

Задание 1

1. Создайте массив x , состоящий из 50 чисел с равным шагом от a до b .
2. Сформируйте массив y , каждый элемент которого является значением функции f от соответствующего элемента массива x .
3. Нарисуйте графики (каждый график отдельно на своём рисунке):
 - a. `plot`
 - b. `scatter`
 - c. `stem`
 - d. `fill_between`
В **`fill_between`** нужно передать один массив x и два массива y . В качестве второго массива y передайте массив нулей ($y * 0$), массив из y со сдвигом ($y + 5$), либо придумайте что-то своё, если хотите.
4. Выберите один из построенных графиков (`plot`, `scatter`, `stem`) и попробуйте максимально его задекорировать. Но как минимум, должны быть следующие пункты:
 - a. Цвет графика
 - b. Цвет фона
 - c. Заголовок графика
 - d. Подпись оси x
 - e. Подпись оси y
 - f. Маркер или стиль линии
5. Сохраните каждый рисунок в картинку в формате `png`
 - a. `'1a.png'`
 - b. `'1b.png'`
 - c. `'1c.png'`
 - d. `'1d.png'`

Подсказка: сохранить рисунок можно кнопкой , либо с помощью `fig.savefig('lol.png')`

Пример

$$f(x) = x ** 2$$

$$a, b = -5, 5$$

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

x = np.linspace(-5, 5, 50)
y = x ** 2

fig, ax = plt.subplots()
ax.plot(x, y, color='red', linestyle='--')
ax.set_facecolor("mistyrose")
ax.set_title('Парабола')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_xlim(-25, 25)
ax.set_ylim(0, 50)
ax.grid()

plt.show()
```

Индивидуальные задания

1. $f(x) = x ** 3$
 $a = -5$
 $b = 5$
2. $f(x) = |x|$
 $a = -5$
 $b = 5$

3. $f(x) = |x^{**2} - 3|$
 $a = -5$
 $b = 5$
4. $f(x) = x^{**4} - 10 * x^{**2}$
 $a = -5$
 $b = 5$
5. $f(x) = \sin(x)$
 $a = -2 * \pi$
 $b = 2 * \pi$
6. $f(x) = |\sin(x)|$
 $a = -2 * \pi$
 $b = 2 * \pi$

Задание 2

1. Создайте массив **t**, состоящий из 1000 чисел с равным шагом от **a** до **b**.
2. Сформируйте массив **x**, каждый элемент которого является значением функции **f** от соответствующего элемента массива **t**.
3. Сформируйте массив **y**, каждый элемент которого является значением функции **g** от соответствующего элемента массива **t**.
4. Нарисуйте график **plot**
5. Попробуйте максимально его задезайнить.
Но как минимум, должны быть следующие пункты:
 - a. Цвет графика
 - b. Цвет фона
 - c. Заголовок графика
 - d. Подпись оси x
 - e. Подпись оси y
 - f. Стилль линии или толщина линии.
6. Сохраните рисунок в картинку '2.png'

Пример

$f(t) = \sin(t)$
 $g(t) = \sin(2 * t)$
 $a = 0$
 $b = 2 * \pi$

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

t = np.linspace(0, 2 * np.pi, 1000)
x = np.sin(t)
y = np.sin(2 * t)

fig, ax = plt.subplots()
ax.plot(x, y, color='red', linestyle='-', linewidth=20, alpha=0.3)
ax.set_facecolor("mistyrose")
ax.set_title('Ничего себе.')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.grid()

plt.show()
```

Индивидуальные задания

1. $f(t) = \cos(t)$
 $g(t) = \sin(t)$
 $a = 0$
 $b = 2 * \pi$

2. $f(t) = \cos(3 * t)$
 $g(t) = \sin(5 * t)$
 $a = 0$
 $b = 2 * \pi$
3. $f(t) = \cos(t) ** 3$
 $g(t) = \sin(t) ** 3$
 $a = 0$
 $b = 2 * \pi$
4. $f(t) = \cos(t) * \sin(5 * t)$
 $g(t) = \sin(t) * \sin(5 * t)$
 $a = 0$
 $b = \pi$
5. $f(t) = \cos(t) * \sin(3/4 * t)$
 $g(t) = \sin(t) * \sin(3/4 * t)$
 $a = 0$
 $b = 8 * \pi$
6. $f(t) = \cos(t) * \sin(9/4 * t)$
 $g(t) = \sin(t) * \sin(9/4 * t)$
 $a = 0$
 $b = 8 * \pi$

Задание 3

У вас есть 4 графика из первого задания и 1 график из второго. Выберите из них любые четыре графика и нарисуйте их на одном рисунке на четырёх разных осях (можно в виде таблицы 2x2, можно четыре в ряд). Сохраните результат в картинку '3.png'

В примере показан способ нарисовать четыре графика на разных осях в виде таблицы 2x2.

Пример

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

nrows, ncols = 2, 2
a0, a1, a2, a3 = 1, 3, 5, 8

x = np.linspace(-np.sqrt(3), np.sqrt(3), 100)
y0 = np.cbrt(x ** 2) + np.sqrt(3.3 - x ** 2) * np.sin(a0 * np.pi * x)
y1 = np.cbrt(x ** 2) + np.sqrt(3.3 - x ** 2) * np.sin(a1 * np.pi * x)
y2 = np.cbrt(x ** 2) + np.sqrt(3.3 - x ** 2) * np.sin(a2 * np.pi * x)
y3 = np.cbrt(x ** 2) + np.sqrt(3.3 - x ** 2) * np.sin(a3 * np.pi * x)

fig, ax = plt.subplots(nrows, ncols)
fig.set_size_inches(8, 6)
ax[0, 0].plot(x, y0)
ax[0, 1].plot(x, y1)
ax[1, 0].plot(x, y2)
ax[1, 1].plot(x, y3)

plt.show()
```

Задание 4*

Найдите в интернете или попробуйте изобрести сами уравнение какой-нибудь интересной кривой линии в 3D или поверхности в 3D (на ваш выбор).

Нарисуйте её, задекорируйте по вкусу и сохраните результат в картинку '4.png'.

Подсказка. Помните, что кривулька задаётся одним линспейсом, а поверхность — двумя.

У меня получился вот такой шарик.

Пример

```
import matplotlib.pyplot as plt
```

```
import numpy as np

t = np.linspace(0, 30 * np.pi, 1000)
r = np.sqrt(np.pi - (t / 30 - np.pi / 2) ** 2)
x = r * np.cos(t)
y = r * np.sin(t)
z = t / (30 * np.pi)

fig, ax = plt.subplots(subplot_kw=dict(projection="3d"))
ax.plot(x, y, z)
plt.show()
```