

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
Троицкий авиационный технический колледж – филиал МГТУ ГА

Отделение «Программирование в компьютерных системах»

Методическое пособие № 4
по подготовке и проведению учебной практики
«Графические интерфейсы Python»

Тема «Библиотека Matplotlib»

Специальность 09.02.03

г. Троицк, 2017

Содержание

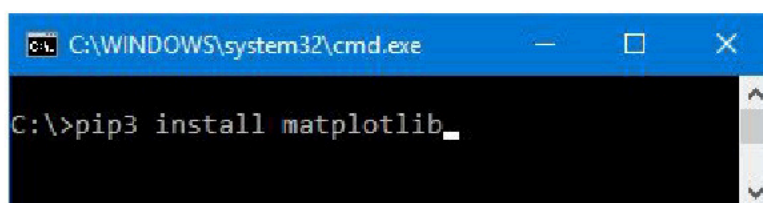
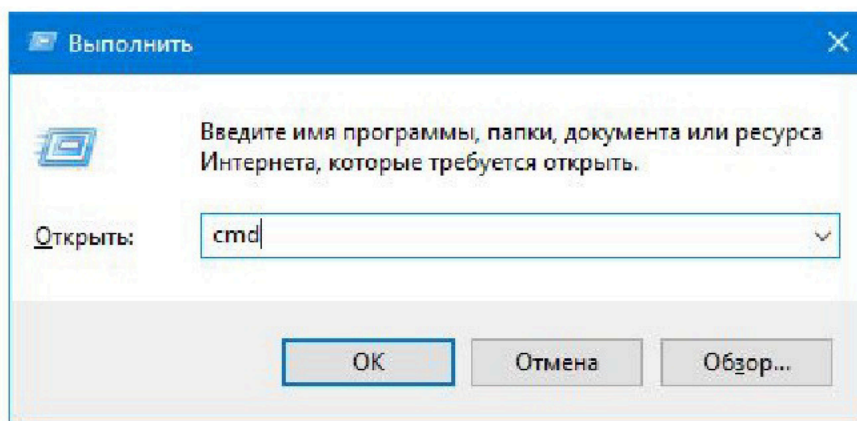
1. Установка библиотеки.
2. Тестирование библиотеки.
3. Управляющие элементы интерфейса окна
4. Построение графика функции.
5. Определение цвета элементов.
6. Определение стиля линии.
7. Определения стиля маркера.
8. Практическая работа.

1. Установка библиотеки.

Сила Python - в его внешних библиотеках, невероятно расширяющих возможности языка. Один из самых интересных и полезных проектов такого рода - библиотека научной графики [Matplotlib](#), разработанная американским нейробиологом Джоном Хантером..

Библиотека Matplotlib не поставляется в стандартном дистрибутиве python, её необходимо предварительно установить.

В зависимости от операционной системы существуют разные способы установки. Например, для ОС Windows со стандартной установкой python можно использовать консольный менеджер пакетов python, открыв консоль (Win+R → cmd) и выполнив команду (внимание! необходим доступ к интернет!): `pip3 install matplotlib`



2. Тестирование библиотеки.

Откройте командную строку Python IDLE.

Последовательно введите в нее три команды:

1. импорт модуля pyplot (и присваивание ей более короткого имени "plt")

