

Графика

Хашин С.И.

<http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html>

Ивановский университет

Простейшая графика

Иваново-2023

План

tkinter

matplotlib

savefig

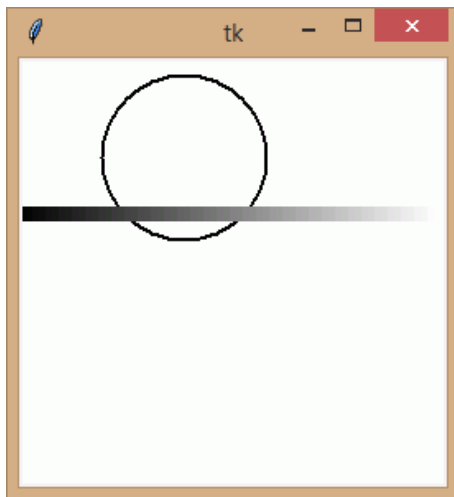
Pillow

tkinter

```
import tkinter as tk
def plot_00():
    mx = 256
    cGray = 1 + 256 + 256 * 256
    root = tk.Tk()
    c = tk.Canvas(root, width=mx, height=mx, bg='white')
    c.pack()
    c.create_oval(50, 10, 150, 110, width=2)
    for y in range(90, 99):
        for x in range(mx):
            c.create_line(x, y, x + 1, y, fill="#" +
                          "{:06x}".format((x) * cGray))
    root.mainloop()
```

Результат

Овал и полоска переменного цвета:



Графика без matplotlib

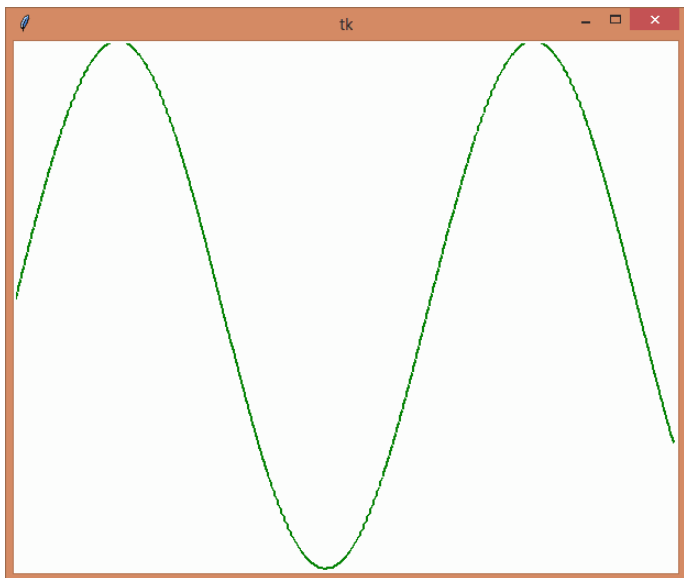
```
def plot(x, y, color, mx=600, my=480 ):
    root = tk.Tk()
    c = tk.Canvas(root, width=mx, height=my, bg='white')
    c.pack()
    minX, maxX = min(x), max(x)
    minY, maxY = min(y), max(y)
    coeffX = mx/(maxX-minX)
    coeffY = my/(maxY-minY)
    for i in range(len(x)-1):
        cx = int(coeffX*(x[i]-minX))
        cy = int(coeffY*(maxY-y[i]))
        cx1 = int(coeffX*(x[i+1]-minX))
        cy1 = int(coeffY*(maxY-y[i+1]))
        c.create_line(cx, cy, cx1, cy1, width=2, fill=color)
    root.mainloop()
```

Test tk_plot

Проверим работу:

```
mx, my = 640, 480
x = np.arange(mx) / 40.
y = np.sin(x)
plot(x, y, 'green', 480, 640)
```

