



1.1. Теоретический материал – Типы данных

Все типы данных в Python делятся на две основные категории:
изменяемые (mutable) и неизменяемые (immutable).

Неизменяемые объекты (immutable)

Объекты, содержимое которых нельзя изменить после создания:

- **Числовые данные:** `int`, `float`
- **Логический тип:** `bool`
- **Отсутствие значения:** `None`
- **Символьные строки:** `str`
- **Кортежи:** `tuple`

Изменяемые объекты (mutable)

Объекты, которые можно изменять после создания:

- **Списки:** `list`
- **Множества:** `set`
- **Словари:** `dict`

1.2. Пример

Задача:

Выведите на печать и определите тип переменной.

Решение:

```
x = 3+5.2*7
y = None
z = 'a', 5, 12.345, (2, 'b')
df = ['Антонова Антонина', 34, 'ж'], ['Борисов Борис', 26, 'м']
A = {1, 'title', 2, 'content'}
print(x, '|', type(x), '\n',
      y, '|', type(y), '\n',
      z, '|', type(z), '\n',
      df, '|', type(df), '\n',
      A, '|', type(A))
```

Ответ:

```
39.4 | <class 'float'>
None | <class 'NoneType'>
```

```
('a', 5, 12.345, (2, 'b')) | <class 'tuple'>
(['Антонова Антонина', 34, 'ж'], ['Борисов Борис', 26, 'м']) |
<class 'tuple'>
{1, 2, 'content', 'title'} | <class 'set'>
```

1.3. Задание

Задача:

Выведите на печать и определите тип переменной.

Решение:

```
In [6]: x = 5 >= 2
A = {1, 3, 7, 8}
B = {2, 4, 5, 10, 'apple'}
C = A & B
df = ('Антонова Антонина', 34, 'ж')
z = 'type'
D = [1, 'title', 2, 'content']
print(x, '|', type(x), '\n',
      A, '|', type(A), '\n',
      B, '|', type(B), '\n',
      C, '|', type(C), '\n',
      df, '|', type(df), '\n',
      z, '|', type(z), '\n',
      D, '|', type(D))
```

```
True | <class 'bool'>
{8, 1, 3, 7} | <class 'set'>
{2, 4, 5, 10, 'apple'} | <class 'set'>
set() | <class 'set'>
('Антонова Антонина', 34, 'ж') | <class 'tuple'>
type | <class 'str'>
[1, 'title', 2, 'content'] | <class 'list'>
```

2.1. Теоретический материал – Условный оператор

if — Условный оператор

В коде часто приходится проверять выполнимость или невыполнимость каких-то условий.

Синтаксис:

```
if <условие1 (булевоe выражение)>:
    <код, который выполнится, если условие верно>
elif <условие2 (булевоe выражение)>:
    <код, который выполнится, если условие1 было неверно, а условие2
    верно>
else:
```

<код, который выполнится, если условие1 и условие2 были неверны>

- ♦ Обратите внимание:

Код, который должен выполняться внутри каждого условия, записывается с **отступом в 4 пробела** от уровня `if`, `elif` и `else`.

В Python области видимости переменных обозначаются **отступами**.

То есть, **отступы позволяют понять**, где начинается код, который должен выполняться при выполнении условия в `if`, и где он заканчивается.

2.2. Пример

Задача:

Вывести на экран, является ли переменная `x` положительной, отрицательной или равна нулю.

Решение:

```
x = 125
if x < 0:
    print("x отрицательный")
elif x == 0:
    print("x равен нулю")
else:
    print("x положительный")
```

Ответ:

x положительный

2.3. Задание

Задача:

Напишите код. Задаётся переменная `x`. Напечатайте, к какому из интервалов принадлежит значение `x`:

- $(-\infty, -5)$
- $[-5, 5]$
- $(5, +\infty)$

Решение:

```
In [7]: x = float(input())
if x < - 5:
    print ("x принадлежит интервалу: (-infinity, -5)")
elif x > 5:
    print ("x принадлежит интервалу: (5, +infinity)")
else:
    print("x принадлежит интервалу: [-5, 5]")
```

x принадлежит интервалу: (-infinity, -5)

3.1. Теоретический материал – Циклы, массивы, срезы

Циклы `for` и `while`

Цикл `while` используется, когда необходимо выполнять блок кода до тех пор, пока условие истинно.

Синтаксис:

```
while условие:
    инструкции
```

Инструкции, которые относятся к циклу `while`, располагаются на последующих строках и должны иметь отступ от начала ключевого слова `while`. **Цикл `for`** проходит по набору значений, помещает каждое значение в переменную с которой в цикле можно производить различные действия.

Синтаксис:

```
for переменная in набор_значений:
    инструкции
```

После ключевого слова `for` идет название переменной, в которую будут помещаться значения. После оператора `in` указывается набор значений и двоеточие. Инструкции, которые относятся к циклу, располагаются на последующих строках и должны иметь отступ. При создании цикла удобно пользоваться функцией **`range(a, b)`**, которая создает последовательность чисел от **`a`** до **`b-1`**.

3.2.1. Пример

Задача:

Вывести все числа от 1 до 10 с шагом 3.

Решение:

```
x = 1
while x <= 10:
    print(x)
    x += 3
```

Ответ:

```
1
4
7
10
```

3.2.2. Пример

Задача:

Задан список. Напечатать все элементы списка.

