https://dfedorov.spb.ru

Пакет matplotlib - это библиотека, предназначенная для построения двумерных диаграмм и графиков.

В интерактивном режиме для проверки введите команду:

```
import matplotlib
```

Пакет matplotlib содержит модуль pyplot, который необходимо импортировать для создания графиков (используем псевдоним plt):

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

Функция **plot** используется для создания линейного графика, который соединяет серию точек данных отрезками прямой. Линейный график имеет горизонтальную ось x и вертикальную ось y. Каждая точка данных на графике имеет координаты (X, Y). Для того чтобы создать линейный график, сначала необходимо создать два списка: один с координатами X каждой точки данных и другой с координатами Y каждой точки данных.

```
x_coords = [0, 2, 3, 4, 6]

y_coords = [0, 3, 1, 5, 1]
```

```
# Эта программа выводит простой линейный график.
import matplotlib.pyplot as plt
#%matplotlib inline

# Создать списки для координат X и Y каждой точки данных.
x_coords = [0, 2, 3, 4, 6]
y_coords = [0, 3, 1, 5, 1]

# Построить линейный график.
plt.plot(x_coords, y_coords)

# Показать линейный график.
plt.show()
```

```
import matplotlib.pyplot as plt
#%matplotlib inline

# Создать списки для координат X и Y каждой точки данных.
x_coords = [0, 2, 3, 4, 6]
y_coords = [0, 3, 1, 5, 1]

# Построить линейный график.
plt.plot(x_coords, y_coords)

# Добавить заголовок.
plt.title('Образец данных')

# Добавить на оси описательные метки.
plt.xlabel('Это ось X')
plt.ylabel('Это ось Y')

# Добавить сетку.
plt.grid(True)

# Показать линейный график.
plt.show()
```

При помощи функции **title** можно добавлять в график заголовок. Заголовок будет выведен чуть выше графика. Кроме того, при помощи функций **ylabel** и **xlabel** можно добавить описательные метки на оси **x** и **y**. В график можно добавить сетку, вызвав для этого функцию **grid(True)**.

```
# Эта программа выводит простой линейный график.
import matplotlib.pyplot as plt
#%matplotlib inline

# Создать списки для координат X и Y каждой точки данных.
x1 = [0, 2, 3, 4, 12]
y1 = [0, 3, 1, 5, 3]
x2 = [0, 5, 7, 10, 12]
y2 = [0, 2, 3, 4, 6]

# Построить линейные графики.
plt.plot(x1, y1, x2, y2)

# Показать линейный график.
plt.show()
```

```
## Dependencies
import matplotlib.pyplot as plt
## Data
cardiac_cycle = [62, 60, 62, 64, 68, 77, 80, 76, 71, 66, 61, 60, 62]

# expected_cycles = cardiac_cycle * 10
expected_cycles = cardiac_cycle

## Result
plt.plot(expected_cycles)
plt.show()
```

Круговая диаграмма - это график, который показывает круг, поделенный на доли. Круг представляет целое, а секторы - процентное содержание целого.

Для создания круговой диаграммы используется функция **pie** из модуля **matplotlib.pyplot**.

Когда вызывается функция **pie**, ей в качестве аргумента передается список значений. Функция **pie** вычисляет сумму значений в списке и затем использует эту сумму в качестве значения целого. Затем каждый элемент в списке станет сектором (долей) в круговой диаграмме.

Размер сектора представляет значение этого элемента как процентное содержание целого.

```
# Эта программа выводит простую круговую диаграмму.
import matplotlib.pyplot as plt
#%matplotlib inline

# Создать список значений
values = [20, 60, 80, 40]

# Создать из этих значений круговую диаграмму.
plt.pie(values)

# Показать круговую диаграмму.
plt.show()
```

https://dfedorov.spb.ru