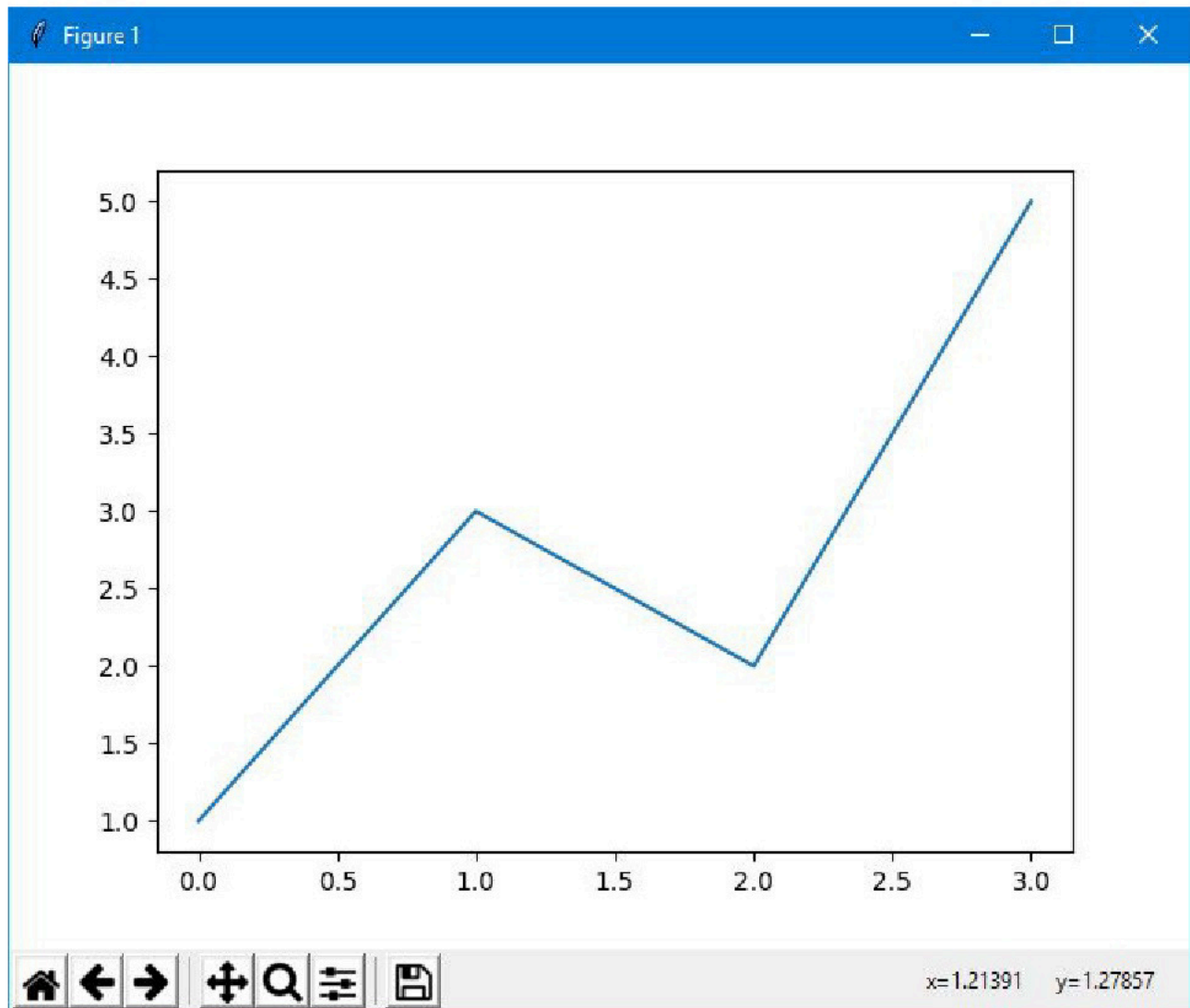
```
>>> import matplotlib.pyplot as plt
```
2. создание списка значений графика (состоящего из точек со следующими значениями Y: [1,3,2,5]. По умолчанию Matplotlib для координат X возьмет значения: [0,1,2,3])

```
>>> plt.plot([1,3,2,5])
```
3. отображение графика на экране

```
>>> plt.show()
```

