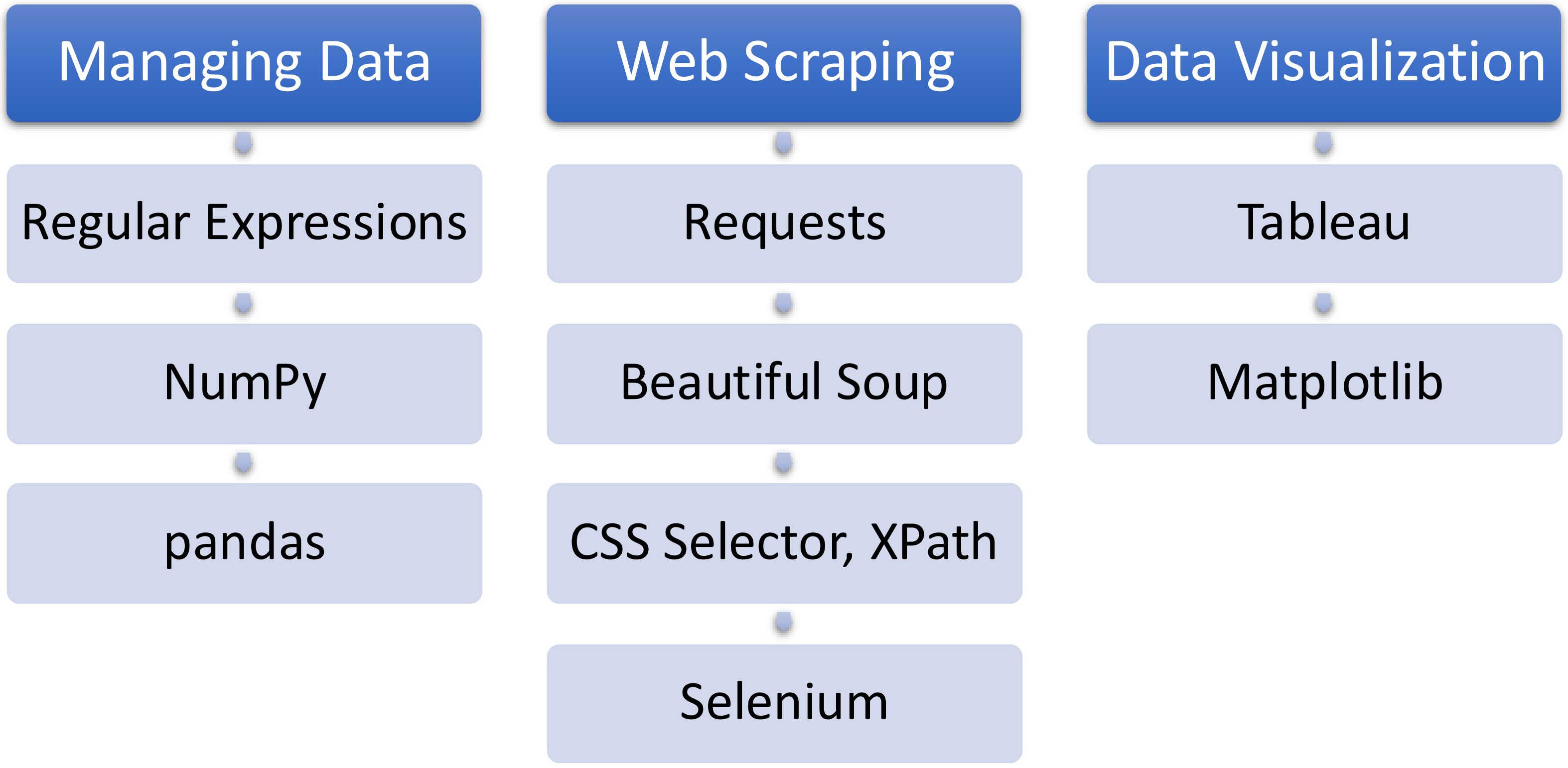
Managing Data II

MSBA7001 Business Intelligence and Analytics
HKU Business School
The University of Hong Kong

Instructor: Dr. DING Chao

Course Roadmap



Agenda

- SciPy
- NumPy

SciPy

What is SciPy?

