Numpy Tutorials

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python

What is an array

An array is a data structure that stores values of same data type. In Python, this is the main difference between arrays and lists. While python lists can contain values corresponding to different data types, arrays in python can only contain values corresponding to same data type

In [1]:
```
## initially Lets import numpy

import numpy as np
```

In [4]:
```
my_lst=[1,2,3,4,5] # creating a list

arr=np.array(my_lst) # converting the list into array
```

In [5]:
```
print(arr)
```

```
[1 2 3 4 5]
```

In [6]:
```
type(arr)
```

Out[6]:
```
numpy.ndarray
```

In [22]:
```
## Multinested array
my_lst1=[1,2,3,4,5]
my_lst2=[2,3,4,5,6]
my_lst3=[9,7,6,8,9]

arr1=np.array([my_lst1,my_lst2,my_lst3]) # converting multiple list into single a
```

In [23]:
```
arr1
```

Out[23]:
```
array([[1, 2, 3, 4, 5],
       [2, 3, 4, 5, 6],
       [9, 7, 6, 8, 9]])
```

In [10]:
```
type(arr)
```

Out[10]:
```
numpy.ndarray
```

In [12]:
```
## check the shape of the array

arr.shape # the .shape is a built in command which is used to check the shape of
```

Out[12]:
```
(3, 5)
```

Indexing

In [17]:
```
## Accessing the array elements

lst = [1,23,4,6,37,47,38,4,84,7]
# Creating a list and the converting the list into array and the accessing the el
arr = np.asarray(lst)
```

In [19]:
```
arr[3]
type(arr)
```

Out[19]:
```
numpy.ndarray
```

In [24]:
```
arr1
```

Out[24]:
```
array([[1, 2, 3, 4, 5],
       [2, 3, 4, 5, 6],
       [9, 7, 6, 8, 9]])
```

In [25]:
```
arr1[1:,:2]
```

Out[25]:
```
array([[2, 3],
       [9, 7]])
```

In [26]:
```
arr1[:,3:]
```

Out[26]:
```
array([[4, 5],
       [5, 6],
       [8, 9]])
```

In [27]:
```
arr
```

Out[27]:
```
array([ 1, 23,  4,  6, 37, 47, 38,  4, 84,  7])
```

In [30]:
```
arr[3:]=101 # It help us to insert the element at the specific location.
```

In [31]:
```
arr
```

Out[31]:
```
array([  1,  23,   4, 101, 101, 101, 101, 101, 101, 101])
```

In [33]:
```
### Some conditions very useful in Exploratory Data Analysis

val=2

arr[arr<3]
```

Out[33]:
```
array([1])
```

In [32]:
```
## Create arrays and reshape
# np.arange function is the function that help to arrange the element.
np.arange(0,10).reshape(5,2)
```

Out[32]:
```
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7],
       [8, 9]])
```

In [34]:
```
np.arange(0,50)# it will create a array
```

Out[34]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

In [42]:
```
arr2 = np.arange(0,50).reshape(10,5)
print(arr2)
arr2.shape
```

```
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [15 16 17 18 19]
 [20 21 22 23 24]
 [25 26 27 28 29]
 [30 31 32 33 34]
 [35 36 37 38 39]
 [40 41 42 43 44]
 [45 46 47 48 49]]
```

Out[42]:
```
(10, 5)
```

In [44]:
```
arr1=np.arange(0,10).reshape(2,5)
```

In [45]:
```
arr2=np.arange(0,10).reshape(2,5)
```

In [46]:
```
arr1*arr2 # multiplying the two column
```

Out[46]:
```
array([[ 0,  1,  4,  9, 16],
       [25, 36, 49, 64, 81]])
```

In [47]:
```
np.ones((2,5),dtype=int) # create rows and column of 1
```

Out[47]:
```
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]])
```

In [49]:
```
## random distribution
np.random.rand(3,3) # will generate random numbers
```

Out[49]:
```
array([[0.21078531, 0.60791362, 0.63092898],
       [0.22883549, 0.21185864, 0.97415728],
       [0.93729829, 0.4822592 , 0.70300795]])
```

In [50]:
```
arr_ex=np.random.randn(4,4)
```

In [51]:
```
arr_ex
```

Out[51]:
```
array([[-2.67105595, -0.12238506, -1.87595607,  0.01838899],
       [ 1.17369358, -0.53912755,  1.242936  ,  0.39939667],
       [ 0.84980885,  1.14986164, -0.11058519,  0.20613945],
       [ 0.62379436, -0.45980941, -1.43902165, -0.34250343]])
```

In [52]:
```
import seaborn as sns
```

In [54]:
```
import pandas as pd
df = pd.DataFrame(arr_ex.reshape(16,1))
df.head()
```

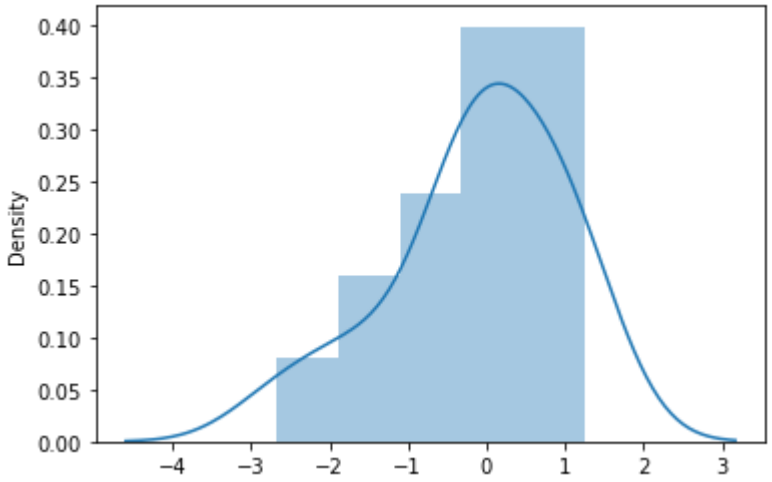
Out[54]:

	0
0	-2.671056
1	-0.122385
2	-1.875956
3	0.018389
4	1.173694

In [55]:
```
sns.distplot(df)
```

```
C:\Users\mksmu\anaconda3\lib\site-packages\seaborn\distributions.py:2557: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future v
ersion. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```

Out[55]: <AxesSubplot:ylabel='Density'>



In [56]:
```
np.random.randint(0,100,8).reshape(4,2) # generate data in integer format
```

Out[56]:
```
array([[29, 64],
       [82, 85],
       [95, 34],
       [76, 58]])
```

In [75]:
```
np.random.random_sample((1,5)) # randon sample
```

Out[75]:
```
array([[0.07005997, 0.98540348, 0.98450756, 0.65948775, 0.34944308]])
```

In [ ]:
```
'''np.random.randint()
np.random.randn()
np.random.random_sample()'''

all the above help in generating the data
```