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import matplotlib.pyplot as plt
>>> plt.plot([1,3,2,5])
[<matplotlib.lines.Line2D object at 0x0C62F750>]
>>> plt.show()
>>>
```

Должно появиться новое окно с графиком.



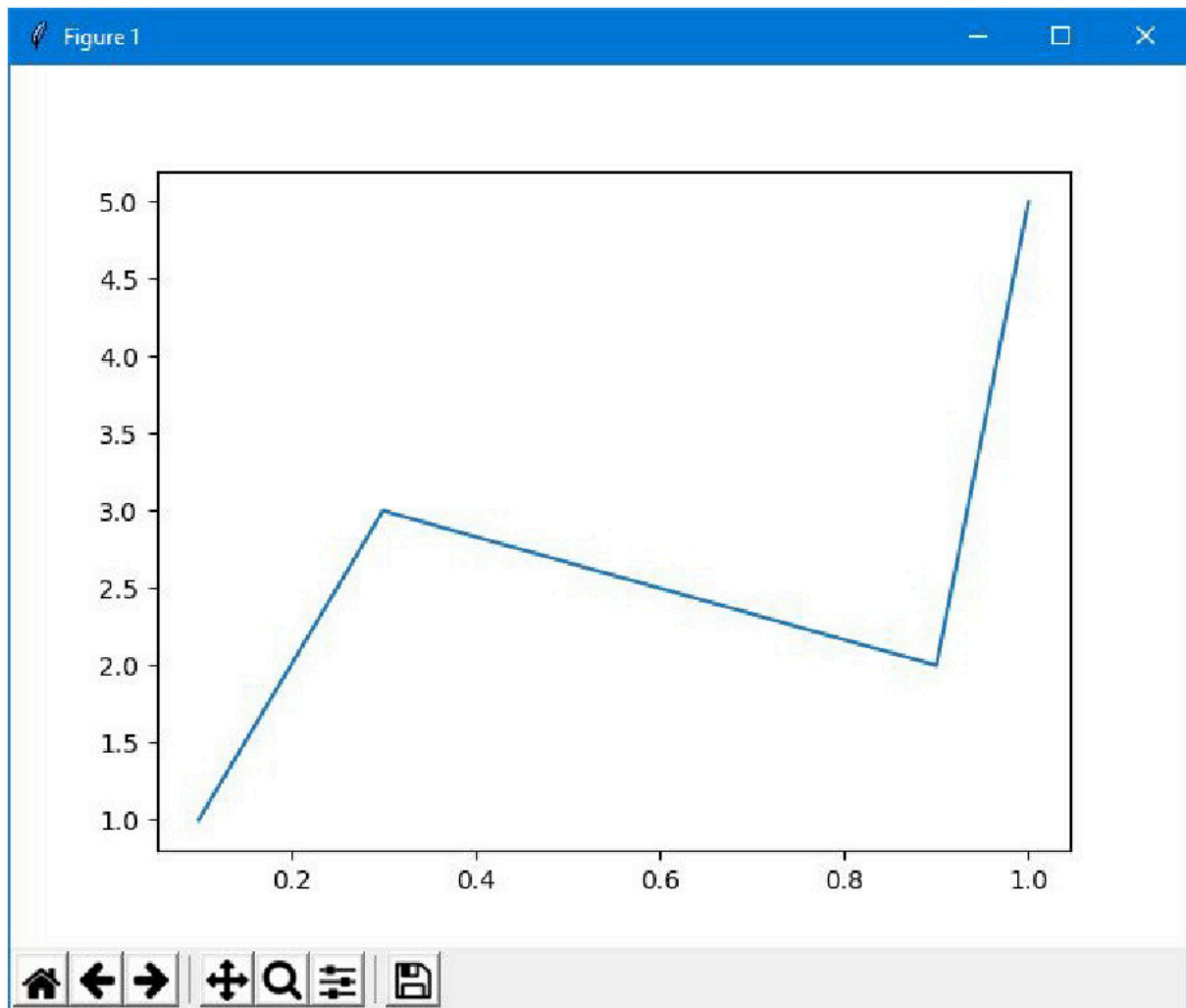
Основными параметрами метода `matplotlib.plot` являются два (или один, если список абсцисс определяется по умолчанию) списка:

1. координаты по оси OX (абсциссы) - значения переменной "x",
2. координаты по оси OY (ординаты) - значения вычисленной функции "y".

Например, выполнение следующих инструкций в командной строке Python IDLE:




```
>>> plt.plot([0.1,0.3,0.9,1], [1,3,2,5])
>>> plt.show()
```

отобразит в новом окне следующий график (сравните значения на оси абсцисс ОХ с предыдущим графиком):



3. Управляющие элементы интерфейса окна

Совместно с графиком в окне (нижней или верхней части) автоматически появляются кнопки управления отображением, которые по функциональности делятся на три группы:

1. Первая группа -  - *навигационные* кнопки. Кнопки со стрелками аналогичны кнопкам "Вперед" и "Назад" интернет-браузеров. Кнопка с "домиком" возвращает графику первоначальный вид, отменяя все сделанные модификации.
2. Вторая группа -  - кнопки, управляющие *параметрами отображения графиков*. Кнопка с "крестом" позволяет перемещать график внутри координатных осей без изменения масштаба. "Лупа" дает возможность разглядеть детали графиков.
3. В третью группу входит одна единственная кнопка - , с помощью которой можно *сохранить* получившийся график *в виде графического файла* с расширением *.jpg, *.tif, *.png, *.pdf и т.д. Такой рисунок можно потом вставить в документ Word или разместить на html-странице в Интернете.

4. Построение графика функции.

Построим график заданной функции $f(x)=\sin(x)/x$, определив цвет графика, стиль линии и добавив легенду (пояснение) с формулой этой функции в нотации TeX.

```
# -*- coding: UTF-8 -*-

# Импортируем стандартную, "внутреннюю" библиотеку math
import math

# Импортируем один из пакетов "внешней" библиотеки Matplotlib
import matplotlib.pyplot as plt

# Импортируем еще один пакет со вспомогательными функциями
from matplotlib import mlab

# Программное представление функции  $f(x)=\sin(x)/x$ 
def func(x):
    """
    sinc(x)
    """
    if x==0:
        return 1.0
    return math.sin(x)/x