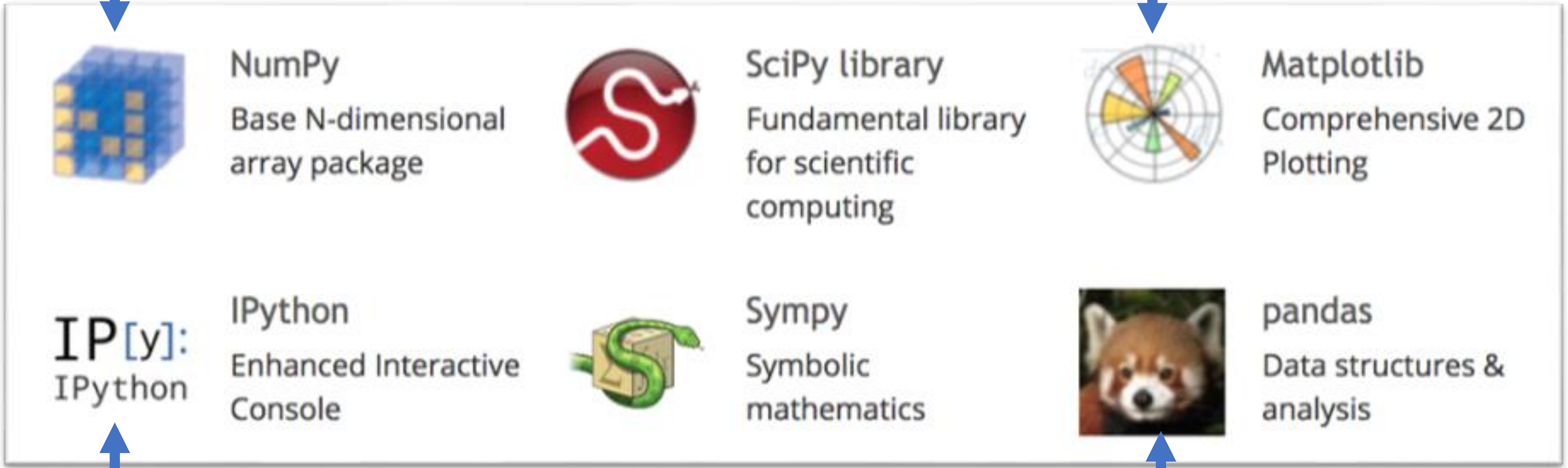
- SciPy (pronounced /saɪpaɪ/) is a Python-based ecosystem of open-source software for mathematics, science, and engineering.
- The SciPy ecosystem includes general and specialized tools for data management and computation, productive experimentation and high-performance computing.
- It offers over 1000 modules/packages for Python.

SciPy.org Sponsored By ENTHOUGHT

The SciPy Ecosystem

It defines numerical array and matrix types

It provides data visualization tools



NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis

It makes possible Jupyter Notebook

It provides high-performance, easy to use data structures

NumPy

What is the problem with lists?

- Lists are ok for storing small amounts of one-dimensional data.
- But, we can’t use them directly with arithmetical operators such as +, -, *, /, …

```
[1, 2, 4] - 10
```

```
TypeError: unsupported operand type(s) for -: 'list' and 'int'
```

- We need efficient arrays with arithmetic and better multidimensional tools.

What is NumPy?

