

مرحباً طلابي

أردت توضيح بعض النقاط:

الدالة `np.random.seed()`

وظيفة هذه الدالة هي منع تغيير القيم العشوائية المنشئة ويتم إعطاؤها رقم أو مصفوفة أحادية الأبعاد

نستخدمها عندما نستخدم الدوال مثل `rand` , `randn`

```
In [1]: from numpy.random import seed
        seed(1)
```

للمزيد عنها

<https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.seed.html>
(<https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.seed.html>)

توضيح لأخذ قطعة من الداتا فريم

```
In [2]: import pandas as pd
        from numpy.random import randn, seed
        seed(0)
        df = pd.DataFrame(randn(5,4), "A B C D E".split(), "W X Y Z".split())
        df
```

Out[2]:

	W	X	Y	Z
A	1.764052	0.400157	0.978738	2.240893
B	1.867558	-0.977278	0.950088	-0.151357
C	-0.103219	0.410599	0.144044	1.454274
D	0.761038	0.121675	0.443863	0.333674
E	1.494079	-0.205158	0.313068	-0.854096

```
In [3]: df.loc[["B", "C"], ["X", "Y"]]
```

Out[3]:

	X	Y
B	-0.977278	0.950088
C	0.410599	0.144044

```
In [62]: from numpy.random import randn, seed
seed(0)
df = pd.DataFrame(randn(5,4), "A B C D E".split(), "W X Y Z".split())
df
```

Out[62]:

	W	X	Y	Z
A	1.764052	0.400157	0.978738	2.240893
B	1.867558	-0.977278	0.950088	-0.151357
C	-0.103219	0.410599	0.144044	1.454274
D	0.761038	0.121675	0.443863	0.333674
E	1.494079	-0.205158	0.313068	-0.854096

```
In [64]: df.loc["C","X"]
```

Out[64]: 0.41059850193837233

```
In [62]: from numpy.random import randn, seed
seed(0)
df = pd.DataFrame(randn(5,4), "A B C D E".split(), "W X Y Z".split())
df
```

Out[62]:

	W	X	Y	Z
A	1.764052	0.400157	0.978738	2.240893
B	1.867558	-0.977278	0.950088	-0.151357
C	-0.103219	0.410599	0.144044	1.454274
D	0.761038	0.121675	0.443863	0.333674
E	1.494079	-0.205158	0.313068	-0.854096

```
In [65]: df.loc[["B","C"],["X","Y"]]
```

Out[65]:

	X	Y
B	-0.977278	0.950088
C	0.410599	0.144044

مرحباً طلابي

أردت توضيح بعض النقاط:

```
In [1]: import pandas as pd
from numpy.random import randn, seed
seed(0)
df = pd.DataFrame(randn(5,4), "A B C D E".split(), "W X Y Z".split())
df
```

Out[1]:

	W	X	Y	Z
A	1.764052	0.400157	0.978738	2.240893
B	1.867558	-0.977278	0.950088	-0.151357
C	-0.103219	0.410599	0.144044	1.454274
D	0.761038	0.121675	0.443863	0.333674
E	1.494079	-0.205158	0.313068	-0.854096

يمكننا وضع شرط وأرجاع قيم منطقة مع الصفوف مثل الأعمدة هكذا

```
In [2]: df.loc["C"] < 0
```

```
Out[2]: W      True
X      False
Y      False
Z      False
Name: C, dtype: bool
```

بس أن جينا نكمل ونحدد يرجعلنا أيه هيدينا خطأ عشان ماينفعش يحذف عمود كامل والشرط بندهوله بالأندكسينج فمثلاً

ولكن إذا قمنا بالتالي سيعطينا خطأ لأن لا يمكن حذف عمود كامل

```
In [3]: df[df.loc["C"] < 0]
```

```
C:\Users\Prog-Mohamed\Anaconda3\envs\py27\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
    """Entry point for launching an IPython kernel.
```

```
-----  
IndexingError                                Traceback (most recent call last)  
<ipython-input-3-6dbb041d76d6> in <module>()  
----> 1 df[df.loc["C"] < 0]
```

```
C:\Users\Prog-Mohamed\Anaconda3\envs\py27\lib\site-packages\pandas\core\frame.py  
yc in __getitem__(self, key)  
    2131         if isinstance(key, (Series, np.ndarray, Index, list)):  
    2132             # either boolean or fancy integer index  
-> 2133             return self._getitem_array(key)  
    2134         elif isinstance(key, DataFrame):  
    2135             return self._getitem_frame(key)
```

```
C:\Users\Prog-Mohamed\Anaconda3\envs\py27\lib\site-packages\pandas\core\frame.py  
yc in _getitem_array(self, key)  
    2171         # check_bool_indexer will throw exception if Series key can  
not  
    2172         # be reindexed to match DataFrame rows  
-> 2173         key = check_bool_indexer(self.index, key)  
    2174         indexer = key.nonzero()[0]  
    2175         return self._take(indexer, axis=0, convert=False)
```

```
C:\Users\Prog-Mohamed\Anaconda3\envs\py27\lib\site-packages\pandas\core\indexing  
g.pyc in check_bool_indexer(ax, key)  
    2021         mask = isna(result._values)  
    2022         if mask.any():  
-> 2023             raise IndexingError('Unalignable boolean Series provided as  
,  
    2024                                     'indexer (index of the boolean Series a  
nd of '  
    2025                                     'the indexed object do not match')
```

```
IndexingError: Unalignable boolean Series provided as indexer (index of the boo  
lean Series and of the indexed object do not match
```