# Интервал изменения переменной по оси X
xmin=-20.0
xmax=20.0

# Шаг между точками
dx=0.01

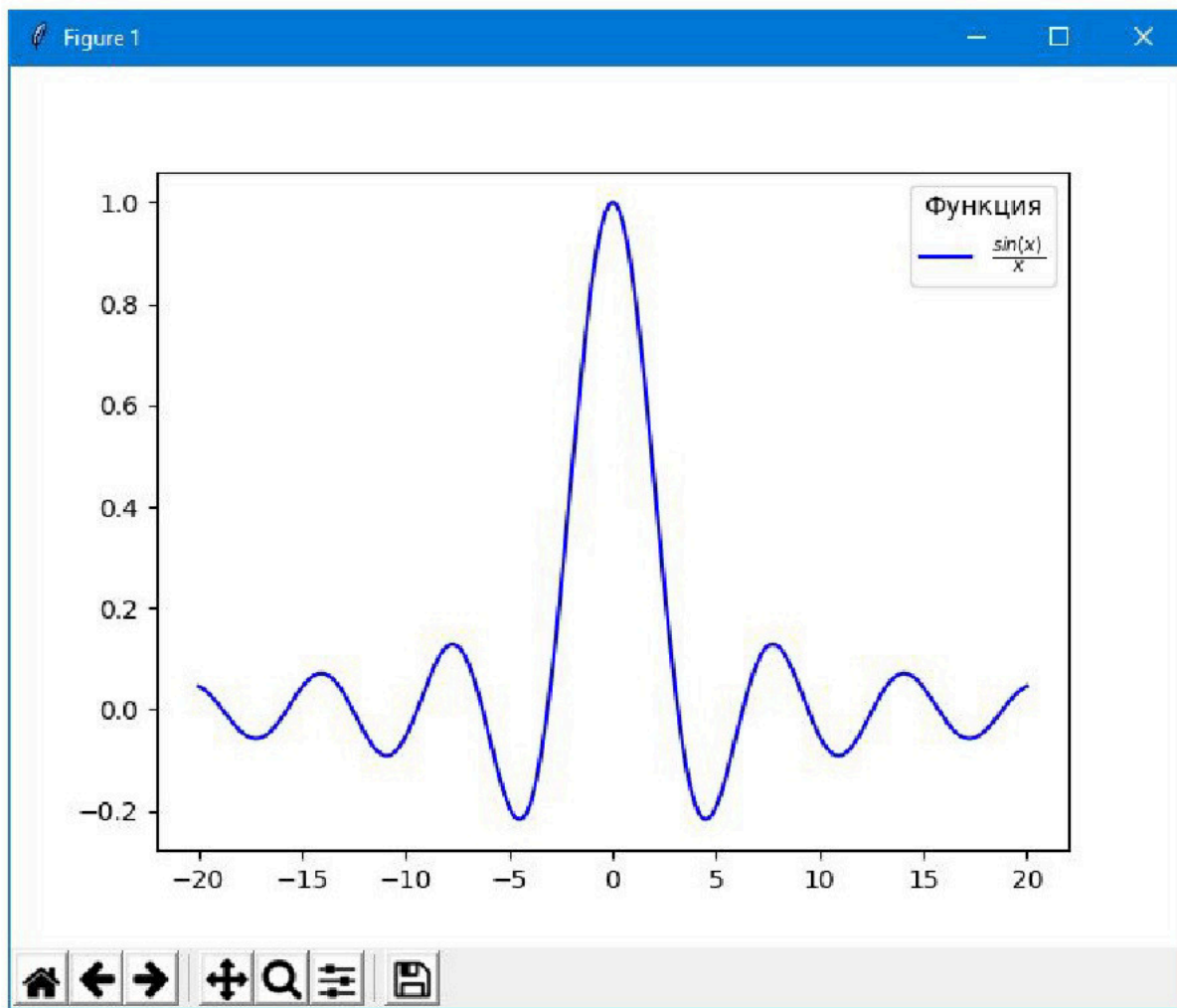
# Создадим список координат по оси X на отрезке [xmin; xmax],
# включая концы
xlist=mlab.frange(xmin, xmax, dx)

# Вычислим значение функции в заданных точках
ylist=[func(x) for x in xlist]

# Рисуем график заданного цвета (color) и стиля (linestyle),
# с определенной формой маркера (marker),
# помечающего отдельную точку
plt.plot(xlist, ylist, color='b', linestyle='-', marker='',
         label="$\\frac{\sin(x)}{x}$")