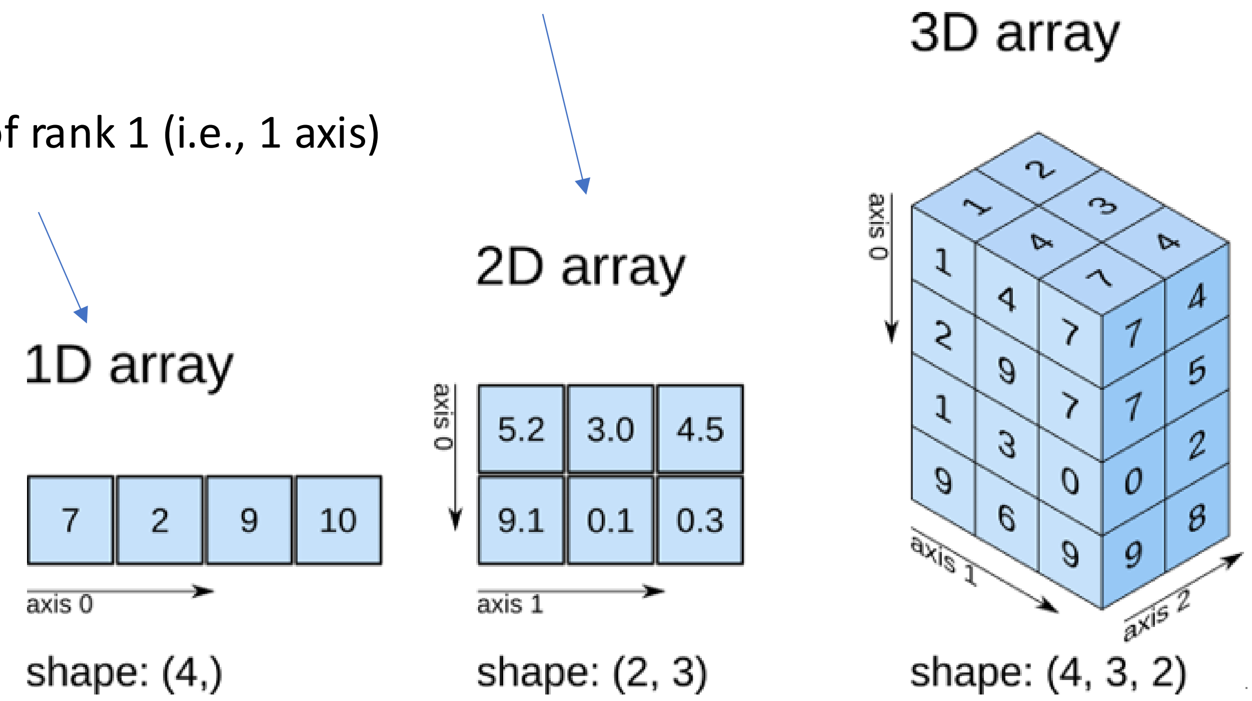
- **NumPy** (/nʌmpaɪ/), short for Numerical Python, is the fundamental package required for high performance scientific computing and data analysis.
- It provides:
  - ➢ Arrays, a fast and space-efficient multidimensional array providing vectorized arithmetic operations and sophisticated broadcasting capabilities
  - ➢ Standard mathematical functions for fast operations on entire arrays of data without having to write loops
  - ➢ Tools for reading / writing array data to disk and working with memory-mapped files
  - ➢ Linear algebra, random number generation, and Fourier transform capabilities

The NumPy Arrays

- An array is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers.
- Dimensions are usually called axes, the number of axes is the rank.

An array of rank 2 (i.e., 2 axes)

An array of rank 1 (i.e., 1 axis)



Creating Arrays

- The **array** method accepts any sequence-like object (e.g., list, tuple) and produces a new array.

```
import numpy as np
data1 = [6, 7.5, 8, 1]
arr1 = np.array(data1, dtype = np.float32)
arr1
```

```
array([6. , 7.5, 8. ,  1. ])
```

Data type	Description
int, float, str	Integer, float, and string, respectively
np.int64	Signed 64-bit integer types
np.float32	Standard double-precision floating point
np.complex	Complex numbers represented by 128 floats
np.bool	Boolean type storing TRUE and FALSE values
np.object	Python object type
np.string_	Fixed-length string type

Array Properties

Property	Description	Example	
arr.size	Returns number of elements in arr	arr2.size	6
arr.shape	Returns dimensions of arr (rows,columns)	arr2.shape	(2,4)
arr.ndim	Returns the dimension of arr	arr2.ndim	2
arr.dtype	Returns type of elements in arr	arr2.dtype	int32
np.info(arr)	View documentation for arr		

```
data2 = [[1, 2, 3, 4], [5, 6, 7, 8]]
arr2 = np.array(data2)
print(arr2)
```