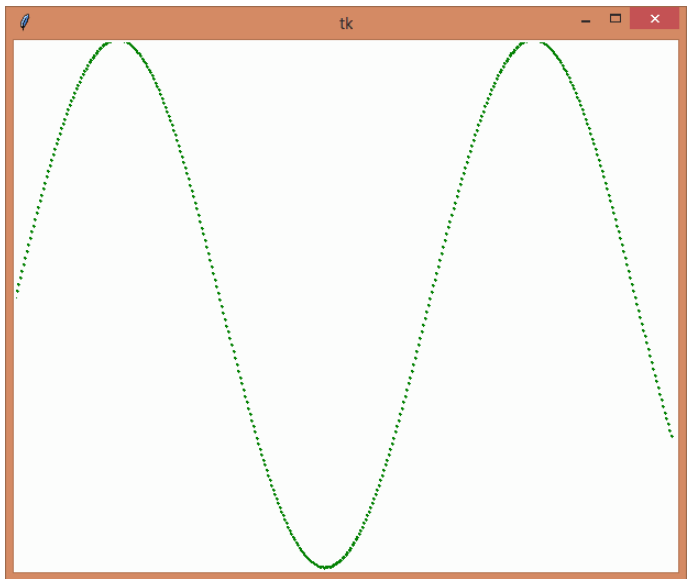
Результат



tk_scatter

```
def scatter(x, y, color, mx=600, my=480 ):
    root = tk.Tk()
    c = tk.Canvas(root, width=mx, height=my, bg='white')
    c.pack()
    minX, maxX = min(x), max(x)
    minY, maxY = min(y), max(y)
    coeffX = mx/(maxX-minX)
    coeffY = my/(maxY-minY)
    for i in range(len(x)-1):
        cx = int(coeffX*(x[i]-minX))
        cy = int(coeffY*(maxY-y[i]))
        c.create_line(cx-1, cy-1, cx+1, cy+1,
                      width=2, fill=color)
    root.mainloop()
```


Результат



matplotlib

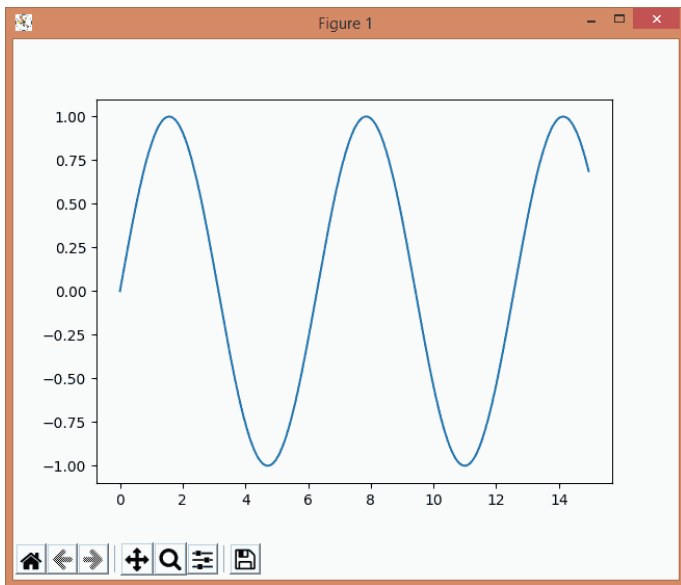
```
import matplotlib.pyplot as plt

def plot_01():
    N, my = 300, 300
    X = np.arange(300)/20 # от 0 до 15
    Y = np.sin(X)
    plt.plot(X, Y)
    #plt.scatter(X, Y, s=5, color='green')
    # s --- ПЛОЩАДЬ точки в пикселах
    #plt.show()
    plt.savefig('t:\plot01a.png', dpi = 100)
```

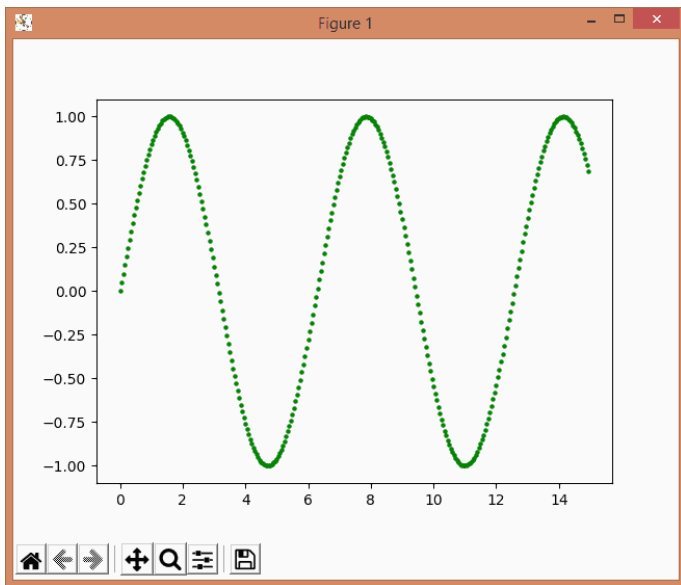
plot_01()

dpi — «dot per inch», точек на дюйм.

