



آشنایی با پایتون و کتابخانه های Pandas Matplotlib Numpy

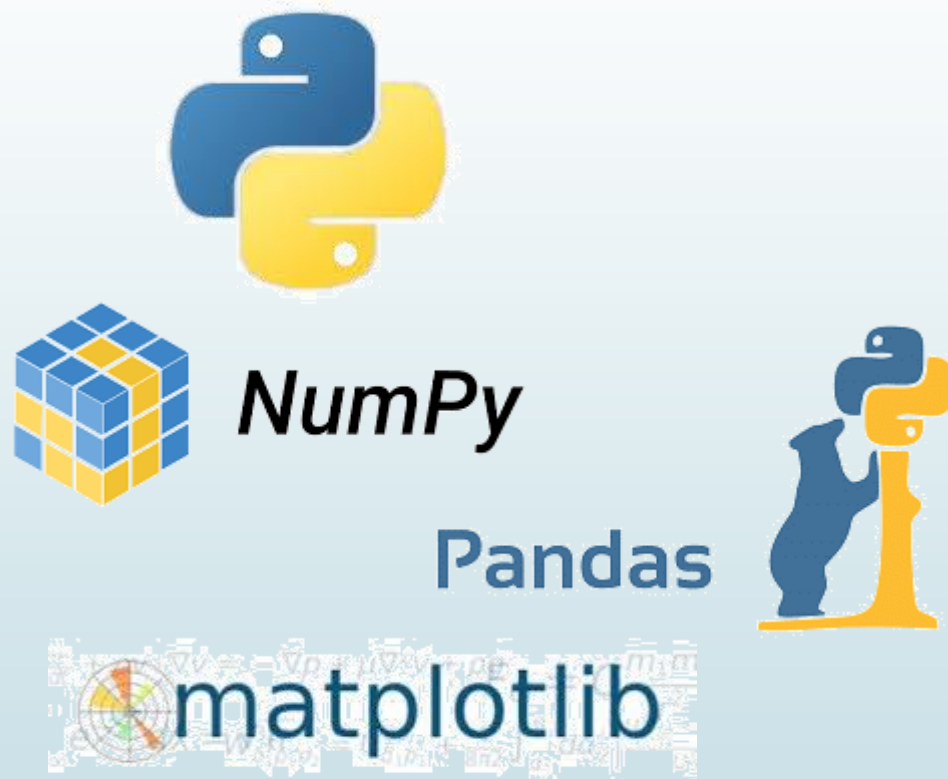
نادیه آرمین

گروه مهندسی کامپیوتر

دانشگاه فردوسی

پاییز ۹۹

سر فصل



- Python ■
- Numpy ■
- Pandas ■
- Matplotlib ■

چرا پایتون

از سال ۲۰۱۶ ، پایتون به عنوان محبوب ترین زبان در دانشگاه ها جایگزین زبان جاوا شد

- منبع باز
- یادگیری سریع و آسان
- کتابخانه های قدرتمند برای علم داده و یادگیری ماشین
- توسعه وب (Django و Flask)



Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

...



پایتون

انتساب

جریان کنترلی

ساختمان‌های داده

توابع



پایتون

انتساب ➡

```
# ---Strings  
data = 'Hello world'  
print(data[0])  
print(len(data))  
print(data)
```

```
# ---Number  
val = 123.4  
print(val)  
val = 10  
print(val)
```

```
# ---Boolean  
a = True  
b = False  
print(a, b)
```

یادگیری ماشین با پایتون



```
# ---NO value
```

```
a = None
```

```
print(a)
```

```
# ---Mutiple Assignment
```

```
a, b, c = 1, 2, 3
```

```
print(a, b, c)
```



پایتون

جریان کنترلی ➡

```
# ---If_THEN_ELSE  
val = 99  
if val == 99 :  
    print('fast')  
elif val > 200:  
    print('too fast')  
else:  
    print('safe')
```




پایتون

جریان کنترلی ➡

```
# ---FOR_Loop  
for i in range(10):  
    print(i)
```

```
# ---while_Loop  
i = 0  
while i < 10:  
    print(i)  
    i += 1
```



ساختمان های داده

---Tuples

```
myTuple = (1, 2, 3)
```

```
print('Tuple:', myTuple)
```

---Lists

```
myList = [1, 2, 3]
```

```
print('Zeroth value: %d' % myList[0])
```

```
myList.append(4)
```

```
print('List length : %d' % len(myList))
```

```
for val in myList:
```

```
    print ( val)
```

