

Python: NumPy Course Notes

Klein
carlj.klein@gmail.com

1 Learning Objectives

Concepts

- Creating and Saving NumPy ndarrays
- Using Built-in Functions to Create ndarrays
- Accessing, Deleting, and Inserting Elements Into ndarrays
- Slicing ndarrays
- Boolean Indexing, Set Operations, and Sorting
- Exercise: Manipulating ndarrays
- Arithmetic Operations and Broadcasting
- Exercise: Creating ndarrays with Broadcasting

Commands

- `np.array`
- `.shape`
- `.size`
- `np.save` and `np.load`
- `np.zeros`
- `np.ones`
- `np.eye`
- `np.diag`
- `np.arange`
- `np.linspace`
- `np.reshape`

- np.random
 - np.random.random
 - np.random.randint
 - np.random.normal
 - np.random.permutation
- np.delete
- np.append
- np.insert
- np.vstack
- np.hstack
- np.unique
- np.intersect1d
- np.setdiff1d
- np.union1d
- np.sort
- Mathematical Functions
 - np.add
 - np.subtract
 - np.multiply
 - np.divide
 - np.sqrt
 - np.exp
 - np.power
- Statistical Functions
 - mean
 - std
 - median
 - max
 - min

NumPy is useful in dealing with arrays of data, which can be thought of in a mathematical sense as vectors and matrices. NumPy is a powerful tool in itself, however an even more useful Python data tool, the Pandas library, is built on top of NumPy. A basic understanding of NumPy is necessary, and we'll cover some of the essential functions and uses in these notes.

2 Creating and Saving NumPy ndarrays

Not needing too much of an explanation, here's a beginning look at creating basic NumPy arrays.

Covered in this section:

- `np.array`: turn a list, or list of lists, into a NumPy array object
- `shape`: return the dimensions of an array object
- `size`: return the total number of elements in an array object
- `save` & `load`: arrays can be saved for future use

```
x = np.array([1, 2, 3, 4, 5])
print(x)
# [1 2 3 4 5]

x.shape
# (5, )

Y = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]])
print(Y)
"""
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
"""

Y.shape
# (4, 3)
Y.size
# 12

# Notes Section:
# dtype changes with mixed data, and will uptype (int + floats ->
#                                     all floats)
# numpy arrays can be saved for later use

# save
np.save('my_array', x)
# load
y = np.load('my_array.npy')
```

- 3 Using Built-in Functions to Create ndarrays
- 4 Accessing, Deleting, and Inserting Elements Into ndarrays
- 5 Slicing ndarrays
- 6 Boolean Indexing, Set Operations, and Sorting
- 7 Exercise: Manipulating ndarrays
- 8 Arithmetic Operations and Broadcasting
- 9 Exercise: Creating ndarrays with Broadcasting