Решение:

```
models = ['KNN', 'decision tree', 'linear model']
for model in models:
    print(model)
```

Ответ:

```
KNN
decision tree
linear model
```

3.2.3. Пример

Задача:

Создать список от 1 до 99 с шагом 7.

Решение:

```
list_int = range(1, 100, 7)
print(list(list_int))
```

Ответ:

```
[1, 8, 15, 22, 29, 36, 43, 50, 57, 64, 71, 78, 85, 92, 99]
```

3.2.4. Пример

Задача:

Вывести на экран числа от 1 до 105 с шагом 25, используя функцию `range()`.

Решение:

```
for i in range(5, 106, 25):  
    print(i)
```

Ответ:

```
5  
30  
55  
80  
105
```

3.2.5. Пример

Задача:

Срез. Необходимо получить каждый второй элемент массива `a`, начиная с индекса 1 (включительно) и заканчивая индексом 7 (не включительно), а также переставить все элементы в обратном порядке.

Решение:

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
b = a[1:7:2]  
c = a[::-1]  
print(b)  
print(c)
```

Ответ:

```
[2, 4, 6]  
[9, 8, 7, 6, 5, 4, 3, 2, 1]
```

3.3.1. Задание

Задача:

Выведите числа из примера на `while` (3.2.1) в обратном порядке.

Решение:

```
In [8]: x = 10
while x > 0:
    print(x)
    x -= 3
```

```
10
7
4
1
```

3.3.2. Задание

Задача:

При решении задач машинного обучения часто в качестве объектов исследования выступает человек.

Создайте список значимых характеристик (признаков), идентифицирующих человека.

Выведите список на экран.

Решение:

```
In [9]: human_characteristics = ['возраст', 'пол', 'рост', 'вес', 'образование', 'проф']
for human_characteristic in human_characteristics:
    print(human_characteristic)
```

```
возраст
пол
рост
вес
образование
профессия
уровень дохода
```

3.3.3. Задание

Задача:

Создать список чисел от 2 до 15 с шагом 1.

Решение:

```
In [10]: numbers = list(range(2, 16))
print(numbers)
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

3.3.4. Задание

Задача:

Выведите числа из примера на `for` с функцией `range()` (3.2.4) в обратном порядке.

Решение:

```
In [13]: for i in range(105, 4, -25):  
         print(i)
```

```
105  
80  
55  
30  
5
```

3.3.5. Задание

Задача:

Срез. Напишите код, который все элементы массива `x` с чётными индексами переставит в обратном порядке.

То есть, если `x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`, то код должен сформировать:

```
[8, 1, 6, 3, 4, 5, 2, 7, 0, 9]
```

Решение:

```
In [14]: x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
         i = x[::2][::-1]  
         x[::2] = i  
         print(x)
```

```
[8, 1, 6, 3, 4, 5, 2, 7, 0, 9]
```

4.1. Теоретический материал – Библиотеки

Машинное обучение опирается на фундаментальные понятия и законы математики и информатики. Библиотека математических функций (`math`) применяется при решении большинства прикладных задач анализа данных. Также, и некоторые другие библиотеки Python содержат математические функции (например, `NumPy`). Важную роль играет библиотека `matplotlib`, которая позволяет производить визуализацию расчетов в удобной для восприятия человеком форме: графики, диаграммы, гистограммы.

Подключение библиотек — синтаксис:

```
import math as m
import numpy as np
import matplotlib.pyplot as plt
etc...
```

Библиотеки в python содержат не только функции, позволяющие решать различные задачи, но и некоторые популярные константы, например число Эйлера.

4.2.1. Пример

Задача:

Рассчитать синус от числа Эйлера (e) в радианах и вывести результат на экран.

Решение:

```
import math as m
print(m.sin(m.e))
```

Ответ:

0.41078129050290885

4.2.2. Пример

Задача:

Для функции $f(x) = \sin(x e^{\cos(x)})$ на интервале $[0; 10]$ с шагом 0.05 построить график функции, закрасить площадь под ним и вычислить значение этой площади. Для вычисления площади воспользуйтесь функцией `trapz()`, вычисляющей интеграл по правилу трапеции. Для ее корректной работы необходимо подключить следующие библиотеки:

```
from scipy.integrate import.simps
from numpy import trapz
```

Решение:

```
In [29]: pip install matplotlib
```

```
Cell In[29], line 1
      pip install matplotlib
      ^
```

SyntaxError: invalid syntax

```
In [22]: pip install --upgrade scipy
```

Requirement already satisfied: scipy in c:\users\user\anaconda3\lib\site-packages (1.15.3)

Collecting scipy

Downloading scipy-1.16.1-cp313-cp313-win_amd64.whl.metadata (60 kB)

Requirement already satisfied: numpy<2.6,>=1.25.2 in c:\users\user\anaconda3\lib\site-packages (from scipy) (2.1.3)

