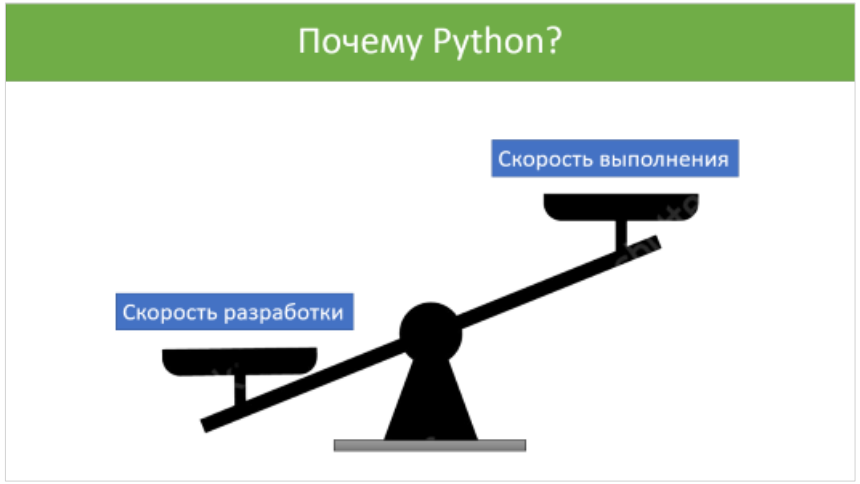
Numpy и компания

Змея выползает на тропу производительности

Скорость – как сделать больше за то же время



Оптимизация алгоритмов

- Обоснованные и необоснованные предположения

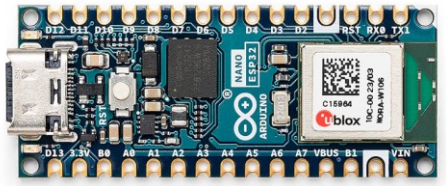
Параллелизация обработки

- Внутри программы
- Средствами ОС/сторонних программ

Сторонние библиотеки

Примеры применения в современной физике

- Обработка (в том числе статистическая) данных, их визуализация
- Численное моделирование – в задачах, менее масштабных, чем high performance computing
  - И даже тут Python хорош для прототипирования
- Символьные вычисления (но не дотягивает до лидеров отрасли)
- Автоматизация эксперимента – например, работа с USB устройствами (датчики etc)
  - MicroPython есть даже для Arduino

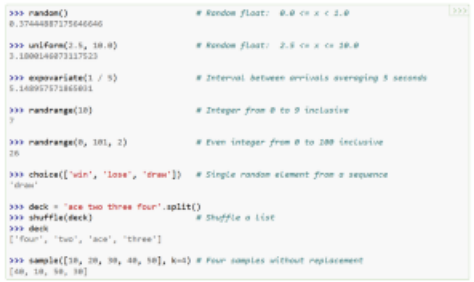


Стандартная библиотека

random

Генерация случайных чисел

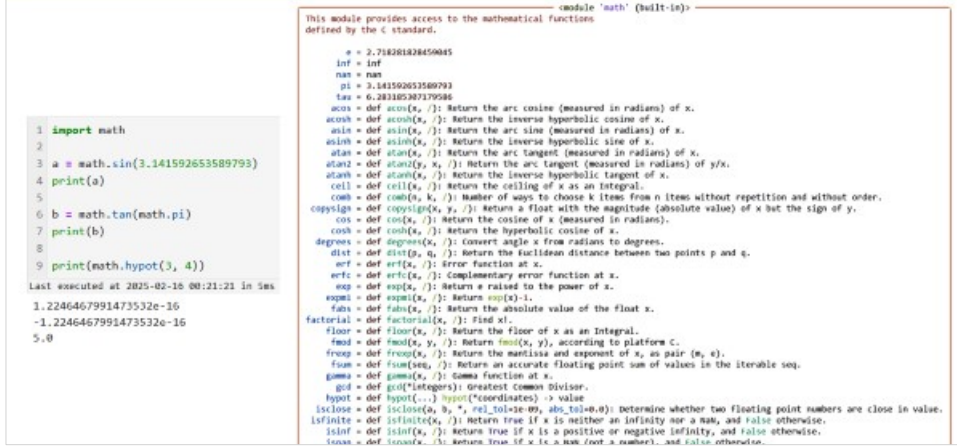
```
random.randrange(stop)
random.randrange(start, stop[, step])
random.randint(a, b)
random.choice(seq)
random.shuffle(x)
random.uniform(a, b)
random.random()
random.choices(population, weights=None, *, cum_weights=None, k=1)
```



https://docs.python.org/3/library/random.html

numpy

math



statistics

Функции для работы со статистическими данными

mean()	Arithmetic mean ("average") of data.
fmean()	Fast, floating-point arithmetic mean, with optional weighting.
geometric_mean()	Geometric mean of data.
harmonic_mean()	Harmonic mean of data.
kde()	Estimate the probability density distribution of the data.
kde_random()	Random sampling from the PDF generated by kde().
median()	Median (middle value) of data.
median_low()	Low median of data.
median_high()	High median of data.
median_grouped()	Median (50th percentile) of grouped data.
mode()	Single mode (most common value) of discrete or nominal data.
multimode()	List of modes (most common values) of discrete or nominal data.
quantiles()	Divide data into intervals with equal probability.

covariance()	Sample covariance for two variables.
correlation()	Pearson and Spearman's correlation coefficients.
linear_regression()	Slope and intercept for simple linear regression.

pstdev()	Population standard deviation of data.
pvariance()	Population variance of data.
stdev()	Sample standard deviation of data.
variance()	Sample variance of data.

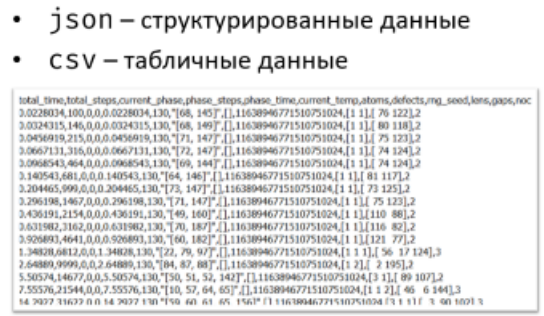
https://docs.python.org/3/library/statistics.html

numpy

Хранение данных (в человекочитаемом виде)

Хранение данных в **одновременно** машино- и человекочитаемом виде

- `json` – структурированные данные
- `csv` – табличные данные



yaml

pandas

numpy

https://docs.python.org/3/library/csv.html
https://docs.python.org/3/library/json.html

Numpy

Numpy = Numerical Python (численный Python)

Numpy – это фундаментальная библиотека для научных вычислений с Python. Она предоставляет интерфейс для работы с многомерными массивами (ndarray, N-dimensional array), а также с многими другими производными объектами (включая массивы с масками и матрицы), а также большое число разнообразных функций, осуществляющих быстрые операции над ними: математические, логические, манипулирующие формой массивов, осуществляющие сортировку, выбор, ввод/вывод, различные преобразования, базовую линейную алгебру, базовые статистические операции, генерацию случайных объектов и многое другое.