ساختمان های داده



```
# ---Dictionary
mydict = {'a': 1, 'b': 2, 'c': 3, 'e': 4}
print("a : %d" % mydict['a'])
mydict['a'] = 11
print("a : %d" % mydict['a'])
print("Keys: { }".format(mydict.keys()))
print("Value: { }".format(mydict.values()))
for k in mydict.keys():
    print(mydict[k])
```



```
# ---Function  
def myfunc(a, b):  
    return (a + b)
```

```
# x = myfunc(b=8, a=3)  
x = myfunc(4, 5)  
print(x)
```

Numpy ■

```
import numpy  
print ('Numpy: { }'.format(numpy.__version__))
```

Matplotlib ■

```
import matplotlib  
print ('Matplotlib: { }'.format(matplotlib.__version__))
```

Pandas ■

```
import pandas  
print ('Pandas: { }'.format(pandas.__version__))
```



NumPy

NumPy

ایجاد آرایه ➡

```
# define an array
```

```
import numpy
```

```
myList = [1, 2, 3]
```

```
myArray = numpy.array(myList)
```

```
print(myArray)
```

```
print(myArray.shape)
```



NumPy

دسترسی به داده ها ➡

```
# access values
myList = [[1, 2, 3], [4, 5, 6]]
myArray = numpy.array(myList)
print(myArray)
print(myArray.shape)
print(myArray[0])
print(myArray[-1])
print(myArray[0, 2])
print("ALL coll { }".format(myArray[:, 2]))
```



NumPy

محاسبات ➡

```
# Array arithmetic
```

```
myArray1 = numpy.array([1, 2, 3])
```

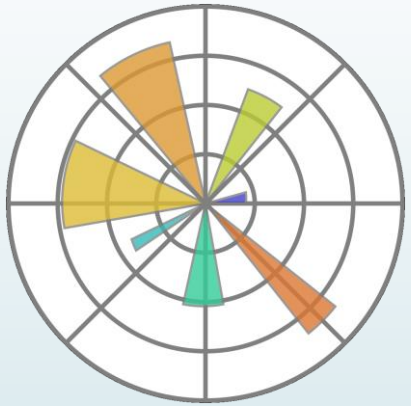
```
myArray2 = numpy.array([4, 5, 6])
```

```
print(myArray1 + myArray2)
```

```
print(myArray1 * myArray2)
```


Matplotlib

رسم خط ➡

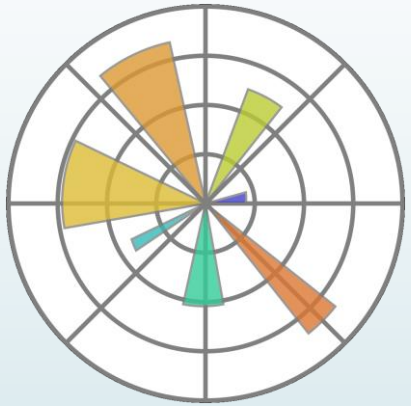


matplotlib

```
# basic line plot
import matplotlib.pyplot as plt
import numpy

myArray = numpy.array([1, 2, 3])
plt.plot(myArray)
plt.xlabel(" x axis")
plt.ylabel(" y axis")
plt.show()
```

Matplotlib



matplotlib

نمودار پراکنشی

```
# basic scatter plot  
import matplotlib.pyplot as plt  
import numpy
```

```
x = numpy.array([1, 2, 3])  
y = numpy.array([3, 5, 7])  
plt.scatter(x, y)  
plt.show()  
plt.plot(x, y)  
plt.show()
```



Pandas

سری‌ها ➡

```
# series
```

```
import numpy
```

```
import pandas
```

```
myArray = numpy.array([1, 2, 3])
```

```
rowname = ['a', 'b', 'c']
```

```
myseries = pandas.Series(myArray, index=rowname)
```

```
print(myseries)
```

```
print(myseries[0])
```

```
print(myseries['a'])
```



Pandas

فریم داده ➡

```
# dataframe
import numpy
import pandas

myArray = numpy.array([[1, 2, 3], [4, 5, 6]])
rowname = ['a', 'b']
colname = ['one', 'two', 'three']
mydataframe = pandas.DataFrame(myArray, index=rowname, columns=colname)
print(mydataframe)
print("method1: ")
print("one column:\n { }".format(mydataframe['one']))
print("method2: ")
print("one column:\n { }".format(mydataframe.one))
```