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

Creating Special Arrays

Method	Description
np.zeros(3)	1D array of length 3 all values 0
np.ones((3,4))	3x4 array with all values 1
np.eye(5)	5x5 array of 0 with 1 on diagonal (Identity matrix)
np.empty((2,3,2))	2x3x2 array without initializing its values to any particular value
np.full((2,3),8)	2x3 array with all values 8
np.linspace(0,100,6)	Array of 6 evenly divided values from 0 to 100
np.arange(0,10,3)	Array of values from 0 to less than 10 with step 3

```
np.zeros(5)
```

```
array([0., 0., 0., 0., 0.])
```

```
np.full((2,3), 4)
```

```
array([[4, 4, 4],
       [4, 4, 4]])
```

```
np.eye(3)
```

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

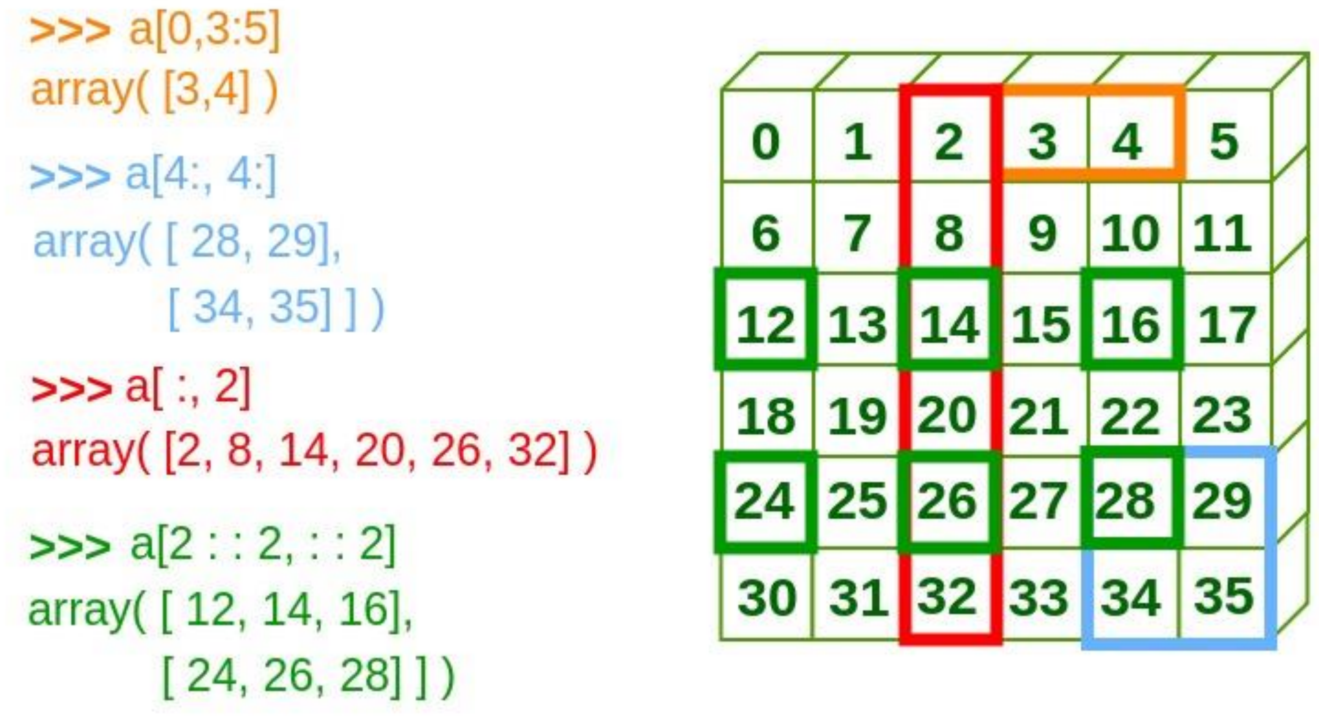
Creating Random Arrays

Method	Description
np.random.rand(4,5)	4x5 array of random floats between 0–1
np.random.rand(6,7)*100	6x7 array of random floats between 0–100
np.random.randint(5,size=(2,3))	2x3 array with random ints between 0–4
np.random.choice([3,5,7,9], size=(3,5))	3x5 array randomly drawn from the list
np.random.randn(5, 3)	5x3 array drawn from a standard normal distribution
np.random.normal(mu, sigma, 10)	1x10 array drawn from a normal distribution

- For a full list of available distributions, see

https://numpy.org/doc/stable/reference/random/legacy.html

Basic Array Indexing and Slicing



```
>>> 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] ] )
```