- Numpy стремится к эффективности за счет работы 'in-memory'.
- Критически важно: циклы for в Python очень медленные.
- Реализована на C и Fortran, вычисления векторизованы.

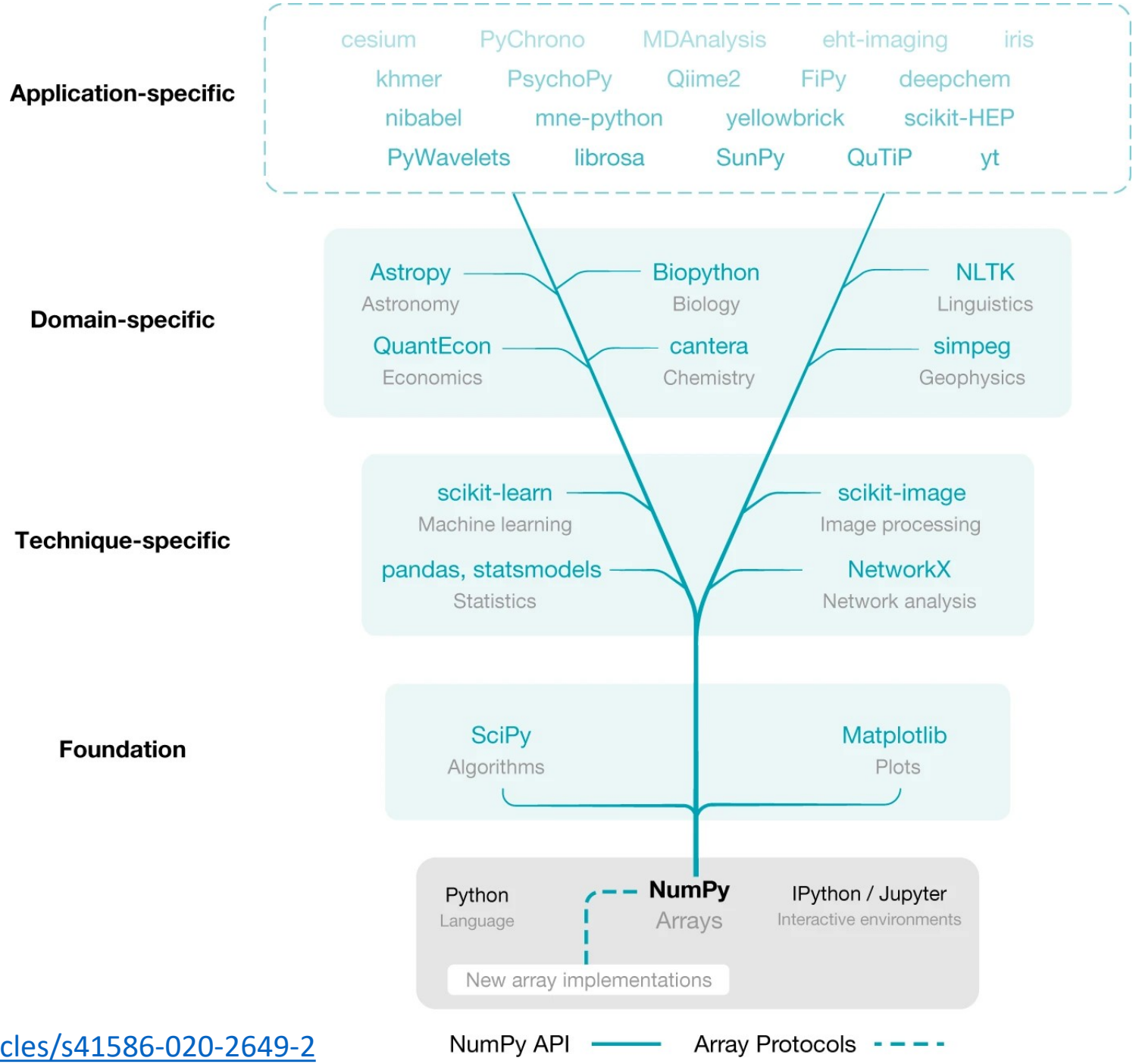
Numpy – основание научной экосистемы Python

Используется почти во всех областях науки и технологии

Универсальный стандарт

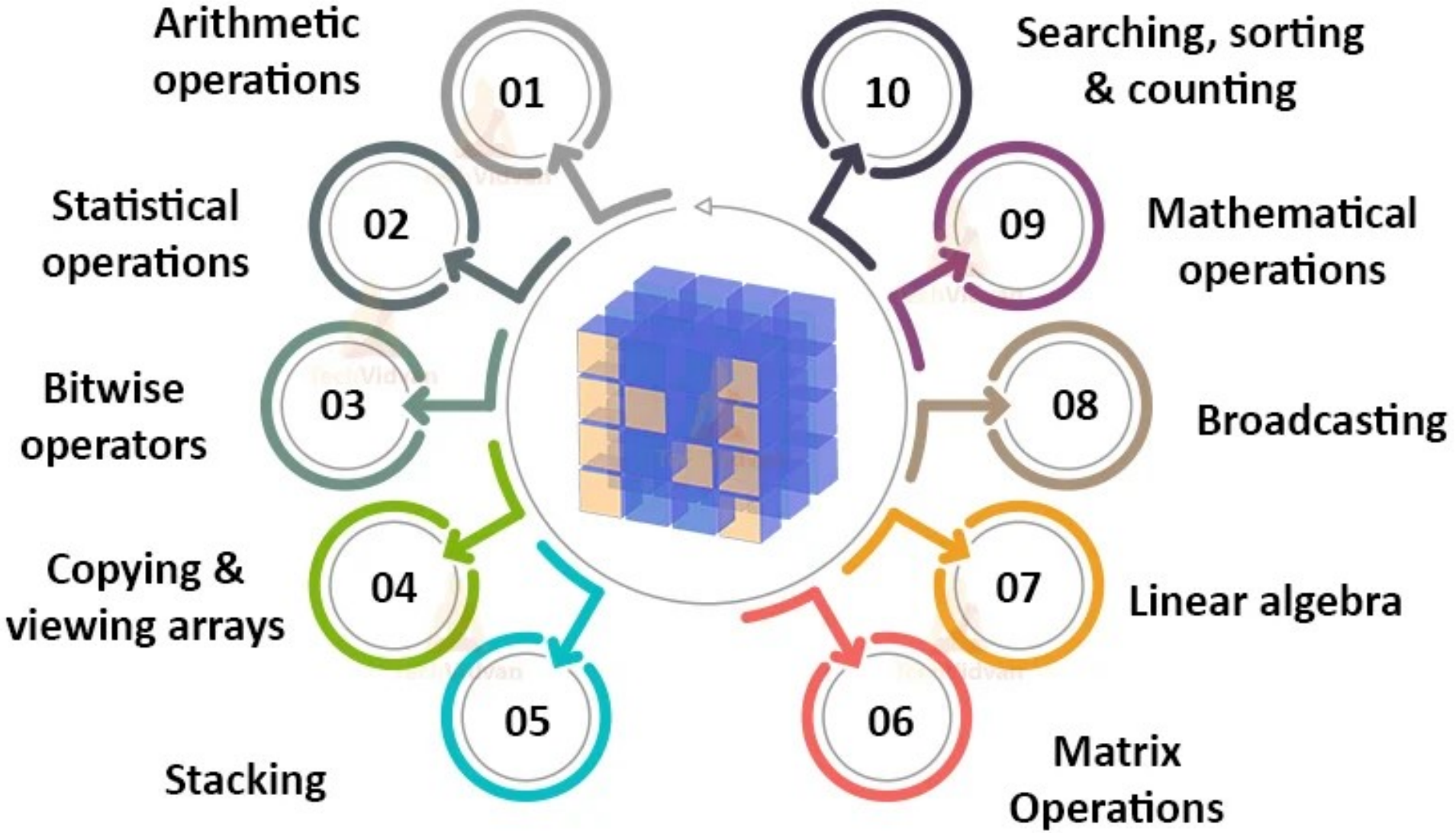
Предоставляет основу для создания других научных пакетов