Downloading scipy-1.16.1-cp313-cp313-win_amd64.whl (38.5 MB)

```
----- 0.0/38.5 MB ? eta -:-:-
-- ----- 2.1/38.5 MB 10.8 MB/s eta 0:00:04
---- ----- 4.5/38.5 MB 11.3 MB/s eta 0:00:04
----- 6.8/38.5 MB 11.2 MB/s eta 0:00:03
----- 8.7/38.5 MB 10.5 MB/s eta 0:00:03
----- 10.2/38.5 MB 10.1 MB/s eta 0:00:03
----- 12.1/38.5 MB 9.8 MB/s eta 0:00:03
----- 13.9/38.5 MB 9.6 MB/s eta 0:00:03
----- 15.2/38.5 MB 9.2 MB/s eta 0:00:03
----- 16.5/38.5 MB 9.0 MB/s eta 0:00:03
----- 17.6/38.5 MB 8.6 MB/s eta 0:00:03
----- 18.6/38.5 MB 8.3 MB/s eta 0:00:03
----- 19.7/38.5 MB 7.9 MB/s eta 0:00:03
----- 20.4/38.5 MB 7.6 MB/s eta 0:00:03
----- 21.5/38.5 MB 7.4 MB/s eta 0:00:03
----- 22.5/38.5 MB 7.3 MB/s eta 0:00:03
----- 23.6/38.5 MB 7.1 MB/s eta 0:00:03
----- 24.4/38.5 MB 6.9 MB/s eta 0:00:03
----- 25.4/38.5 MB 6.8 MB/s eta 0:00:02
----- 26.0/38.5 MB 6.6 MB/s eta 0:00:02
----- 26.5/38.5 MB 6.4 MB/s eta 0:00:02
----- 27.3/38.5 MB 6.2 MB/s eta 0:00:02
----- 27.8/38.5 MB 6.1 MB/s eta 0:00:02
----- 28.6/38.5 MB 5.9 MB/s eta 0:00:02
----- 29.1/38.5 MB 5.8 MB/s eta 0:00:02
----- 29.9/38.5 MB 5.7 MB/s eta 0:00:02
----- 30.7/38.5 MB 5.6 MB/s eta 0:00:02
----- 31.5/38.5 MB 5.5 MB/s eta 0:00:02
----- 32.0/38.5 MB 5.5 MB/s eta 0:00:02
----- 32.5/38.5 MB 5.4 MB/s eta 0:00:02
----- 33.3/38.5 MB 5.3 MB/s eta 0:00:01
----- 33.8/38.5 MB 5.2 MB/s eta 0:00:01
----- 34.9/38.5 MB 5.2 MB/s eta 0:00:01
----- 35.4/38.5 MB 5.1 MB/s eta 0:00:01
----- 35.9/38.5 MB 5.0 MB/s eta 0:00:01
----- 36.4/38.5 MB 5.0 MB/s eta 0:00:01
----- 37.2/38.5 MB 4.9 MB/s eta 0:00:01
----- 37.7/38.5 MB 4.9 MB/s eta 0:00:01
----- 38.5/38.5 MB 4.8 MB/s eta 0:00:00
```

Installing collected packages: scipy

Attempting uninstall: scipy

Found existing installation: scipy 1.15.3

Uninstalling scipy-1.15.3:

Successfully uninstalled scipy-1.15.3

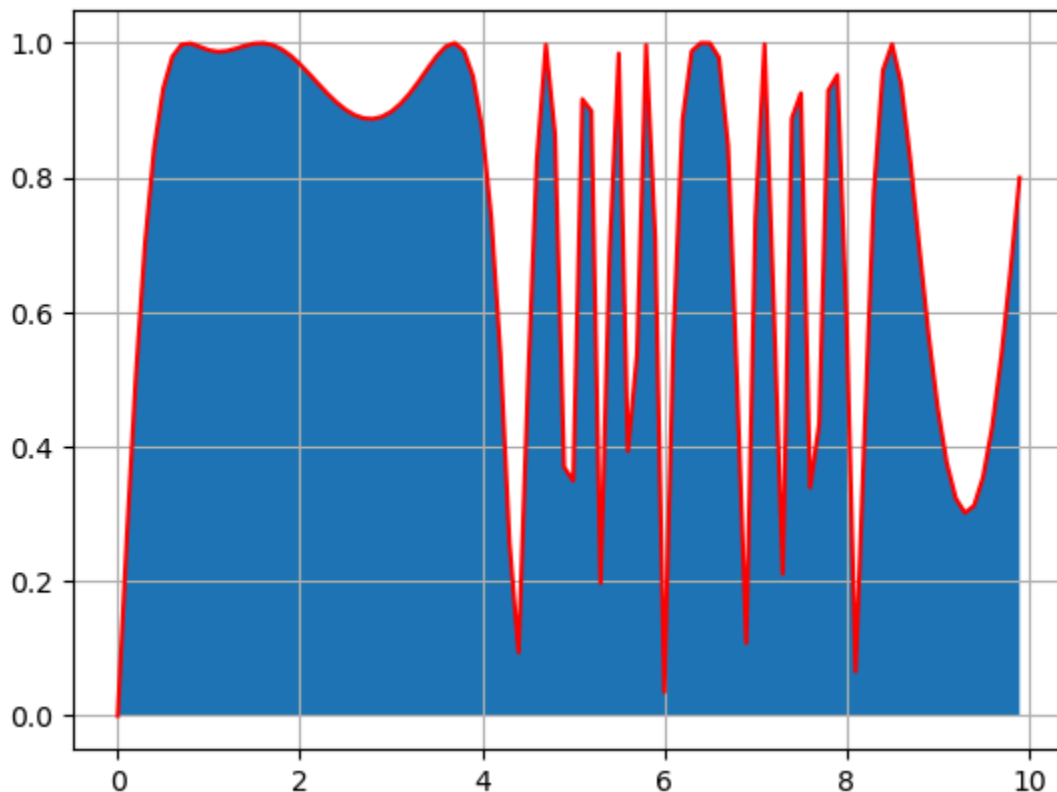
Successfully installed scipy-1.16.1

Note: you may need to restart the kernel to use updated packages.

```
In [43]: import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simpson
from numpy import trapezoid
x = np.arange(0.0, 10, 0.1)
y = np.abs(np.sin(x * np.exp(np.cos(x))))
plt.grid()
plt.plot(x, y, c="r")
plt.fill_between(x, y)

area = trapezoid(y, x)
print(area)
```

7.350792907748665



4.2.3. Пример

Задача:

Дано некоторое распределение оценок в группе за экзамен.