# Выводим легенду, описанную в предыдущей команде в значении
параметра label
plt.legend(title=u"Функция")

# Покажем окно с нарисованным графиком
plt.show()
```

5. Определение цвета элементов.

Цвет в Matplotlib можно определять следующими способами:

- с помощью однобуквенной строки (например, 'g'),
- с помощью словесного описания цвета (например, 'goldenrod'),
- с помощью указания компонент цвета в формате #RRGGBB (например, '#31D115'),
- с помощью указания компонент цвета в виде кортежа или списка трех чисел в диапазоне 0.0 - 1.0 (например, (0.5, 0.2, 0.3)),
- с помощью указания компонент цвета и альфа-канала в виде кортежа или списка четырех чисел в диапазоне 0.0 - 1.0 (например, (0.5, 0.2, 0.3, 0.8)),
- для задания серого цвета можно использовать число с плавающей точкой в диапазоне 0.0 - 1.0 (например, 0.3).

5.1. Один из способов задания цвета в Matplotlib - указание одиночного символа. Для это определены 8 обозначений стандартных цветов:

- 'b' - **синий** цвет
- 'g' - **зеленый** цвет
- 'r' - **красный** цвет
- 'c' - **голубой** цвет
- 'm' - **пурпурный** цвет
- 'y' - **желтый** цвет
- 'k' - **черный** цвет
- 'w' - **белый** цвет

В связи с ограниченностью цветового восприятия некоторыми людьми, разной возможностью компьютерного оборудования рекомендуется использовать эту палитру..