- Науки о Земле: GeoPandas, MovingPandas, Xarray
- Космос: SpacePy, AstroPy, SunPy
- Scientific Domain: SciPy, SimPy, Shapely
- Data Science: Pandas, Dask
- Machine Learning: Scikit-Learn, TensorFlow
- Визуализация: Matplotlib, Cartopy, Seaborn, HoloViews



https://www.nature.com/articles/s41586-020-2649-2

Uses of NumPy



https://colab.research.google.com/github/astg606/astg_pymaterials/blob/main/numpy/understand_numpy.ipynb

Базовые идеи

ndarray - это гомогенный объект, n-мерный массив + методы для работы с ним: это структура (данные + метаданные).

Индексация как обычно: срезы и индексы.

Индексация при помощи скалярных координат, масок или массивов координат, при этом еще и дёшево.

Эффективная параллелизация операций за счет векторизации.

Транслирование массивов

Редукция вдоль одной и более осей.

Документация

https://numpy.org/doc/stable/user/index.html

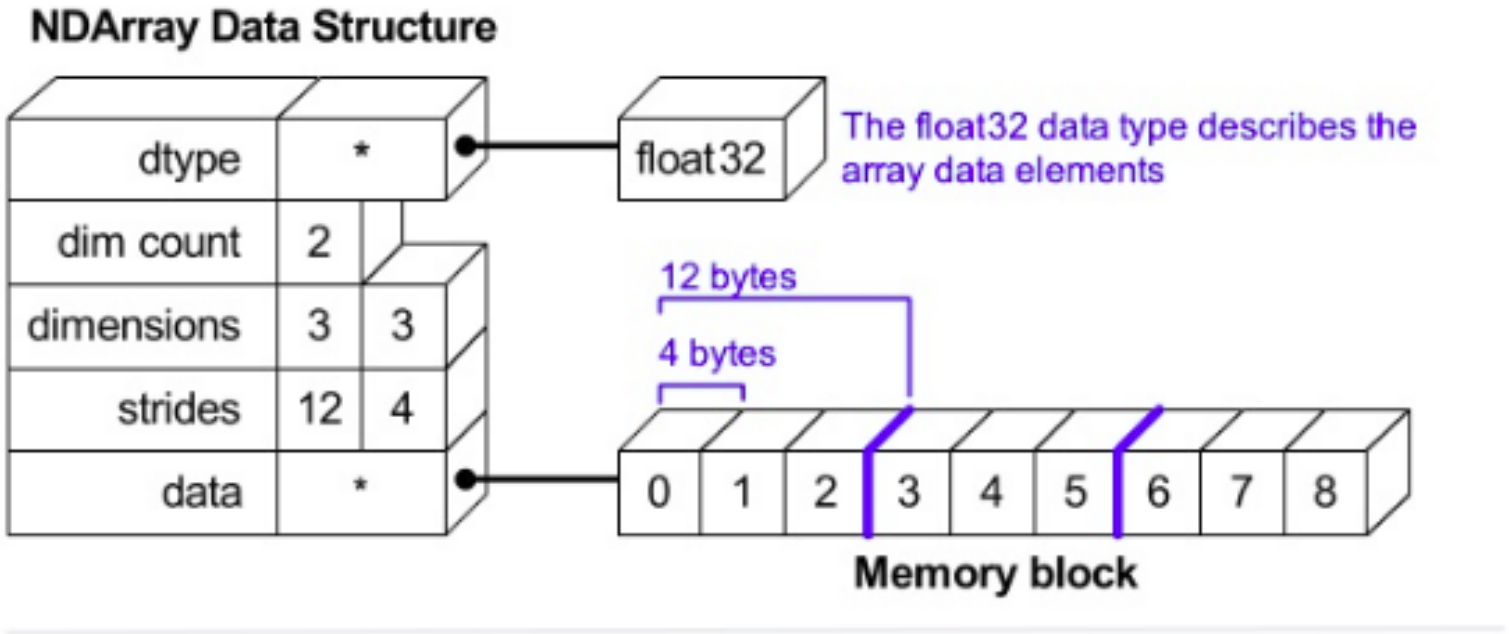
https://numpy.org/doc/stable/user/quickstart.html

https://numpy.org/doc/stable/user/absolute_beginners.html

Fundamentals and usage

- NumPy fundamentals
  - Array creation
  - Indexing on `ndarrays`
  - I/O with NumPy
  - Data types
  - Broadcasting
  - Copies and views
  - Working with Arrays of Strings And Bytes
  - Structured arrays
  - Universal functions (`ufunc`) basics

Что такое ndarray?



