

Лекция 2. Библиотека визуализации Matplotlib

1. Опишите иерархию объектов на рисунке в Matplotlib.

Все объекты в Matplotlib имеют иерархическую (древоподобную) структуру. Основными элементами являются *рисунок*, *панели* и *оси координат* (рис. 1).

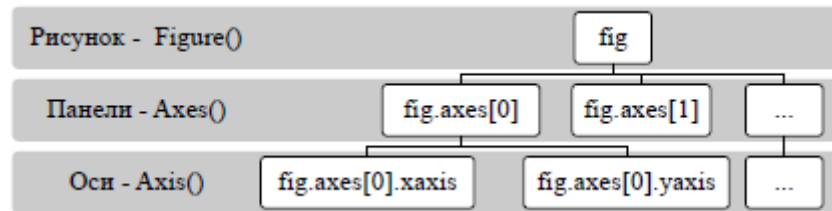


Рис. 1. Схема иерархии объектов, служащих для создания рисунка в библиотеке Matplotlib

На верхнем уровне иерархии находится окно рисунка, которое является контейнером для остальных элементов. *Рисунок* является объектом класса **Figure**.

На рисунке может располагаться несколько (одна или больше) *панелей* — независимых областей рисования. Панели являются объектами класса **Axes**.

Каждая из панелей содержит основные элементы рисунка: оси координат, метки на осях, текстовые метки, графики и т. д.

2. Пусть *fig* — объект-рисунок, созданный функцией **figure()**. Поясните результат работы следующих команд:

```
ax1 = fig.add_subplot(2, 1, 1)
```

```
ax2 = fig.add_subplot(2, 1, 2)
```

Как будут располагаться панели на рисунке **fig**?

Переменная *ax1* добавляет в окно **fig** первое положение в таблице из 2 строк и одного столбца, а переменная *ax2* займет второе положение.

3. Ниже приведены команды рисования графиков зависимостей $y1(x)$, $y2(x)$, $y3(x)$. Какими линиями будут нарисованы графики 1, 2, и 3?

```
plt.plot(x, y1, 'k-', label='1')
```

```
plt.plot(x, y2, 'g:', label='2')
```

```
plt.plot(x, y3, '--', color='orange', label='3')
```

График 1 будет нарисован черной сплошной линией, график 2 - зеленой пунктирной линией, график 3 – оранжевой штриховой линией.

4. В чём отличие функций **contour()** и **contourf()**?

contourf — то же, что **contour()**, только пространство между контурами заполняется цветовой заливкой;

5. Пусть функция f определена в некоторой области на плоскости (x, y) и имеет в этой области минимальное значение 0, максимальное значение 10. Чем будет отличаться вывод команд **contour**($x, y, f, levels=[0, 5, 10]$) и **contour**($x, y, f, 10$)?

contour($x, y, f, levels=[0, 5, 10]$) – на графике будет отображено 0, 5 и 10 изолиния.

contour($x, y, f, 10$) - на графике будет отображено 10 равномерно распределённых изолиний.

6. Воспользуйтесь функцией **help()** и выясните, для чего предназначена функция **matplotlib.pyplot.imshow()**.

matplotlib.pyplot.imshow($X, cmap=None, norm=None, *, aspect=None, interpolation=None, alpha=None, vmin=None, vmax=None, origin=None, extent=None, interpolation_stage=None, filternorm=True, filterrad=4.0, resample=None, url=None, data=None, **kwargs$)

Display data as an image, i.e., on a 2D regular raster.

The input may either be actual RGB(A) data, or 2D scalar data, which will be rendered as a pseudocolor image. For displaying a grayscale image set up the colormapping using the parameters `cmap='gray', vmin=0, vmax=255`.

The number of pixels used to render an image is set by the Axes size and the *dpi* of the figure. This can lead to aliasing artifacts when the image is resampled because the displayed image size will usually not match the size of X (see [Image antialiasing](#)). The resampling can be controlled via the *interpolation* parameter and/or `rcParams["image.interpolation"]` (default: 'antialiased').