

СПбГЭТУ “ЛЭТИ”, ФКТИ, МОЭВМ

Информатика. Введение в Python

Модуль Numpy

Для вопросов по курсу: cs@moevm.pro
Иванов Дмитрий Владимирович, dmitry.ivanov@moevm.pro
Бутенко Дмитрий Сергеевич, dmitriy.butenko@moevm.pro
Префикс в теме письма [CS_63XX]

Основные определения

Специальная библиотека для работы с массивами.

Позволяет быстро выполнять математические и статистические операции над массивами.

Часто используется в математике и физике.

Основной объект – однородный и неизменяемый многомерный массив (numpy.ndarray)

Описание библиотеки: https://numpy.org/

Матрицы на основе списков list

```
L = [[i for i in range(5)]
     for j in range(5)]
```

построили матрицу 5 на 5:

выводим матрицу на экран:

```
for item in L:
    print(item)
```

```
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4]
```

Создание объекта ndarray

```
import numpy as np

L = [[i for i in range(5)] for j in range(5)]
arr = np.array(L)
print(type(arr))
print(arr)
```

```
<class 'numpy.ndarray'>
[[0 1 2 3 4]
 [0 1 2 3 4]
 [0 1 2 3 4]
 [0 1 2 3 4]
 [0 1 2 3 4]]
```

Массивы

```
import numpy as np


M = np.random.random((3, 3))
M_min, M_max, M_mean = M.min(), M.max(), M.mean()
sum_x = M.sum(axis=1)
sum_y = M.sum(axis=0)
print(M_min, M_max, M_mean, sum_y, sum_x)

# поиск минимума, максимума, среднего значения
# суммы по строкам axis=1, суммы по столбцам axis=0
```





Атрибуты объекта ndarray

`ndarray.ndim` - число измерений (осей)

`ndarray.shape` - размеры массива (сколько элементов по каждой оси)

`ndarray.size` - количество элементов массива (по всем осям)

`ndarray.dtype` - объект, описывающий тип элементов массива

`ndarray.itemsize` - размер каждого элемента массива в байтах



Атрибуты объекта ndarray. Пример

```
print(arr.ndim)                    2

print(arr.shape)                   (5, 5)

print(arr.size)                    25

print(arr.dtype)                   int64

print(arr.itemsize)                8
```

Тип элементов во время создания объекта ndarray

Параметр по умолчанию dtype может принимать значения:

```
bool_, character, int8, int16, int32, int64, float8,
float16, float32, float64, complex64, object_
```

```
import numpy as np

L = [[i for i in range(5)] for j in range(5)]
arr = np.array(L,dtype=np.int8)
print(arr.itemsize)
print(arr)
```

```
float16
[[0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]]
```

Другие способы создания массивов

```
print(np.zeros((3, 5)))
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

```
print(np.ones((3, 5)))
```

```
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
```

```
print(np.empty((3, 5)))
print(np.eye(3))
```



Тригонометрия над ndarrays

```
a = np.array([20, 30, 40, 50])

np.cos(a)

np.sin(a)

np.tan(a)
```

Другие тригонометрические функции:

Mathematical functions — NumPy v1.21 Manual

Срезы в ndarrays

- Для одномерных массивов
- Взятие среза:

a[3:7]

- Использование среза:

a[3:7] = 8



Особенности работы с массивами



1. Математические операции выполняются над массивами одного размера
2. Математические операции выполняются поэлементно
3. Допустимы математические операции с массивом и числом
4. Всегда создается новый массив (т.к. неизменяемый тип)
5. Можно получить inf как результат операции



Операции с массивами

1. Доступны математические операции **+, -, *, /, **, %**
2. Фильтрация элементов массива с использованием операций сравнения: **>, <, >=, <=, ==, !=**
3. Доступны тригонометрические функции: **numpy.cos(<ndarray>)**
4. Доступны основные операции для работы с задачами линейной алгебры **numpy.linalg**
5. Можно применять срезы **[i:j:k]**

Элементы линейной алгебры

Система линейных алгебраических уравнений (СЛАУ)

– это набор двух или более линейных уравнений, включающих один и тот же набор переменных.

Решение СЛАУ:

$$\begin{cases} 2x + 5y = 1 \\ x - 10y = 3 \end{cases} \rightarrow \begin{pmatrix} 2 & 5 \\ 1 & -10 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \rightarrow$$

```
import numpy

M1 = numpy.array([[2., 5.], [1., -10.]])
v1 = numpy.array([1., 3.])

print(numpy.linalg.solve(M1, v1))
```

Вопросы по курсу задавайте по почте:

Иванов Дмитрий Владимирович
dmitry.ivanov@moevm.pro

Бутенко Дмитрий Сергеевич
dmitriy.butenko@moevm.pro

Правила коммуникации по электронной почте:
https://se.moevm.pro/doku.php/courses:communication_rules