Python View :

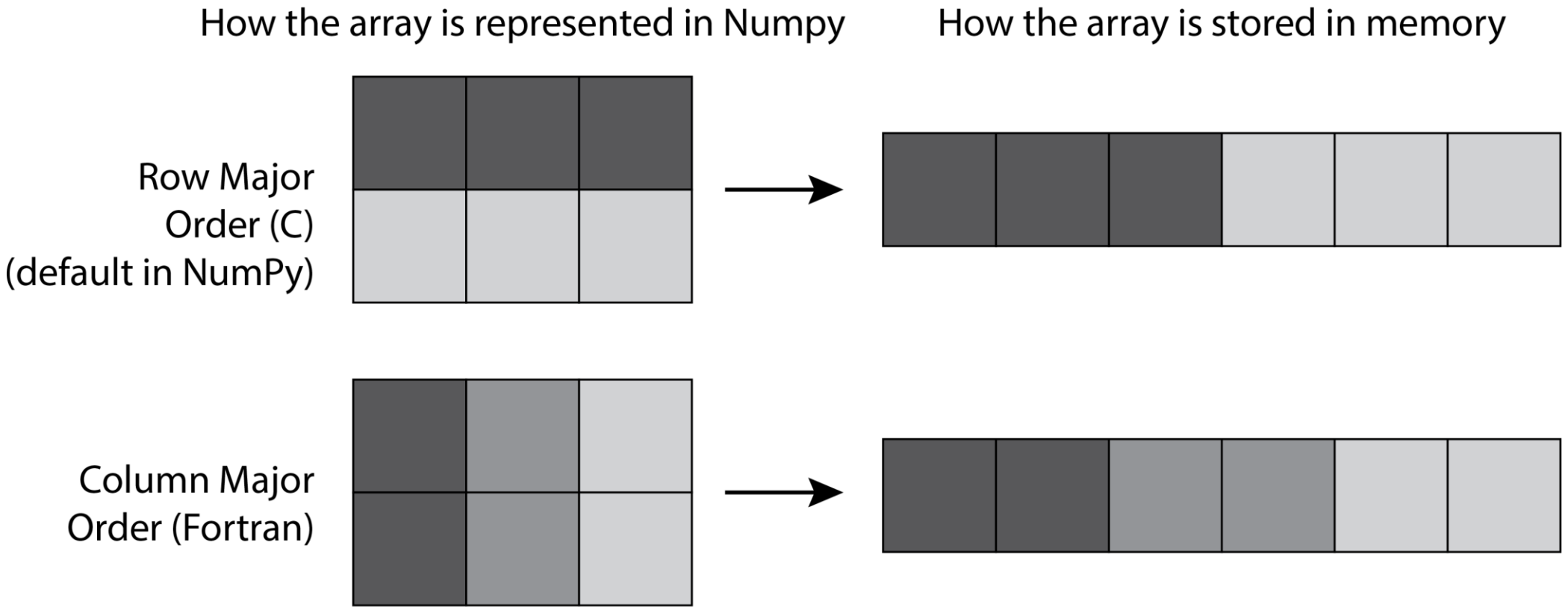
0	1	2
3	4	5
6	7	8

Преимущества

Изменяя метаданные, можно менять
- интерпретацию данных,
- порядок следования данных

без использования промежуточного буфера.

C-order vs F-order



Начало работы

```
import numpy as np
```

```
a = [[1,2,0],[3,8,7]]

b = np.array(a, dtype=float)
print(b, b.dtype, b.shape, b.size)
```

```
[[1. 2. 0.]
 [3. 8. 7.]] float64 (2, 3) 6
```

Создание массивов с нуля

numpy.array

numpy.empty
numpy.ones
numpy.zeros
numpy.full

numpy.empty_like
numpy.ones_like
numpy.zeros_like
numpy.full_like

numpy.eye
numpy.identity

numpy.arange
numpy.linspace
numpy.logspace
numpy.geomspace
numpy.meshgrid

numpy.diag
numpy.tri
numpy.tril
numpy.triu

```
>>> np.empty([2, 2], dtype=int)
array([[-1073741821, -1067949133],
       [  496041986,    19249760]])
```

```
>>> np.ones((5,), dtype=int)
array([1, 1, 1, 1, 1])
```

```
>>> np.zeros((2, 1))
array([[ 0.],
       [ 0.]])
```

```
>>> np.full((2, 2), np.inf)
array([[inf, inf],
       [inf, inf]])
>>> np.full((2, 2), 10)
array([[10, 10],
       [10, 10]])
```

```
>>> np.full((2, 2), [1, 2])
array([[1, 2],
       [1, 2]])
```

```
>>> np.eye(3, k=1)
array([[0.,  1.,  0.],
       [0.,  0.,  1.],
       [0.,  0.,  0.]])
```

```
>>> y = np.arange(3, dtype=float)
>>> y
array([0., 1., 2.])
>>> np.ones_like(y)
array([1.,  1.,  1.])
```

```
>>> np.arange(3)
array([0, 1, 2])
>>> np.arange(3.0)
array([ 0.,  1.,  2.])
>>> np.arange(3,7)
array([3, 4, 5, 6])
>>> np.arange(3,7,2)
array([3, 5])
```

```
>>> np.linspace(2.0, 3.0, num=5)
array([2.  , 2.25, 2.5 , 2.75, 3.  ])
```

```
>>> x = np.arange(9).reshape((3,3))
>>> x
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
>>> np.diag(x)
array([0, 4, 8])
```

numpy.ones(*shape, dtype=None, order='C', *, device=None, like=None*)

Чтение данных в массив

numpy.array

numpy.copy
numpy.frombuffer
numpy.from_dlpack
numpy.fromfile
numpy.fromfunction
numpy.fromiter
numpy.fromstring
numpy.loadtxt

numpy.load
numpy.save

```
>>> x = np.array([1, 2, 3])
>>> y = x
>>> z = np.copy(x)

>>> x[0] = 10
>>> x[0] == y[0]
True
>>> x[0] == z[0]
False
```

```
>>> np.fromfunction(lambda i, j: i + j, (3, 3), dtype=int)
array([[0, 1, 2],
       [1, 2, 3],
       [2, 3, 4]])
```

```
>>> np.fromstring('1 2', dtype=int, sep=' ')
array([1, 2])
>>> np.fromstring('1, 2', dtype=int, sep=',')
array([1, 2])
```

```
>>> import numpy as np
>>> from io import StringIO
>>> c = StringIO("0 1\n2 3")
>>> np.loadtxt(c)
array([[0., 1.],
       [2., 3.]])
```

```
>>> np.save('/tmp/123', np.array([[1, 2, 3], [4, 5, 6]]))
>>> np.load('/tmp/123.npy')
array([[1, 2, 3],
       [4, 5, 6]])
```