Нарисовать круговую (pie) и точечную (scatter) диаграммы распределения.

Решение:

```
In [45]: from matplotlib import pyplot as plt
import numpy as np

marks = ['Неуд', 'Удовл', 'Хор', 'Отл']
```

```

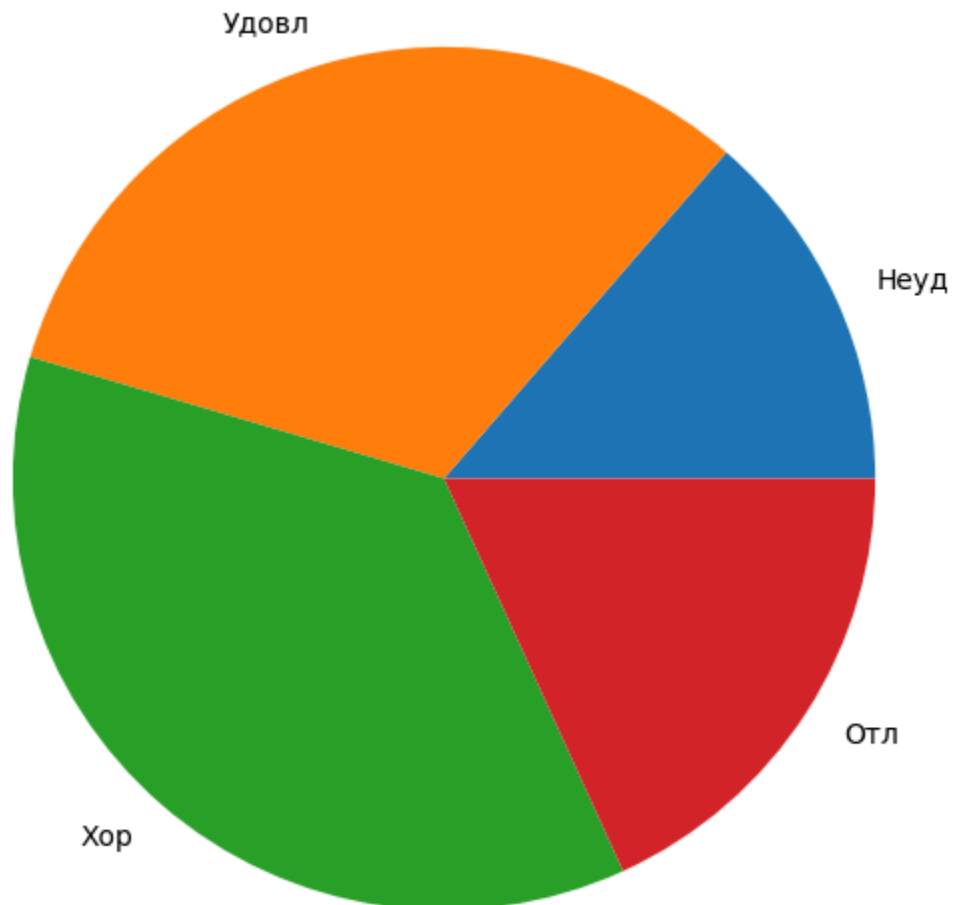
data = [3, 7, 8, 4]

fig = plt.figure(figsize=(10, 7))

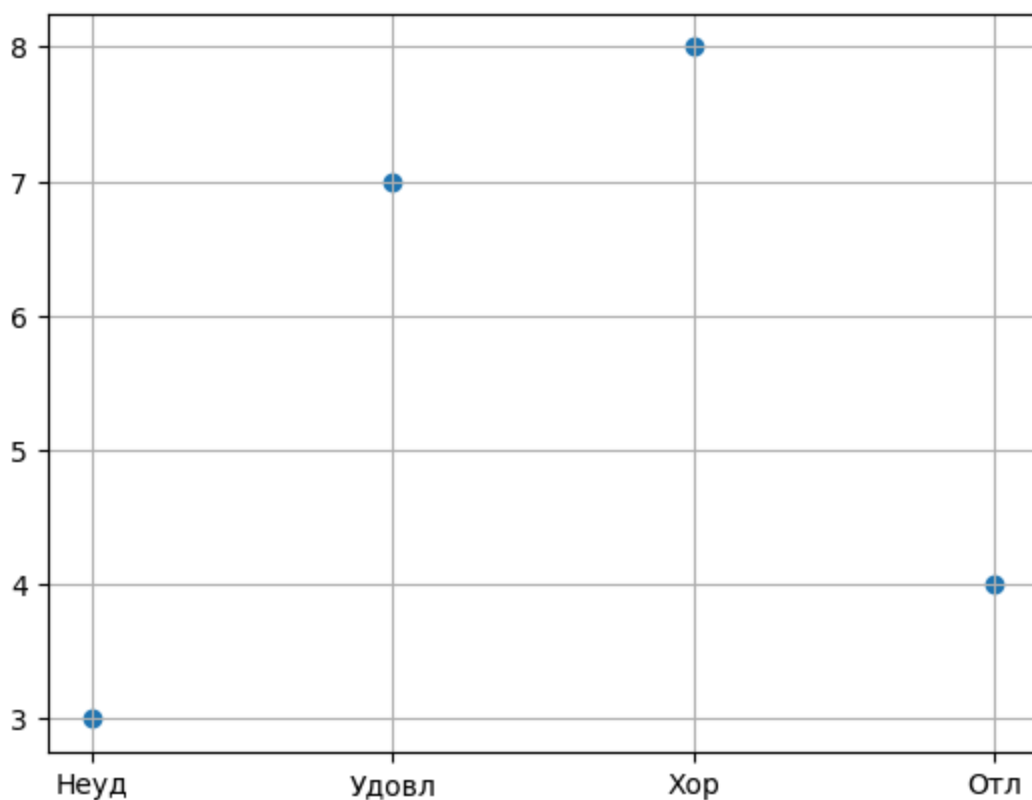
data = [3, 7, 8, 4]
fig = plt.figure(figsize=(10, 7))
plt.pie(data, labels = marks)
plt.show()
plt.grid()
plt.scatter(marks, data)

