true'
plt.show()

# 2. Plot Fermat's Spiral, as in https://en.wikipedia.org/wiki/Fermat%27s_spiral
length = 40 # change the length to change the number of times the curve spirals
t = np.linspace(0,length,10000)
x0 = (t**0.5)*np.cos(t)
y0 = (t**0.5)*np.sin(t)
x1 = (-t**0.5)*np.cos(t)
y1 = (-t**0.5)*np.sin(t)
x = np.concatenate((x0[::-1],x1)) # stick the two vectors together.
y = np.concatenate((y0[::-1],y1))
plt.plot(x,y)
plt.gca().set_aspect('equal')
plt.show()

# 3. Plot the Butterfly Curve, as in
t = np.linspace(0,12*np.pi,10000) # the curve is specified from 0 to 12*pi.
x = np.sin(t)*( np.exp(np.cos(t)) - 2*np.cos(4*t) - (np.sin(t/12))**5 )
y = np.cos(t)*( np.exp(np.cos(t)) - 2*np.cos(4*t) - (np.sin(t/12))**5 )
plt.plot(x,y)
plt.gca().set_aspect('equal')
plt.show()

# Extension: using colormaps on the butterfly curve (and, LineCollection for
faster plotting)
from matplotlib import cm
from matplotlib.collections import LineCollection
r = np.sqrt(x**2 + y**2) # find the radius, then normalise.
r = r/max(r)

fig,ax = plt.subplots()
segments = []
for i in range(len(x)-1):
    segments.append([(x[i],y[i]),(x[i+1],y[i+1])])

coll = LineCollection(segments, cmap=plt.cm.rainbow,linewidth=0.8)
coll.set_array(r)
ax.add_collection(coll)
ax.set_aspect('equal')
ax.autoscale_view()
ax.axis('off')
plt.show()

```