Простейшие операции с данными

Базовые операции
Изменение формы
Операции типа транспонирования
Изменение числа измерений
Изменение типа массива
Объединение массивов
Деление массивов на части
Повторение массивов
Добавление/удаление элементов
Перестановка элементов

`copyto` (dst, src[, casting, where])

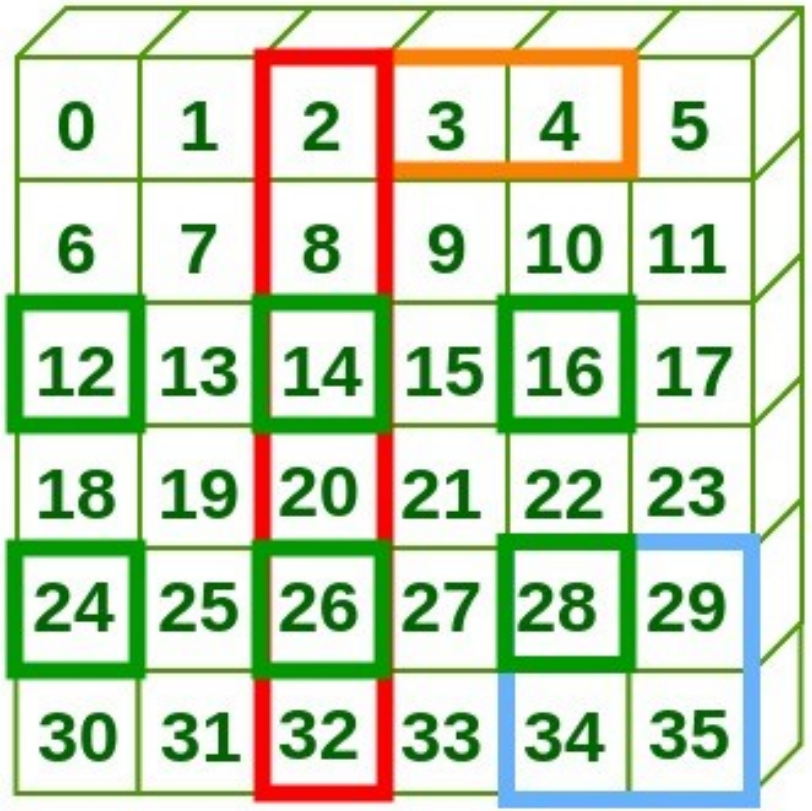
`ndim` (a)

`shape` (a)

`size` (a[, axis])

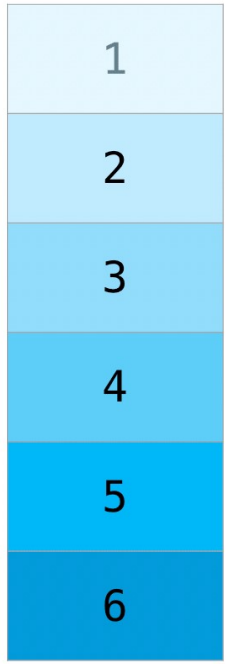
Доступ по индексам etc

```
>>> a[0,3:5]
array( [3,4] )

>>> a[4:, 4:]
array( [ 28, 29],
       [ 34, 35] ] )

>>> a[ :, 2]
array( [2, 8, 14, 20, 26, 32] )

>>> a[2 : : 2, : : 2]
array( [ 12, 14, 16],
       [ 24, 26, 28] ] )
```

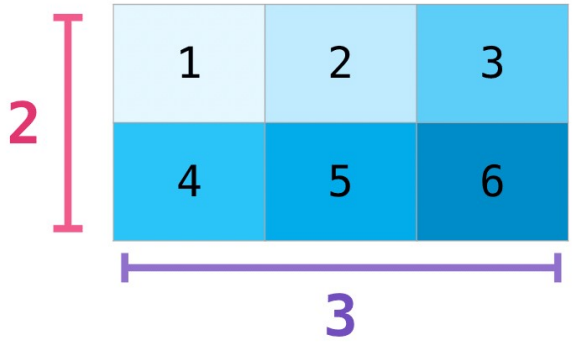


Изменение формы массива

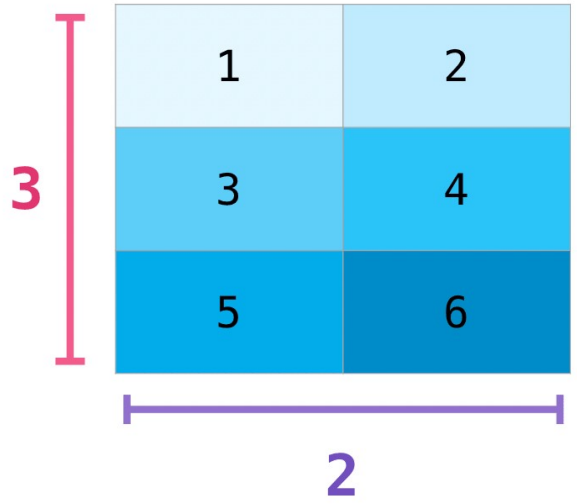
data



data.reshape(2,3)



data.reshape(3,2)