5.2. Более расширенный способ задания цвета в Matplotlib - с помощью строки, описывающей цвет полным английским словом. Таких именованных цветов намного больше, чем односимвольных. В документации есть хороший [пример](#), генерирующий таблицу со всеми существующими именованными цветами. Она приведена ниже.

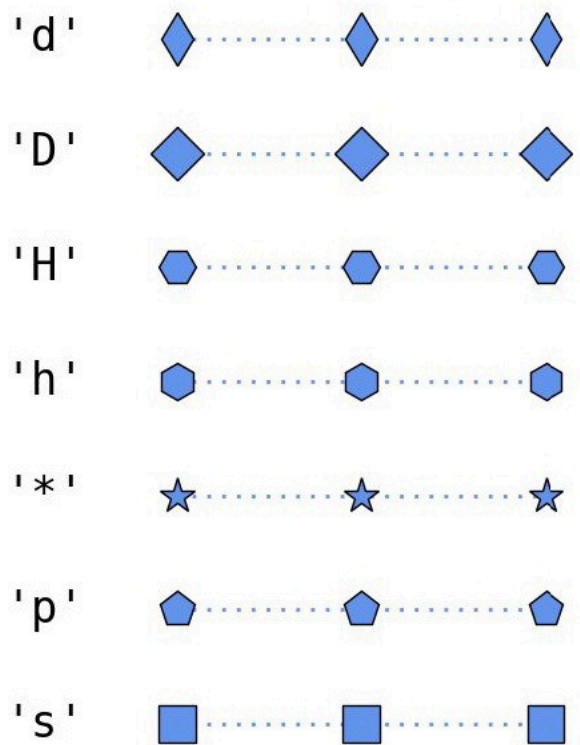
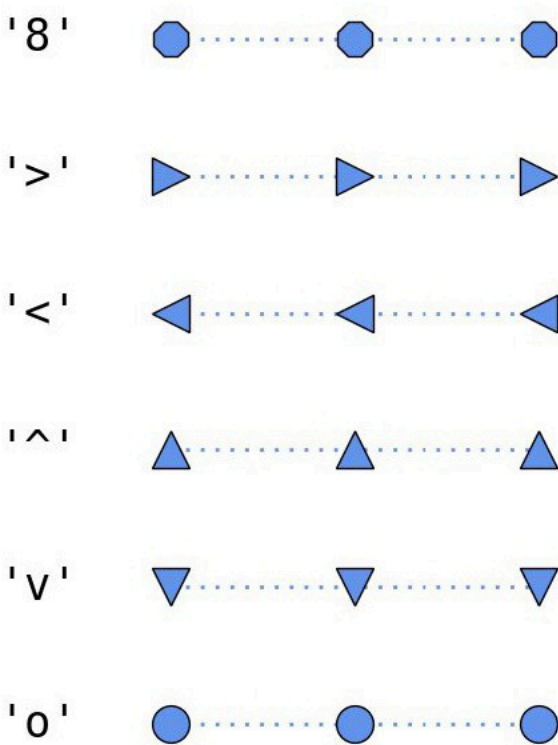
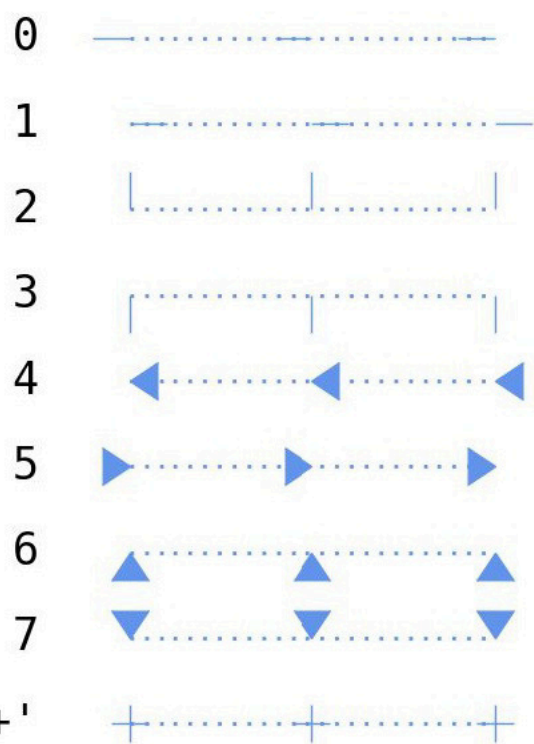
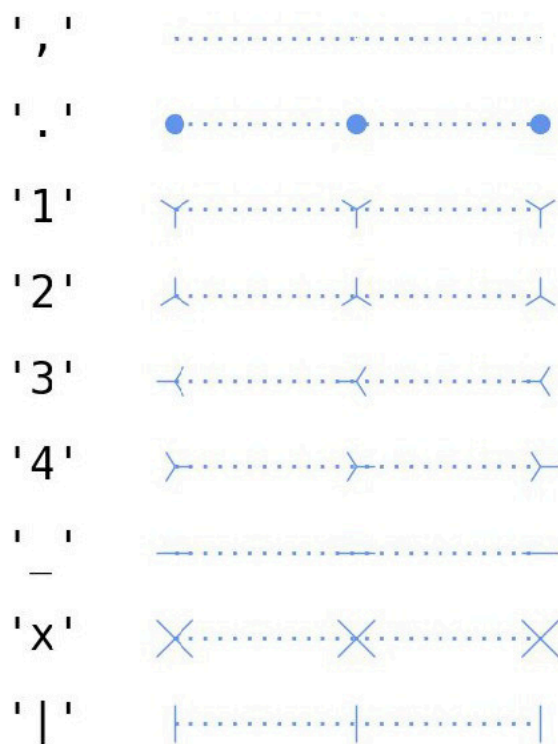


6. Определение стиля линии.

Стиль линии также можно определять одиночным символом:

Сим вол	Линия	Сим вол	Линия	Сим ол	Линия	Сим вол	Линия
-		--		:		-.	

7. Определения стиля маркера.



8. Практическая работа № 1.

8.1. Построить график одной из нижеперечисленных функций с заданными цветом, стилем линии и формой маркера; указать в легенде формулу функции:

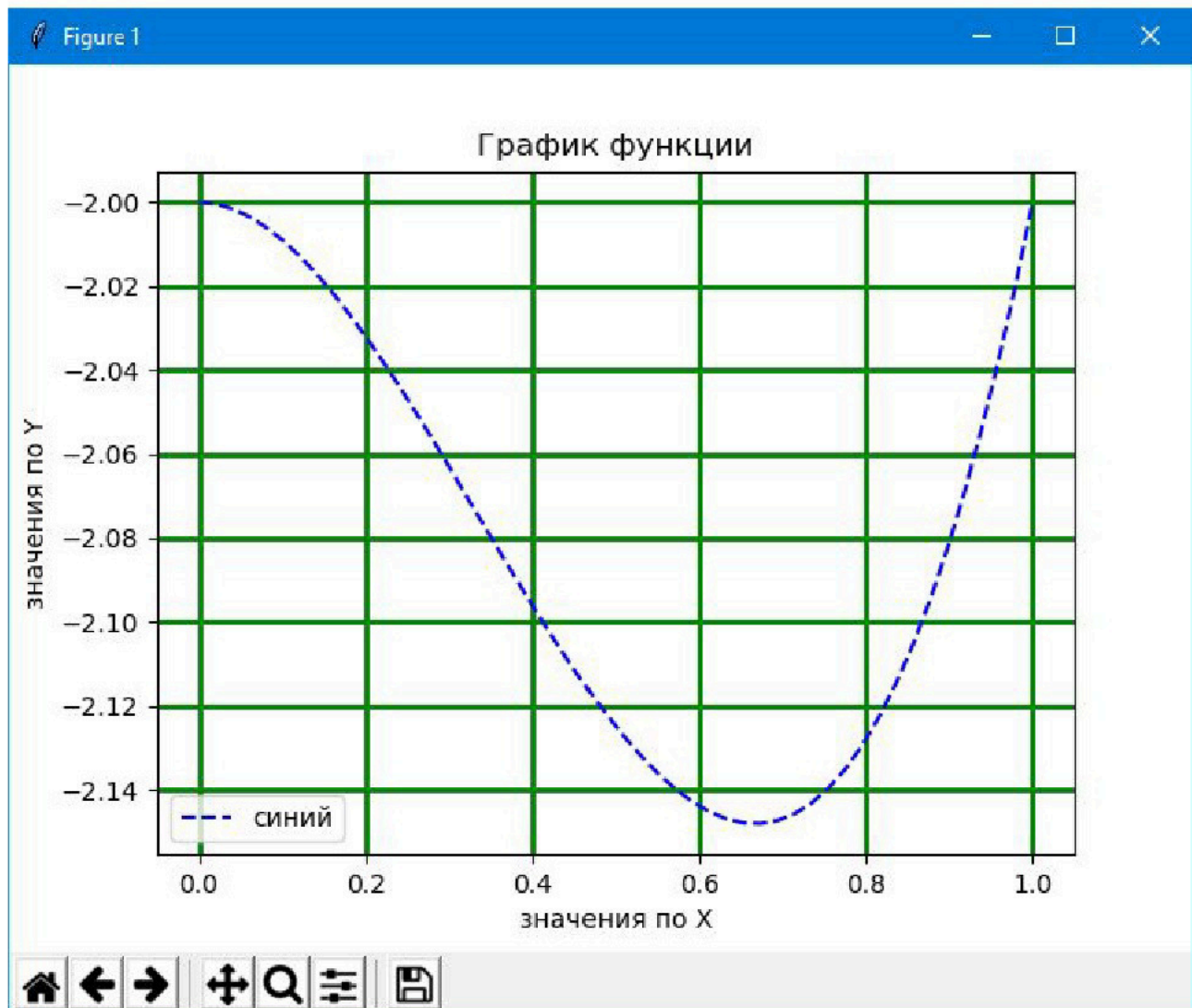
Варианты заданий:

1. $f(x) = -\sin(x) + a$ на интервале $[2; 8]$ (синий, сплошная линия)
2. $f(x) = \cos(x) + a$ на интервале $[-6; 2]$ (черный, штриховая линия)
3. $f(x) = \cos(x) - a$ на интервале $[-10; 10]$ (желтый, пунктирная линия)
4. $f(x) = (x + a)^2$ на интервале $[-15; 1]$ (зеленый, штрихпунктирная линия)
5. $f(x) = \sin(x - a)$ на интервале $[-16; -3]$ (красный, пунктирная линия)
6. $f(x) = \sin(x) - a$ на интервале $[-17; -2]$ (пурпурный, сплошная линия)
7. $f(x) = 2 \cdot (x - a)^2$ на интервале $[-8; 8]$ (голубой, сплошная линия)
8. $f(x) = -(3 \cdot x - a)^2$ на интервале $[3; 12]$ (синий, сплошная линия)