```

<Figure size 1000x700 with 0 Axes>



Out[45]: <matplotlib.collections.PathCollection at 0x1f754e6fc50>



4.3.1. Задание

Задача:

Задайте массив случайных значений из интервала $(0; 1)$. Рассчитайте:

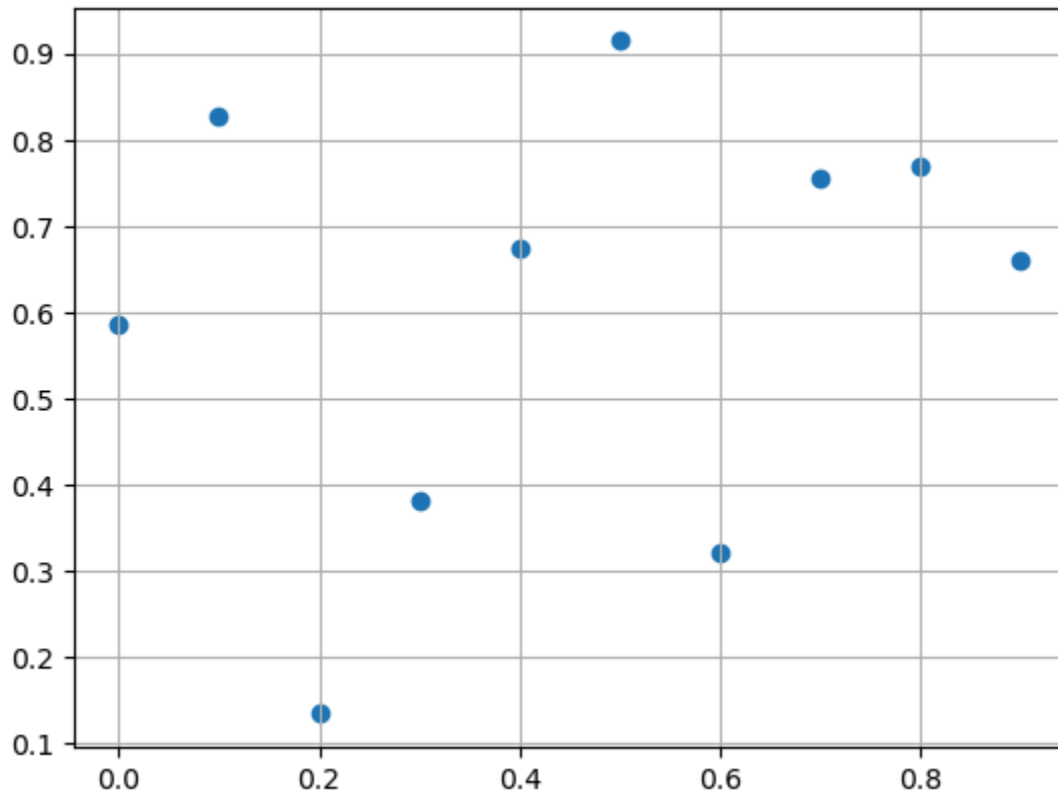
- среднее значение (среднее арифметическое)
- медиану Сравните полученные результаты и сделайте выводы о распределении значений. Постройте точечную диаграмму рассеяния (scatter plot) для полученного ряда.

Решение:

```
In [48]: import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simpson
x = np.arange(0.0, 1, 0.1)
y = np.random.rand(10)
mean_y = np.mean(y)
median_y = np.median(y)
print(mean_y)
print(median_y)
plt.grid()
plt.scatter(x, y)
```

0.6018031261740249
0.6656631592278098

Out[48]: <matplotlib.collections.PathCollection at 0x1f754f94050>



4.3.2. Задание

Задача:

Дана функция:

$$f(x) = \frac{\sqrt{1 + e^{\sqrt{x}}} + \cos^2 x}{|1 - \sin^3 x| + \ln|2x|}$$

Создать массив из 10 значений функции (\$x\$ например, при \$x\$ от 1 до 10).