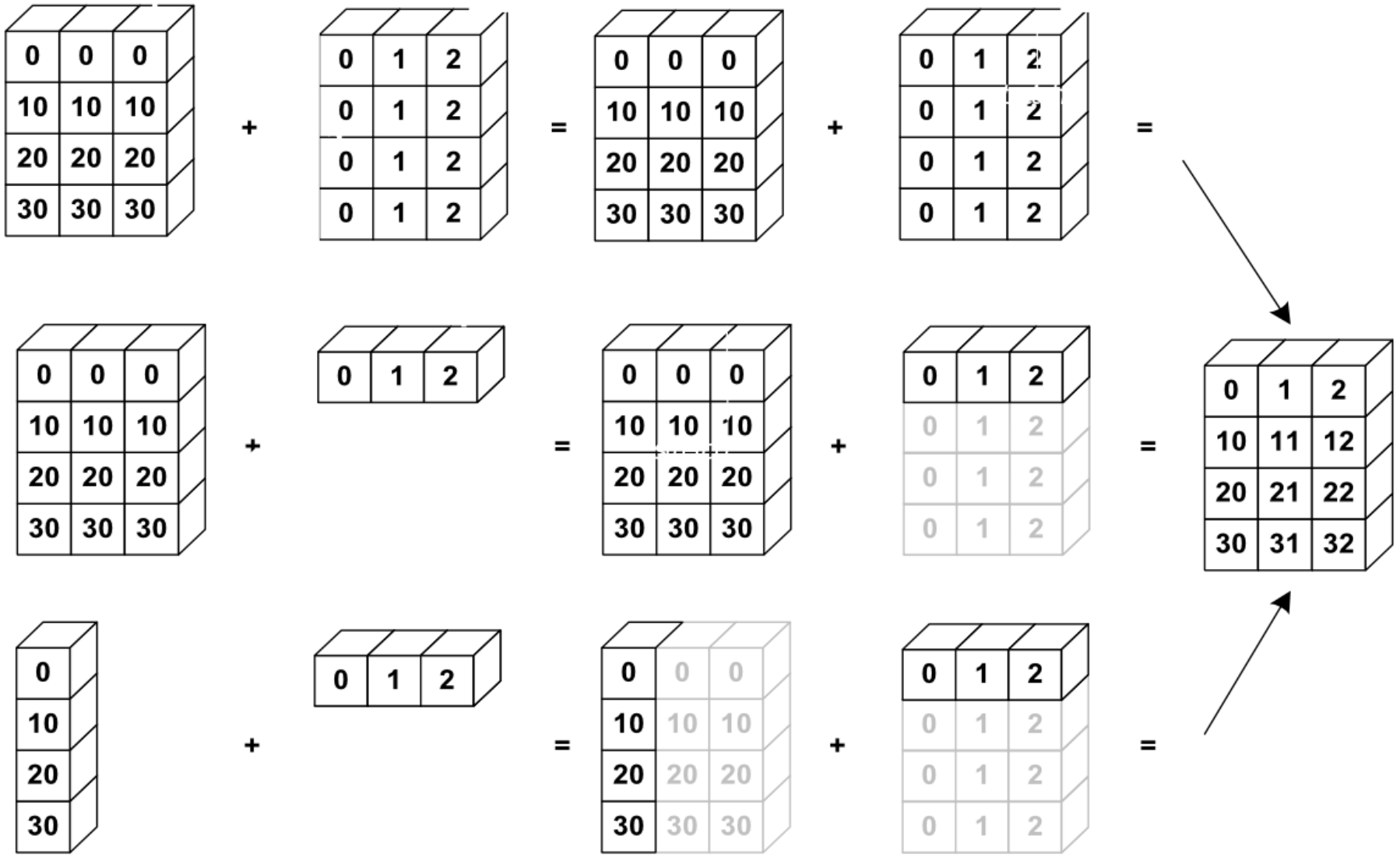
Транслирование массивов (broadcasting)

Если не все входные массивы имеют одинаковое число измерений, то к тем, у кого размерность ниже, к их форме спереди добавляется единица, пока размерности всех массивов не станут одинаковыми.

Одномерные массивы длиной, равной 1, вдоль некоторой оси ведут себя так, как если бы они имели вдоль этой оси такой же размер, как и другие массивы, входящие в выражение. Каждое значение в массиве вдоль этой оси будет равно тому единственному, которое там было изначально.

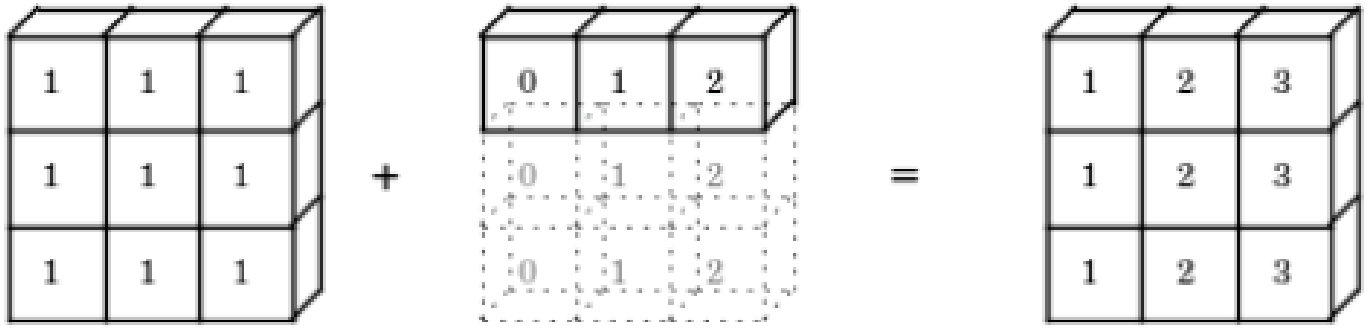
Если эти два правила не выполняются, генерируется ошибка.

Транслирование массивов (broadcasting)