يمكننا إضافة صف جديد كالتالي:

```
In [4]: df.loc["New Row"] = [1, 2, 3, 4]
```

In [5]: df

Out[5]:

	W	X	Y	Z
A	1.764052	0.400157	0.978738	2.240893
B	1.867558	-0.977278	0.950088	-0.151357
C	-0.103219	0.410599	0.144044	1.454274
D	0.761038	0.121675	0.443863	0.333674
E	1.494079	-0.205158	0.313068	-0.854096
New Row	1.000000	2.000000	3.000000	4.000000

ويجب أن يكون طول القائمة تساوي عدد أعمدة الداتا فريم

يمكننا تغيير قيمة في الداتا فريم عن طريق تحديد موضعها ثم إعادة تعيين قيمة لها

In [6]: df.loc[["New Row"],["W"]]

Out[6]:

	W
New Row	1.0

In [7]: df.loc[["New Row"],["W"]] = 3

In [8]: df.loc[["New Row"],["W"]]

Out[8]:

	W
New Row	3.0

In [9]: df

Out[9]:

	W	X	Y	Z
A	1.764052	0.400157	0.978738	2.240893
B	1.867558	-0.977278	0.950088	-0.151357
C	-0.103219	0.410599	0.144044	1.454274
D	0.761038	0.121675	0.443863	0.333674
E	1.494079	-0.205158	0.313068	-0.854096
New Row	3.000000	2.000000	3.000000	4.000000

مرحباً طلابي

أردت توضيح بعض النقاط:

تفسير:

```
In [1]: import pandas as pd
d = {
    "A": [1,2,3],
    "B": [4,5,6],
    "C": [7,8,9]
}
```

```
In [2]: pd.DataFrame(d.values(), columns=d.keys())
```

Out[2]:

	A	C	B
0	1	2	3
1	7	8	9
2	4	5	6

```
In [3]: pd.DataFrame(d)
```

Out[3]:

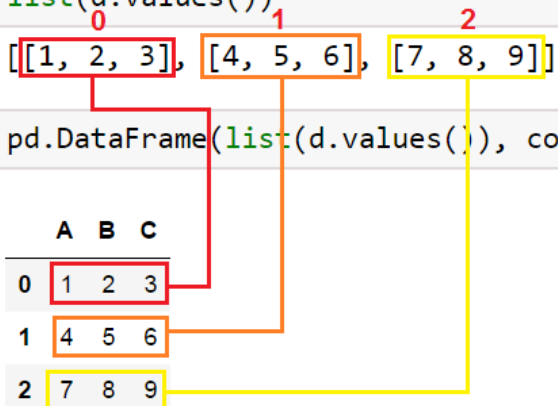
	A	B	C
0	1	4	7
1	2	5	8
2	3	6	9

التفسير الأول:

```
In [4]: d.values()
```

Out[4]: [[1, 2, 3], [7, 8, 9], [4, 5, 6]]

```
In [20]: list(d.values())
```

Out[20]: 

```
In [21]: pd.DataFrame(list(d.values()), columns=list(d.keys()))
```

Out[21]:

	A	B	C
0	1	2	3
1	4	5	6
2	7	8	9

التفسير الثاني:

```
In [5]: zip(*d.values())
```

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

```
In [2]: d = {  
    "A": [1,2,3],  
    "B": [4,5,6],  
    "C": [7,8,9]  
}  
import pandas as pd  
pd.DataFrame(d)
```

Out[2]:

	A	B	C
0	1	4	7
1	2	5	8
2	3	6	9

في الدقيقه 5 الثانية 45

الميثود min, max قامت بأرجاع أعلى أو أقل بأتعين محققين مبيعات من كل شركة وليس حسب الحروف

مرحباً طلابي

أردت توضيح طريقة عمل `pd.Series()` مع القواميس

الطريق هي أن الدالة تقوم بأخذ مفاتيح القاموس ووضعها كـ `label` والقيم كـ `Series`

مثال توضيحي:

```
In [1]: import pandas as pd
d = {"a":10, "b":20, "c":30}
pd.Series(d)
```

```
Out[1]: a    10
       b    20
       c    30
       dtype: int64
```

```
In [2]: pd.Series(list(d.values()), list(d.keys()))
```

```
Out[2]: a    10
       c    30
       b    20
       dtype: int64
```