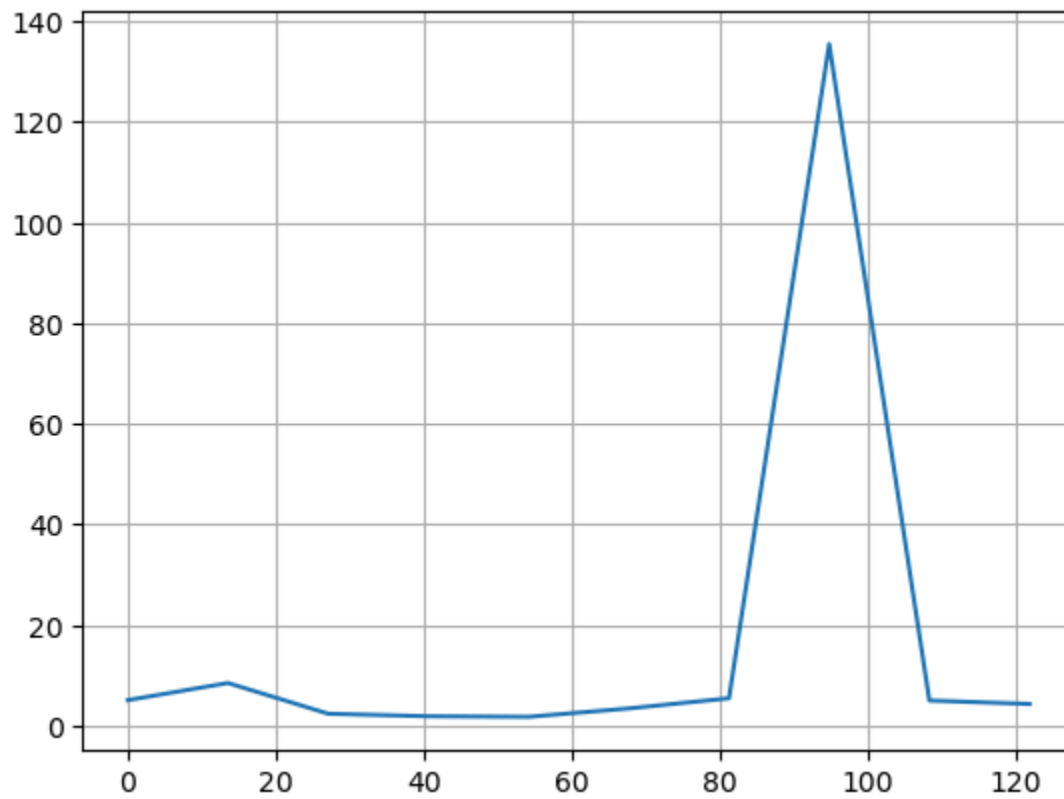
Выделить срез первой половины массива и построить:

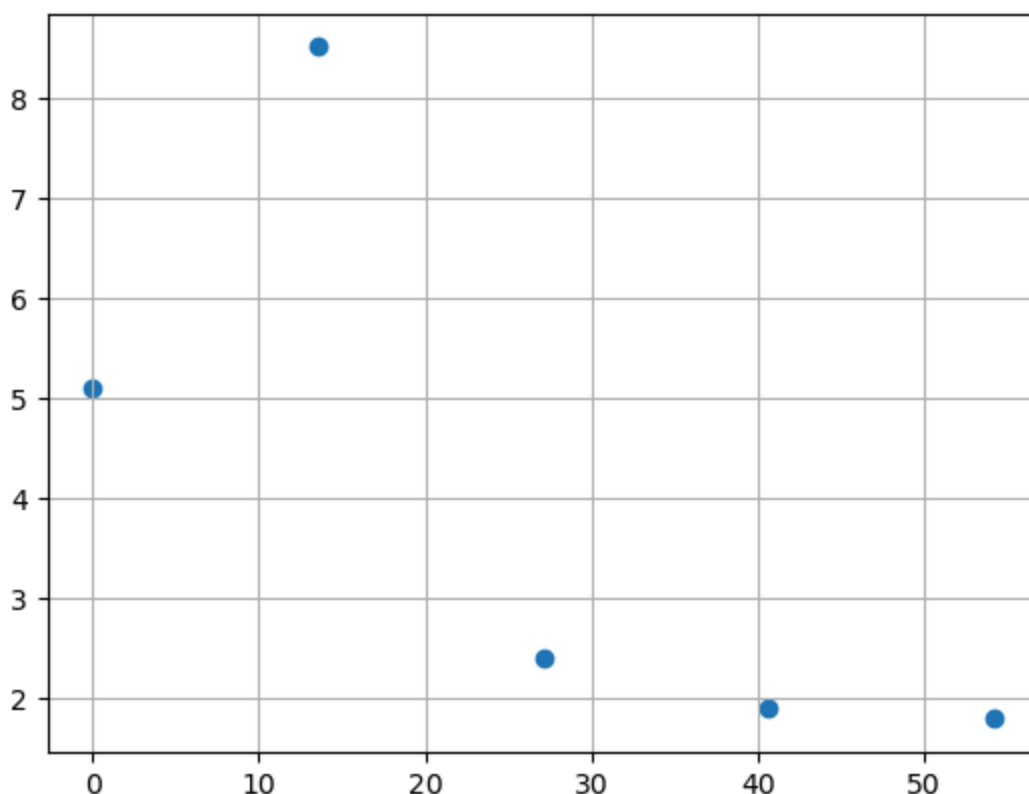
- линейный график для всего массива
- точечный график для первой половины (среза)