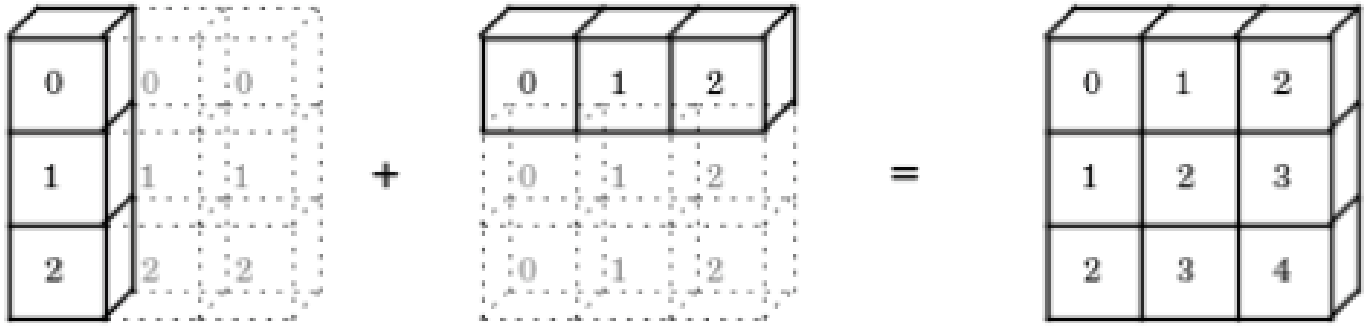


`np.arange(3) + 5`

`np.ones((3, 3)) + np.arange(3)`

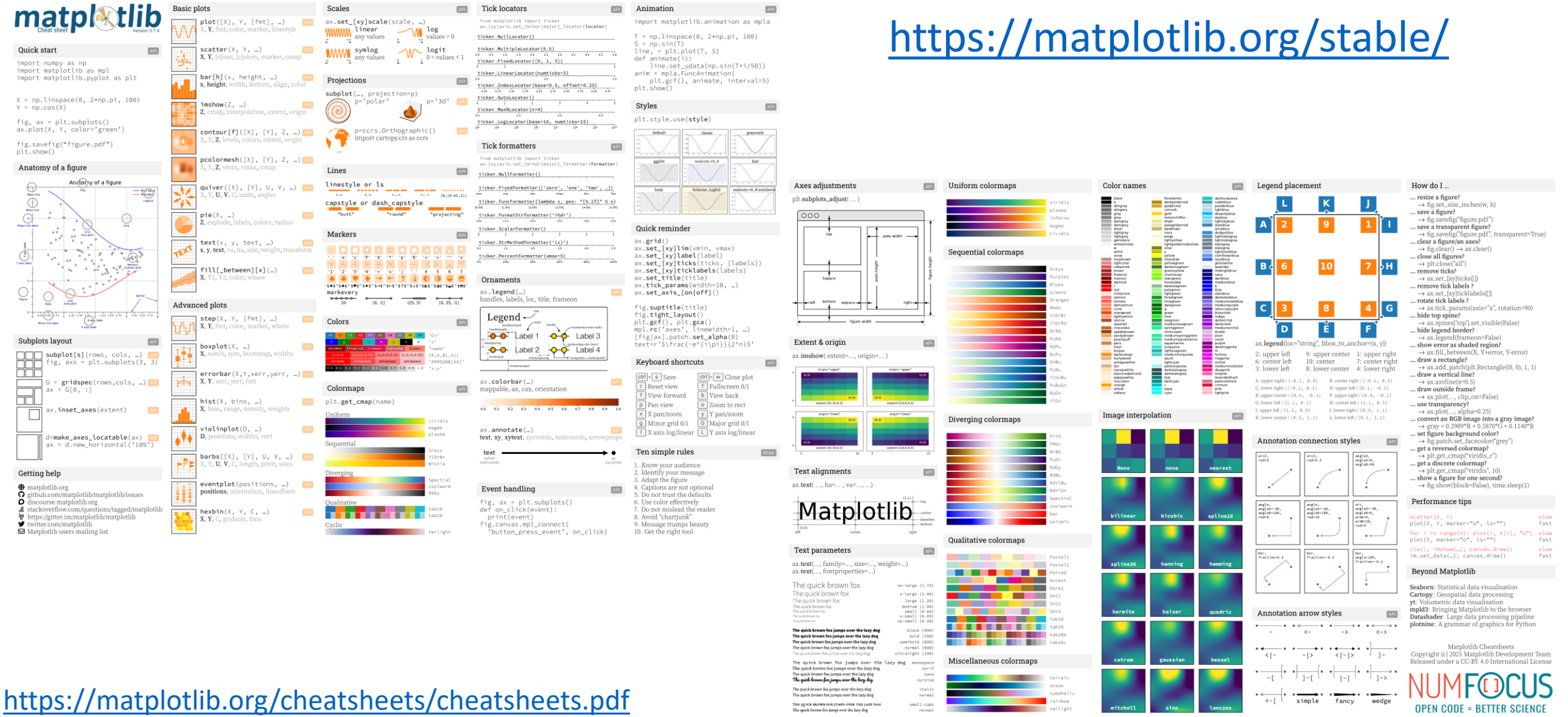


`np.arange(3).reshape((3, 1)) + np.arange(3)`

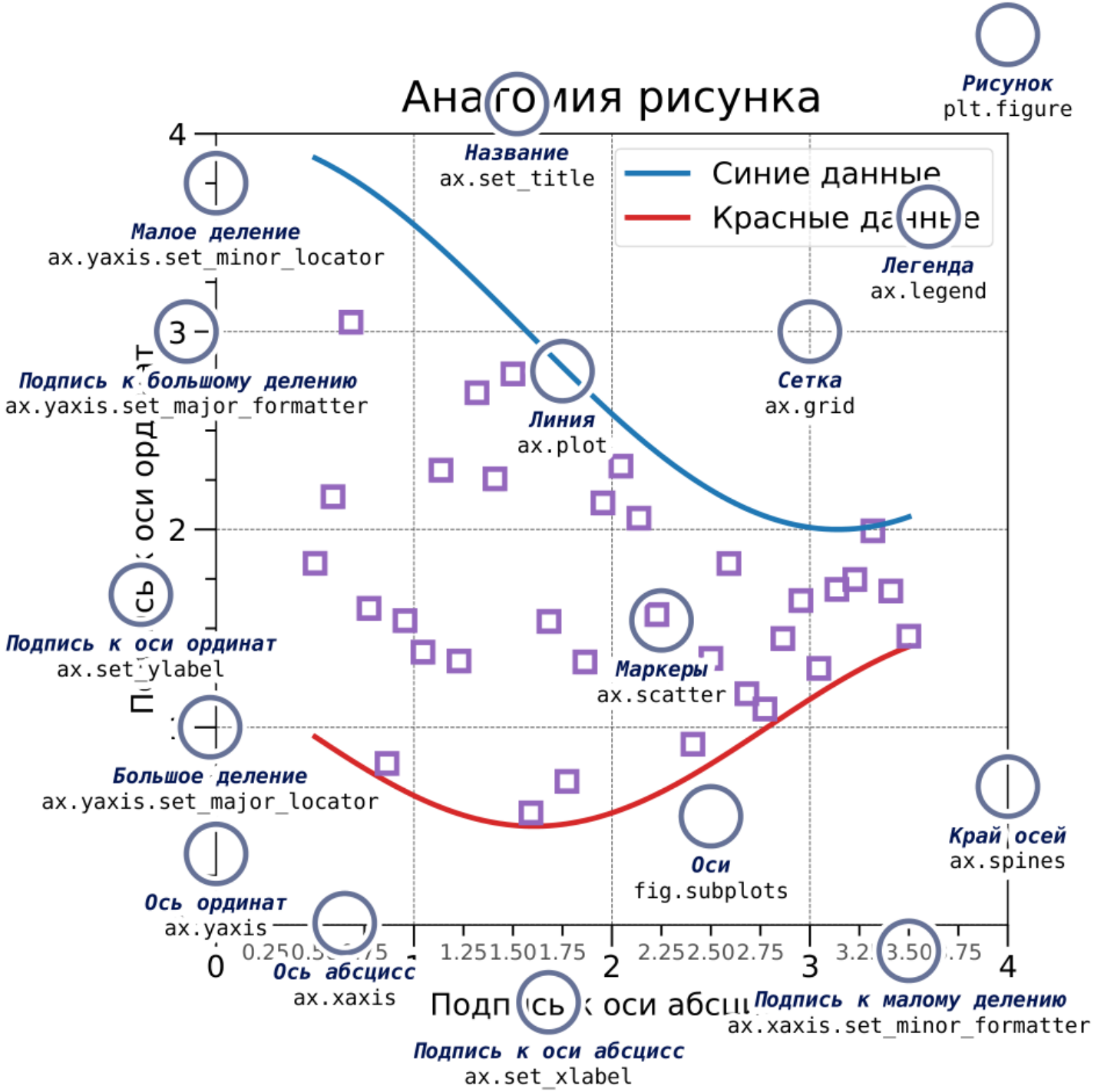


Matplotlib

https://matplotlib.org/stable/

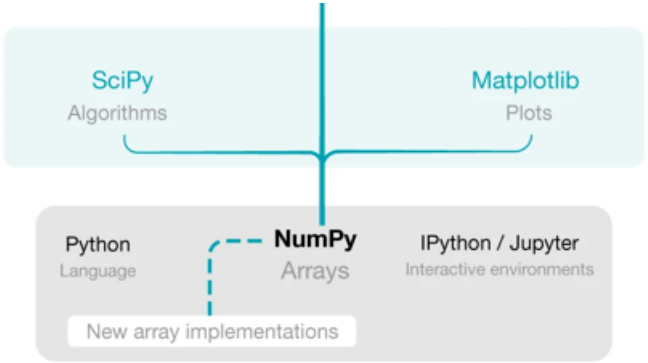


https://matplotlib.org/cheatsheets/cheatsheets.pdf


Анатомия рисунка
Рисунок
plt.figure
Название
ax.set_title
Синие данные
Красные данные
Легенда
ax.legend
Малое деление
ax.yaxis.set_minor_locator
Подпись к большому делению
ax.yaxis.set_major_formatter
Сетка
ax.grid
Линия
ax.plot
Подпись к оси ординат
ax.set_ylabel
Маркеры
ax.scatter
Большое деление
ax.yaxis.set_major_locator
Края осей
ax.spines
Оси
fig.subplots
Ось ординат
ax.yaxis
Ось абсцисс
ax.xaxis
Подпись к малому делению
ax.xaxis.set_minor_formatter
Подпись к оси абсцисс
ax.set_xlabel
0
1
2
3
4

Scipy

https://docs.scipy.org/doc/scipy/tutorial/index.html



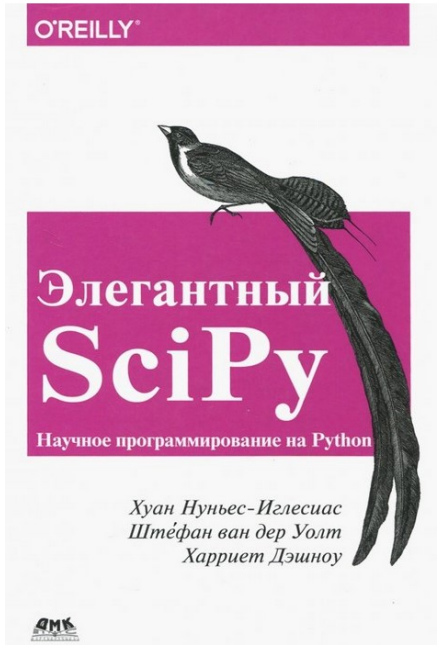
Subpackage	Description and User Guide
cluster	Clustering algorithms
constants	Physical and mathematical constants
differentiate	Finite difference differentiation tools