9. $f(x) = 7 \cdot x - a$ на интервале $[-5; 3]$ (черный, штриховая линия)
10. $f(x) = \cos(3 \cdot x) - a$ на интервале $[0; 25]$ (желтый, пунктирная линия)
11. $f(x) = \sin(2 \cdot x) + a$ на интервале $[-2; 6]$ (зеленый, штрихпунктирная линия)
12. $f(x) = \cos(3 \cdot x) + a$ на интервале $[-6; 2]$ (красный, пунктирная линия)
13. $f(x) = -\cos(x) + a$ на интервале $[-9; 10]$ (пурпурный, сплошная линия)
14. $f(x) = (x + a)^2$ на интервале $[-5; 5]$ (голубой, сплошная линия)
15. $f(x) = \sin(2 \cdot x - a)$ на интервале $[-16; 3]$ (синий, сплошная линия)
16. $f(x) = 2 \cdot \sin(x) - a$ на интервале $[-16; 3]$ (черный, штриховая линия)
17. $f(x) = 2 \cdot (x + a)^2$ на интервале $[-7; 6]$ (желтый, пунктирная линия)
18. $f(x) = -(2 \cdot x - a)^2$ на интервале $[-3; 112]$ (зеленый, штрихпунктирная линия)
19. $f(x) = 5 \cdot x - a$ на интервале $[-3; 4]$ (красный, пунктирная линия)
20. $f(x) = \cos(3 \cdot x) + a$ на интервале $[-12; 9]$ (пурпурный, сплошная линия)

8.2. Добавить дополнительные элементы (все возможные значения параметров смотрите на сайте: https://matplotlib.org/api/pyplot_summary.html):

- `xlabel` и `ylabel` — позволяет подписывать оси координат,
- `title` и `suptitle` — устанавливает заголовок графика,

- `legend` — устанавливает описание функции (в качестве аргумента может принимать массив),
- `grid` — включает сетку на графике. Можно просто: `plt.grid(True)`, а можно, как на рисунке:



созданным следующим скриптом:

```
import matplotlib.pyplot as plt
import numpy as np

X = np.linspace(0,1)
Y = X**3 - X**2 - 2

plt.title("График функции")
plt.xlabel("значения по X")
plt.ylabel("значения по Y")

plt.plot(X, Y, 'b--', label='синий')
legend = plt.legend()

plt.grid(color='g', linestyle='-', linewidth=2)

plt.show()
```

9. Другие виды графиков.

Примеры графиков можно посмотреть на сайте [Matplotlib](https://matplotlib.org/).

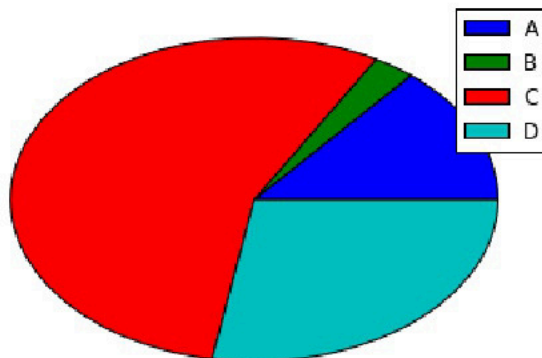
9.1. Круговая - pie

Демонстрирует соотношение между величинами (желательно не более 2-х).

```
import matplotlib.pyplot as plt
import numpy as np

X = [50, 10, 200, 100]
L = ['A', 'B', 'C', 'D']

plt.pie(X)
plt.legend(L)
plt.show()
```



9.2. Гистограмма - bar

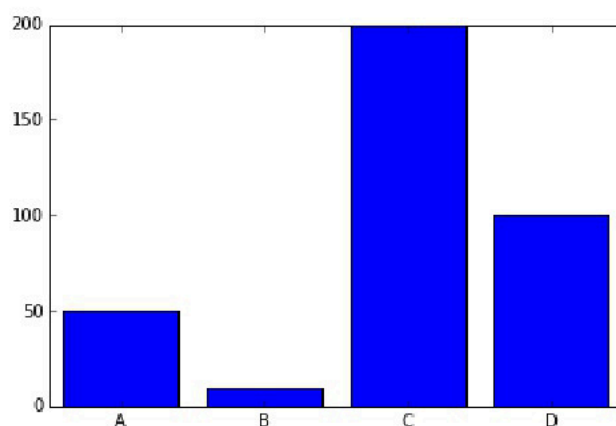
Отображает множество соизмеримых величин.

```
import matplotlib.pyplot as plt
import numpy as np

X = [50, 10, 200, 100]
Y = np.arange(len(X))

L = ('A', 'B', 'C', 'D')

plt.bar(Y, X, align='center')
plt.xticks(Y, L)
plt.show()
```

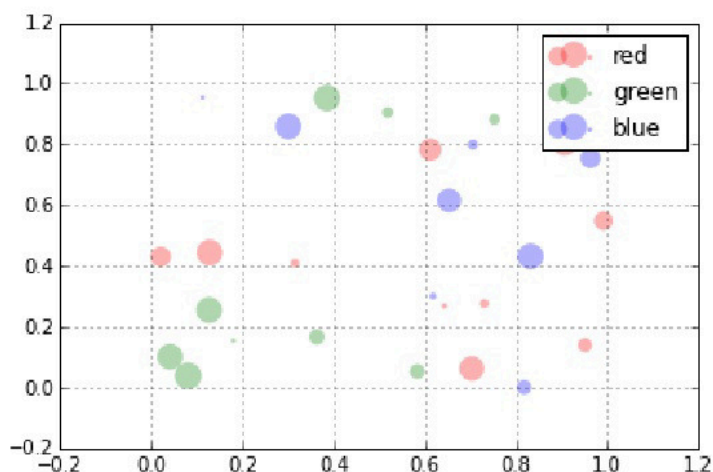


9.3. Точечная - scatter

Отображает три численных параметра (на осях + размер точки) и класс объекта.

```
import matplotlib.pyplot as plt
from numpy.random import rand

for color in ['red', 'green', 'blue']:
    n = 10
    x, y = rand(2, n)
    scale = 200.0 * rand(n)
    plt.scatter(x, y, c=color,
                s=scale, label=color,
                alpha=0.3,
                edgecolors='none')
```




```
plt.legend()
plt.grid()

plt.show()
```

9.4. Полярная диаграмма - polar

Круговой график, где для вывода точек данных используется угол и расстояние от центральной точки. Ось X располагается на границе круга, а ось Y соединяет центр круга с осью X.

```
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r

ax = plt.subplot(111,
projection='polar')
ax.plot(theta, r)
ax.set_rmax(2)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.set_rlabel_position(-22.5)
ax.grid(True)

ax.set_title("График в полярных
координатах",
va='bottom')

plt.show()
```



10. Практическая работа № 2.

10.1. Напишите скрипт, отображающий в виде круговой диаграммы зависимость между курсантами группы и количеством пропусков занятий по одной дисциплине (данные по пяти наиболее злостным прогульщикам).

10.2. Напишите скрипт, отображающий в виде гистограммы зависимость между курсантами группы и количеством пропусков занятий по разным дисциплинам (данные по пяти наиболее злостным прогульщикам).

Контрольные вопросы.

1. В чем разница между библиотеками math и Matplotlib с точки зрения программиста?
2. Для чего в теле функции func(x) примера из п.4 стоит оператор условия? Что будет, если его убрать?