Решение:

```
In [51]: import math
import numpy as np
import matplotlib.pyplot as plt
f = [(math.sqrt(1 + math.e ** math.sqrt(x)) + math.cos(x ** 2)) / abs(1 - (mat
x = np.arange(0.0, max(f), max(f)/10)
newf = f[:5]
```

```
plt.grid()
plt.plot(x, f)
plt.show()
plt.grid()
plt.scatter(x[:5], newf)
plt.show()
```





4.3.3. Задание

Задача:

Дана функция:

$$f(x) = \left| \cos \left(x e^{\{\cos(x) + \ln(x+1)\}} \right) \right|$$

Построить график функции на интервале $(0; 10)$ с шагом 1 , закрасить площадь под кривой и вычислить значение этой площади. Для вычисления площади воспользуйтесь функцией `trapez()`, вычисляющей интеграл по правилу трапеции.

Для ее корректной работы необходимо подключить следующие библиотеки:

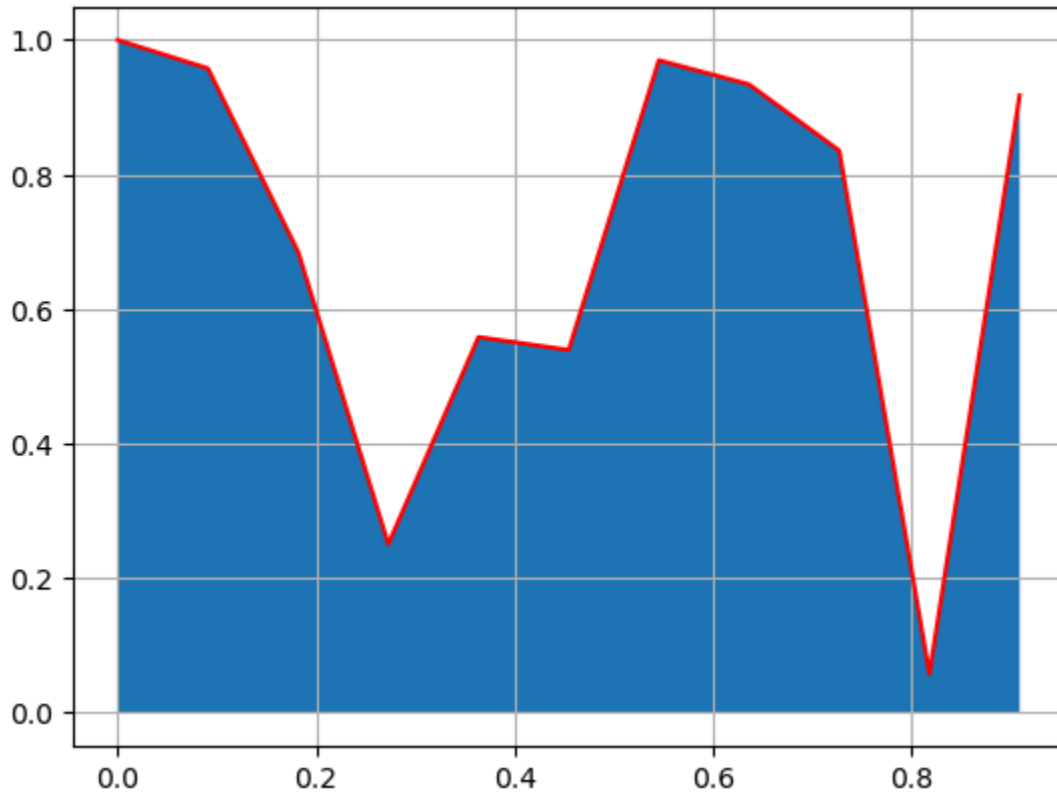
```
from scipy.integrate import simps
from numpy import trapez
```

Решение:

```
In [58]: import math
import numpy as np
import matplotlib.pyplot as plt
from numpy import trapezoid
f = [(abs(math.cos(x * (math.e ** (math.cos(x) + math.log(x + 1)))))) for x in
```

```
x = np.arange(0.0, max(f), max(f)/11)
plt.grid()
plt.plot(x, f, c = "r")
plt.fill_between(x, f)
area = trapezoid(f, x)
print(area)
```

0.6134712013325203



4.3.4. Задание

Задача:

Откройте статистические данные по стоимости акций компаний **Apple**, **Microsoft** и **Google** за 12 месяцев 2021 года.

- Постройте **три графика на одной плоскости** (например, с помощью `subplots`) и оцените динамику изменения цен.
- Подумайте, как можно улучшить читаемость результатов:

Для получения данных можно воспользоваться ресурсом:
<https://finance.yahoo.com>

Решение:

In [60]: `import numpy as np`

```

import matplotlib.pyplot as plt

x = np.arange(1, 13, 1)
Apple = [131.96, 121.26, 122.15, 131.46, 124.61, 136.96, 145.86, 151.83, 141.5
Microsoft = [231.96, 232.38, 235.77, 252.17, 249.67, 270.89, 284.91, 301.88, 2
Google = [91.78, 101.84, 103.43, 120.50, 120.57, 125.31, 135.22, 145.46, 133.2

plt.figure(figsize=(12, 7))
plt.grid(alpha=0.3, linestyle='--')
plt.xlabel("Месяц", fontsize=12)
plt.ylabel("Цена акции, $", fontsize=12)
plt.title("Динамика цен акций технологических компаний", fontsize=14, pad=20)

plt.plot(x, Apple, c="r", linewidth=2.5, label="Apple")
plt.fill_between(x, Apple, color="r", alpha=0.1)

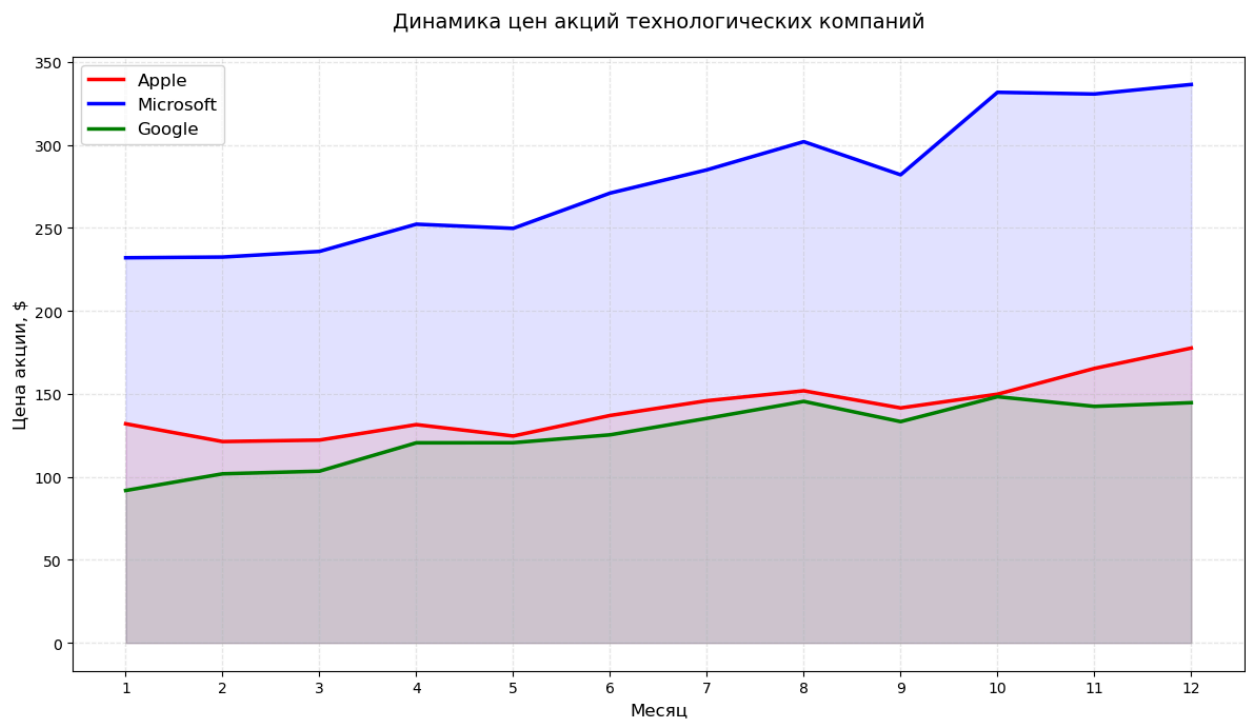
plt.plot(x, Microsoft, c="blue", linewidth=2.5, label="Microsoft")
plt.fill_between(x, Microsoft, color="blue", alpha=0.1)

plt.plot(x, Google, c="green", linewidth=2.5, label="Google")
plt.fill_between(x, Google, color="green", alpha=0.1)

# Настройка легенды и осей
plt.xticks(x)
plt.legend(loc='upper left', fontsize=12)
plt.tight_layout()

plt.show()

```



4.3.5. Задание

Задача:

Создайте простейший калькулятор, включающий основные арифметические действия для двух переменных:

`+`, `-`, `*`, `/`, а также вычисление следующих функций:

- e^{x+y}
- $\sin(x + y)$
- $\cos(x + y)$
- x^y

Решение:

```
In [64]: import math

print("Введите x: \n")
x = float(input())
print("Введите y: \n")
y = float(input())
print("Введите действие:\n",
      "'+' (x + y)\n",
      "'-' (x - y)\n",
      "'*' (x * y)\n",
      "'/' (x / y)\n",
      "'e' (e ^ (x + y))\n",
      "'sin' (sin(x + y))\n",
      "'cos' (cos(x + y))\n",
      "'^' (x ^ y)\n",)
d = str(input())
if d == "+":
    print("x + y = ", x + y)
elif d == "-":
    print("x - y = ", x - y)
elif d == "*":
    print("x * y = ", x * y)
elif d == "/":
    if y == 0:
        print("На ноль делить нельзя!")
    else:
        print("x / y = ", x / y)
elif d == "e":
    print("e ^ (x + y) = ", math.e ** (x + y))
elif d == "sin":
    print("sin(x + y) = ", math.sin(x + y))
elif d == "cos":
    print("cos(x + y) = ", math.cos(x + y))
elif d == "^":
    print("x ^ y = ", x**y)
```

```
else:  
    print("Ошибка!!!")
```

Введите x:

Введите y:

Введите действие:

```
'+' (x + y)  
 '-' (x - y)  
 '*' (x * y)  
 '/' (x / y)  
 'e' (e ^ (x + y))  
 'sin' (sin(x + y))  
 'cos' (cos(x + y))  
 '^' (x ^ y)
```

$e^{(x + y)} = 2.718281828459045$