fft	Fourier Transforms (scipy.fft)
fftpack	Fast Fourier Transform routines (legacy)
integrate	Integration (scipy.integrate)
interpolate	Interpolation (scipy.interpolate)
io	File IO (scipy.io)
linalg	Linear Algebra (scipy.linalg)
ndimage	Multidimensional Image Processing (scipy.ndimage)
odr	Orthogonal distance regression
optimize	Optimization (scipy.optimize)
signal	Signal Processing (scipy.signal)
sparse	Sparse Arrays (scipy.sparse)
spatial	Spatial Data Structures and Algorithms (scipy.spatial)
special	Special Functions (scipy.special)
stats	Statistics (scipy.stats)

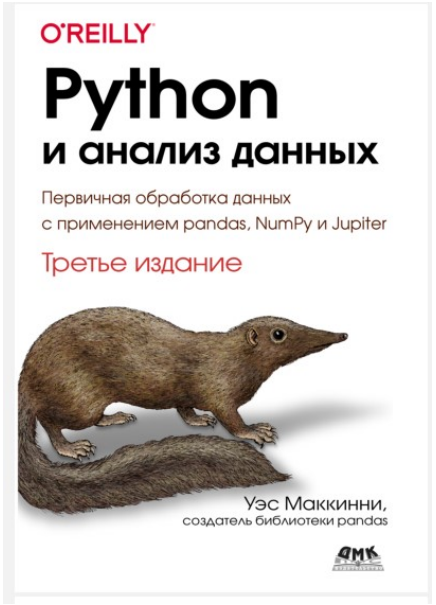
Дальнейшее чтение

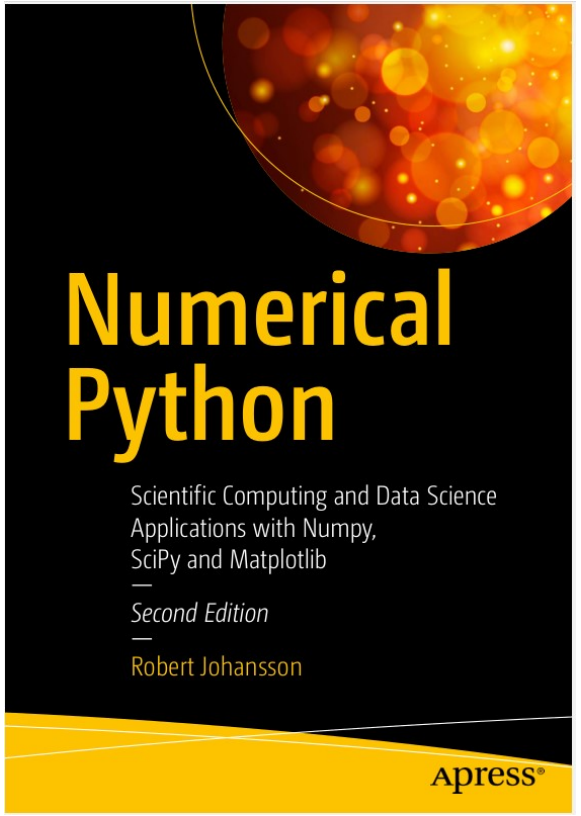
Маккинни, У. *Python и анализ данных* 3 издание. 536 с. isbn: 978-6-01810-347-6 (ДМК Пресс, 2023).

Нуньес-Иглесиас, Х., Уолт, Ш., Дэшноу, Х. *Элегантный Scipy*. 266 с. isbn: 978-5-97060-600-1 (ДМК Пресс, 2018).

Johansson, R. *Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib* isbn: 9781484242469. http://dx.doi.org/10.1007/978-1-4842-4246-9 (Apress, 2019).







https://mathstat.dal.ca/~brown/sound/python/Numerical%20Python.pdf