מבוא ל-PYTHON – מה אנחנו כבר יודעים

- הכרות עם שפת התכנות פייתון (PYTHON)
- הכרות עם סביבת העבודה של PYTHON
- הכרות עם התכנות הבסיסי בפייתון

- **ספריות – הכנה ללמידת מכונה:**

mathplotlib✓

pandas✓

Numpy✓

seaborn✓

מחברת מלווה לשקפי המצגת

הכל נמצא ברשת, צריך רק לדעת איפה לחפש:

אתרים ללמוד עצמי - numpy

- NumPy Cheat Sheet: Data Analysis in Python
- NumPy Tutorial: Your First Steps Into Data Science in Python
- W3School
- Numpy videos

- ✓ תמיכה נוחה במערכים מרובי ממדים (לא רשימה של רשימות כי אם מערך -nממדי אמיתי)
- ✓ מימוש יעיל יותר, הן של המערכים והן של פונקציות עזר.
- ✓ מערכים משמשים לעתים קרובות מאוד במדעי הנתונים, כאשר מהירות ומשאבים חשובים מאוד.

ספריית numpy

Step 1: Install NumPy

Use the command:

'pip install numpy'.

Step 2: Import NumPy

Use the keyword

'import numpy as np'

at the beginning of your Python file.

Step 3: Check the Version

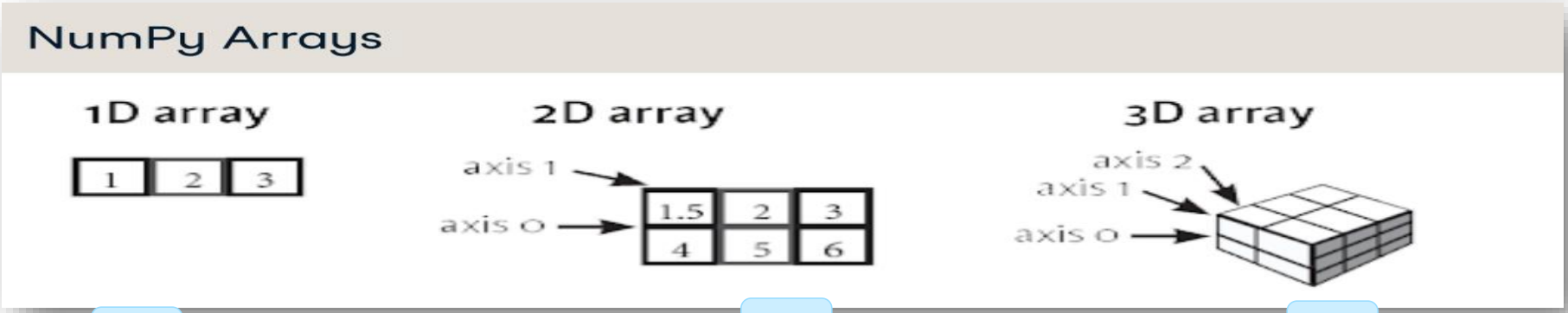
Use

'print(np.__version__)'

Numpy – מה נלמד?

- ✓ יצירת מערך (מימד 1, דו מימדי, תלת מימדי)
- ✓ גישה לאיברי המערך (מימד 1, דו מימדי, תלת מימדי)
- ✓ Slicing
- ✓ Shape ו reshape
- ✓ מעבר על איברי המערך בלולאה
- ✓ העתקת מערך copy
- ✓ איחוד מערכים concatenate, hstack, vstack
- ✓ חיפוש, מיון, וסינון ערכים במערך
- ✓ יצירת מערך עם ערכים רנדומלים
- ✓ פונקציות מתמטיות על מערכים: add, subtract, multiply, divide, power
- ✓ פונקציות אגריגציה – sum, min, max, median , (לפי שורה, לפי עמודה)

יצירת מערך – מתוך רשימה



1

#מימד גנרי
```
lst1 = [1,2,3]
arr1 = np.array(lst1)
```

#מימד אחד-שלוש שורות,עמודה אחת
```
lst1 = [[1],[2],[3]]
arr1 = np.array(lst1)
```