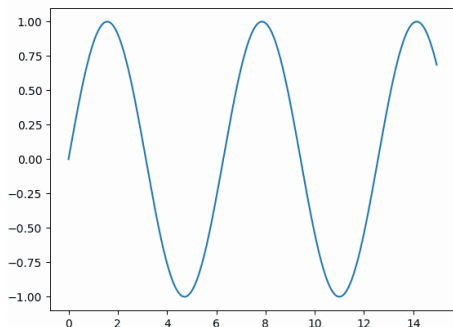
Результат plt.plot(), plt.show()



Результат plt.plot(), plt.scatter()



Результат plt.savefig()



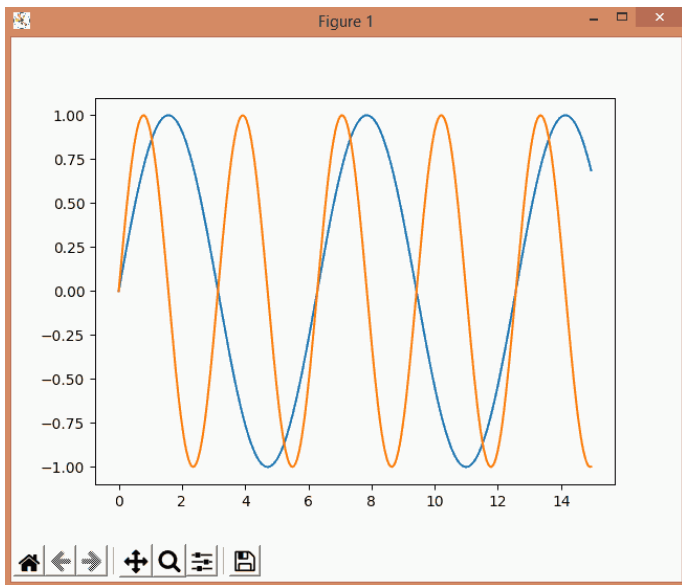
Файл получается большой, так как цвет = 'TrueColor', 3 байта на пиксел. Если хотите уменьшить, то, например, в IrfanView в меню:

Изображение → уменьшить глубину цвета до 256 → 16.

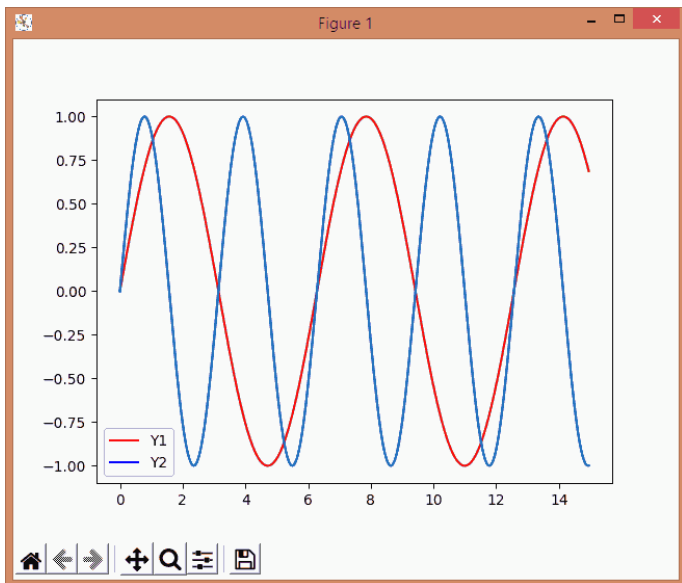
Несколько графиков

```
def plot_02():  
    N, my = 300, 300  
    X = np.arange(300)/20 # от 0 до 15  
    Y1 = np.sin(X)  
    Y2 = np.sin(2*X)  
    plt.plot(X, Y1, label='Y1', color='red')  
    plt.plot(X, Y2, label='Y2', color='blue')  
    plt.plot(X, Y2)  
    #plt.legend()  
    plt.show()
```

без легенды



с легендой



Модуль Pillow

Установка

```
pip install Pillow
```

Использование

```
import numpy as np
from PIL import Image #Pillow

img = Image.open('D:/00/best/12.jpg')
a = np.array(img)
print('a:', a.shape, type(a[0,0,0]))
> a: (2000, 3000, 3) <class 'numpy.uint8'>
```

Варианты:

```
img = Image.open('D:\\00\\best\\12.jpg')
img = Image.open('D:/00/best/12.jpg')
img = Image.open(r'D:\00\best\12.jpg')
```

save_png

```
def save_png(a, fname): # сохранить a в виде png-8
    a[a<0]=0
    a[a>255]=255
    a = a.round().astype(np.uint8)
    imgOut = Image.fromarray(a, 'L')
    imgOut.save(fname)
```

save_png3

Обычно размер RGB-массива $(3, m_y, m_x)$, то есть $a[0, :, :]$ — красная компонента цвета, $a[1, :, :]$ — зелёная, $a[2, :, :]$ — синяя. Но модуль PIL используем формат $(m_y, m_x, 3)$. Поэтому может потребоваться перестройка массива.

```
def save_png3(a, fname): # сохранить a в виде png-24
    a = a.astype(np.uint8)
    b = a
    if a.shape[0]==3:
        b = np.zeros((a.shape[1], a.shape[2], 3), dtype=np.uint8)
        for i in range(3):
            b[:, :, i] = a[i]
    imgOut = Image.fromarray(b, 'RGB')
    imgOut.save(fname)
```