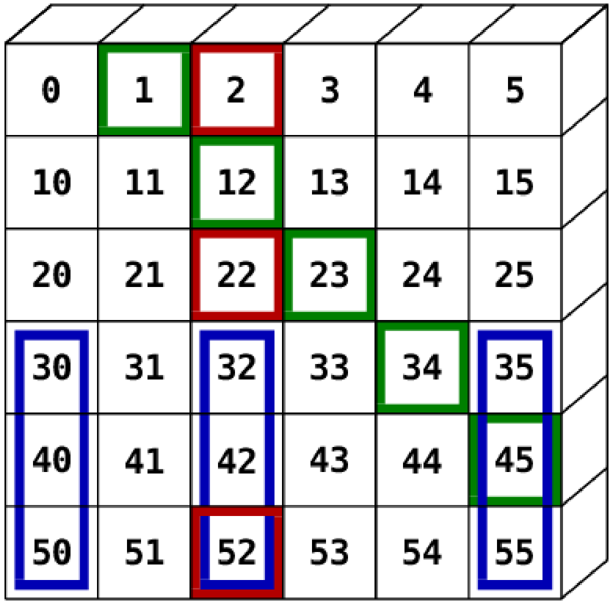
source: www.geeksforgeeks.org

Fancy Indexing

```
>>> a[(0,1,2,3,4), (1,2,3,4,5)]
array([1, 12, 23, 34, 45])

>>> a[3:, [0,2,5]]
array([[30, 32, 35],
       [40, 42, 45],
       [50, 52, 55]])

>>> mask = np.array([1,0,1,0,0,1], dtype=bool)
>>> a[mask, 2]
array([2, 22, 52])
```



source: www.scipy-lectures.org

Array Operations

Method	Description
np.copy(arr)	Copies arr to new memory
arr.view(dtype)	Creates view of arr elements with type dtype
np.append(arr,values)	Appends values to end of arr
np.insert(arr,2,values)	Inserts values into arr before index 2
np.delete(arr,3,axis=0)	Deletes row on index 3 of arr
np.isnan(arr)	Checks for nan values and returns Boolean results.
np.argwhere(arr)	Finds the indices of array elements that are non-zero
arr.fill(value)	Fills the array with scalar values.
arr.astype(np.int64)	Converts arr elements to type np.int64
arr.tolist()	Converts arr to a Python list

Maths

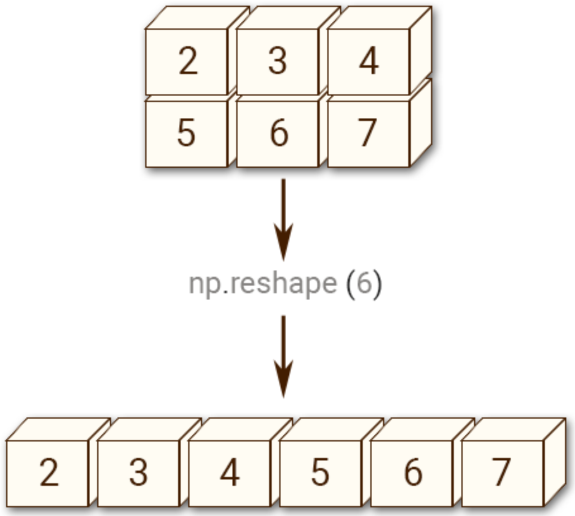
Method	Description
np.add(arr1,arr2)	Elementwise add arr2 to arr1
np.subtract(arr1,arr2)	Elementwise subtract arr2 from arr1
np.multiply(arr1,arr2)	Elementwise multiply arr1 by arr2
np.divide(arr1,arr2)	Elementwise divide arr1 by arr2
np.power(arr1,arr2)	Elementwise raise arr1 raised to the power of arr2
np.array_equal(arr1,arr2)	Returns True if the arrays have the same elements and shape
np.sqrt(arr)	Square root of each element in the array
np.sin(arr)	Sine of each element in the array
np.log(arr)	Natural log of each element in the array
np.abs(arr)	Absolute value of each element in the array
np.round(arr)	Rounds to the nearest int