#מימד אחד-שורה אחת שלוש עמודות
```
lst1 = [[1,2,3]]
arr1 = np.array(lst1)
```

2

#משני מימדים
```
lst1 =
[[1.5,2,3],[4,5,6]]
arr1 = np.array(lst1)
```

3

#שלוש מימדים
```
lst1 = [[1,2,3],[4,5,6]]
lst2 = [[1.5,2.5,3.5]
,[4.5,5.5,6.5]]
arr1 =
np.array([lst1,lst2])
```

יצירת מערך – טווח

np.arange(start, stop, step)

צעדים, 1 כדיפולט — סיום, לא כולל — התחלה, 0 כדיפולט

```
#הגדרה באמצעות טווח
arr1=np.arange(4)
print(arr1)
arr1=np.arange(2,9)
print("\n",arr1)
arr1=np.arange(2,9,3)
print("\n",arr1)
```

```
[0 1 2 3]

 [2 3 4 5 6 7 8]

 [2 5 8]
```

מימדים של מערך

1. **Dimension of an Array**
   'arr.ndim' to get the number of dimensions of an array.

2. **Shape of an Array**
   'arr.shape' to get the number of rows and columns in a 2D arra

3. **Size of an Array**
   'arr.size' to get the total number of elements in an array.

4. **Change shape of an Array**
   np.arange(9).reshape((3,3))' to change from(9,) to (3,3)

גישה לאיברים במערך

1 **גישה לתא בודד באמצעות []**

'arr[row_num][col_num]'

2 **גישה למספר איברים**

'arr[row_num][:]'
שורה מסוימת, כל העמודות

'arr[:][col_num]'
כל השורות, עמודה מסוימת

3 **שינוי ערך מסוים ע"י עידכן הערך באינדקס מסוים**

arr1[0,1]=999
עדכון הערך בשורה הראשונה ובעמודה השניה

יצירת מערכים באמצעות פונקציות

1 **`ones()`, `zeros()`, `empty()`**

Create arrays filled with ones, zeros, or random values, respectively.

2 **`eye()`, `diag()`**

Create 2D arrays with ones on the diagonal, or with a given number on the diagonal, respectively.

3 **`linspace()`, `logspace()`**

Create arrays with values evenly spaced between two endpoints and logarithmically spaced, respectively.

יצירת מספר / מערך – Random

```
#Generate a random integer from 0 to 100:
x = np.random.randint(100)
print("\n print random integer:\n",x)
#Generate a random float from 0 to 1:
y = np.random.rand()
print("\n print random float:\n",y)
#Generate a 1-D array containing 5 random integers from 0 to 100:
z1 = np.random.randint(100, size=(5))
print("\n print random integer array:\n",z1, "shape", z1.shape)
z2 = np.random.randint(100, size=(1,5))
print("\n print random integer array:\n",z2, "shape", z2.shape)
#random.uniform(low=0.0, high=1.0, size)
u1 = np.random.uniform(size=[3,5])  #random uniformly from 0 to 1
print("\n print u1:\n",u1)
u2 = np.random.uniform(-1,1,size=[2,4])
print("\n print u2:\n",u2)
#random.normal(low=0.0, high=1.0, size)
n1 = np.random.uniform(size=[3,5])  #random normal from 0 to 1
print("\n print n1:\n",n1)
n2 = np.random.uniform(-1,1,size=[2,4])
print("\n print n2:\n",n2)
```

יצירת מערך עם איברים זהים

'np.full((dim), value)'

חד מימד

1

```
my_dim=(1,6)
print(np.full(my_dim,7))
```

```
[[7 7 7 7 7 7]]
```

דו מימדי

2

```
my_dim=(2,2)
print(np.full(my_dim,7)
)
```

```
[[7 7]
 [7 7]]
```

תלת מימדי

3

```
my_dim=(2,3,4)
print(np.full(my_dim,7))
```

```
[[[7 7 7 7]
 [7 7 7 7]
 [7 7 7 7]]
 [[7 7 7 7]
 [7 7 7 7]
 [7 7 7 7]]]
```