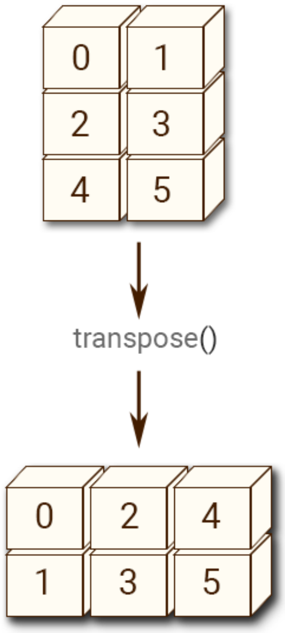
Statistics

Method	Description
arr.mean(arr,axis=0)	Returns mean along specific axis
arr.sum()	Returns sum of arr
arr.min()	Returns minimum value of arr
arr.max(axis=0)	Returns maximum value of specific axis
np.var(arr)	Returns the variance of array
np.std(arr,axis=1)	Returns the standard deviation of specific axis
np.corrcoef(arr1,arr2)	Returns correlation coefficient of arr1 and arr2

Array Transformations

Method	Description
arr.sort(axis=0)	Sorts specific axis of arr
arr.T or arr.transpose()	Transposes arr (rows become columns and vice versa)
arr.reshape(3,4)	Reshapes arr to 3 rows, 4 columns without changing data
arr.ravel()	Flattens the array

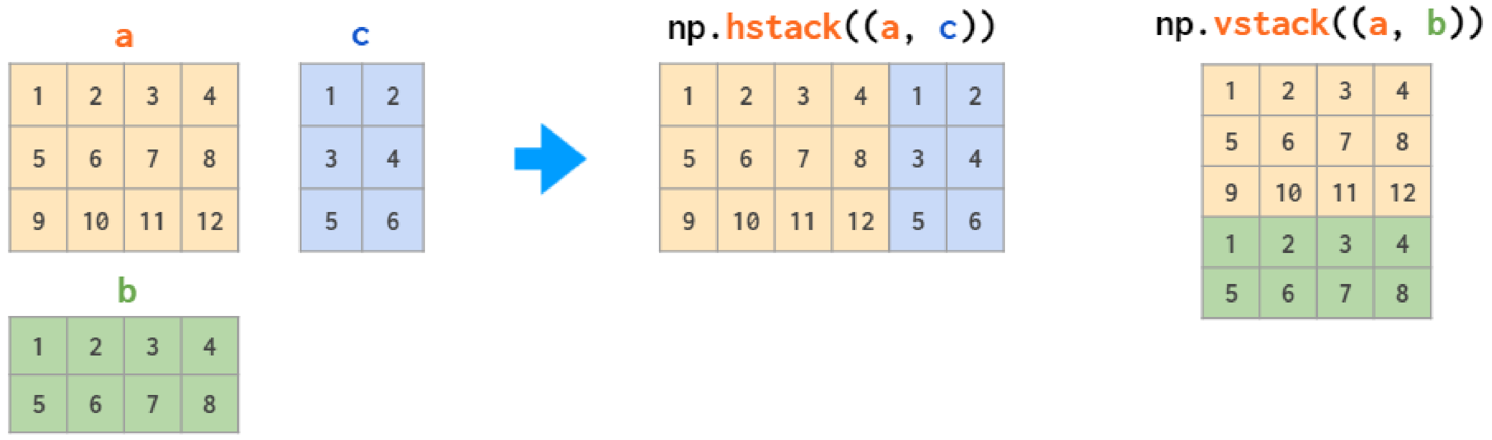




source: www.w3schools.com

Merging and Splitting Arrays

Method	Description
np.concatenate((arr1,arr2),axis=0)	Adds arr2 as rows to the end of arr1
np.concatenate((arr1,arr2),axis=1)	Adds arr2 as columns to end of arr1
np.vstack((arr1,arr2))	Stacks arrays in sequence vertically
np.hstack((arr1,arr2))	Stacks arrays in sequence horizontally
np.split(arr,3)	Splits arr into 3 sub-arrays
np.hsplit(arr,5)	Splits arr horizontally on the 5th index



File I/O

Method	Description
np.loadtxt()	Reads from a text file
np.genfromtxt()	Reads from a CSV file
np.savetxt()	Writes to a text or CSV file