חיתוך מערכים

```
import numpy as np
x = np.array([0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print("\nprint all:\n",x)
print("start=1:stop=7:step=no ",x[1:7])
print("start=5:stop=no:step=no ",x[5:])
print("start=no:stop=5:step=no ",x[:5])
print("start=1:stop=7:step=2 ",x[1:7:2])
print("start=no:stop=no:step=-1 ",x[::-1])
print("start=-3:stop=-6:step=-1 ",x[-3:-6:-1])
```

```
print all:
 [ 0  1  2  3  4  5  6  7  8  9 10]
start=1:stop=7:step=no  [1 2 3 4 5 6]
start=5:stop=no:step=no  [ 5  6  7  8  9 10]
start=no:stop=5:step=no  [0 1 2 3 4]
start=1:stop=7:step=2  [1 3 5]
start=no:stop=no:step=-1  [10  9  8  7  6  5  4  3  2  1  0]
start=-3:stop=-6:step=-1  [8 7 6]
```

חיתוך מערכים דו מימדים

```
x2 = np.array([ [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] ])
print("\nprint all:\n",x2)
print("FROM:start=1:stop=4:step=no TO:start=1:stop=4:step=no\n",x2[1:4,1:4])
print("FROM:start=2:stop=no:step=no TO:start=2:stop=no:step=no\n",x2[2:,2:])
print("FROM:start=no:stop=no:step=2 TO:start=no:stop=no:step=2\n",x2[::2,::2])
print("FROM:start=no:stop=no:step=-1 TO:start=no:stop=no:step=-1\n",x2[::-1,::-1])
print("FROM:start=-2:stop=no:step=-1 TO:start=-2:stop=no:step=-1\n",x2[-2::-1,-2::-1])
```

פיצול מערכים

```
x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
x_split = np.split(x,2)
print("\nprint all:\n",x)
print("\nprint all split:\n",x_split)
print("\nprint array[0]:\n",x_split[0])
print("\nprint array[1]:\n",x_split[1])
print(type(x_split))
```

```
print all:
 [ 1  2  3  4  5  6  7  8  9 10]

print all split:
 [array([1, 2, 3, 4, 5]), array([ 6,  7,  8,  9, 10])]

print array[0]:
 [1 2 3 4 5]

print array[1]:
 [ 6  7  8  9 10]
<class 'list'>
```

שירשור מערכים

```
x1 = np.array([1, 2, 3, 4, 5 ])
x2 = np.array([6, 7, 8, 9, 10])
x = np.vstack((x1,x2))
print("\nprint x:\n",x)
y = np.hstack((x1,x2))
print("\nprint y:\n",y)
```

```
print x:
 [[ 1  2  3  4  5]
 [ 6  7  8  9 10]]

print y:
 [ 1  2  3  4  5  6  7  8  9 10]
```

פעולות חשבון על מערכים

```
x1 = np.array([1, 2, 3, 4, 5 ])
x2 = x1 * 10
print("\nprint x1:\n",x1)
print("\nprint x2:\n",x2)

x1= x1 +10
print("\nprint x1:\n",x1)
```

```
print x1:
 [1 2 3 4 5]

print x2:
 [10 20 30 40 50]

print x1:
 [11 12 13 14 15]
```

פעולות לוגיות על מערכים

```
x1 = np.array([1, 2, 3, 4, 5 ])
x2 = np.array([1, 3, 3, 6, 7 ])
x3 = x1 == x2
x4 = x1 < x2
x5 = x1 > x2
x6 = x1 != x2
print("\n x1 == x2: \n",x3)
print("\n x1 > x2: \n",x4)
print("\n x1 < x2: \n",x5)
print("\n x1 != x2: \n",x6)
```

```
x1 == x2:
[ True False  True False False]

x1 > x2:
[False  True False  True  True]

x1 < x2:
[False False False False False]

x1 != x2:
[False  True False  True  True]
```

סכימה על מערכים

```
y=[1,2,3,4,5,6]
y=np.reshape(y,(3, 2))
print("\n", y)
#סכום שורות
print("\n", y.sum(axis=0))
#סכום עמודות
print("\n", y.sum(axis=1))
print("\n", y)
```

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

 [ 9 12]

 [ 3  7 11]

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