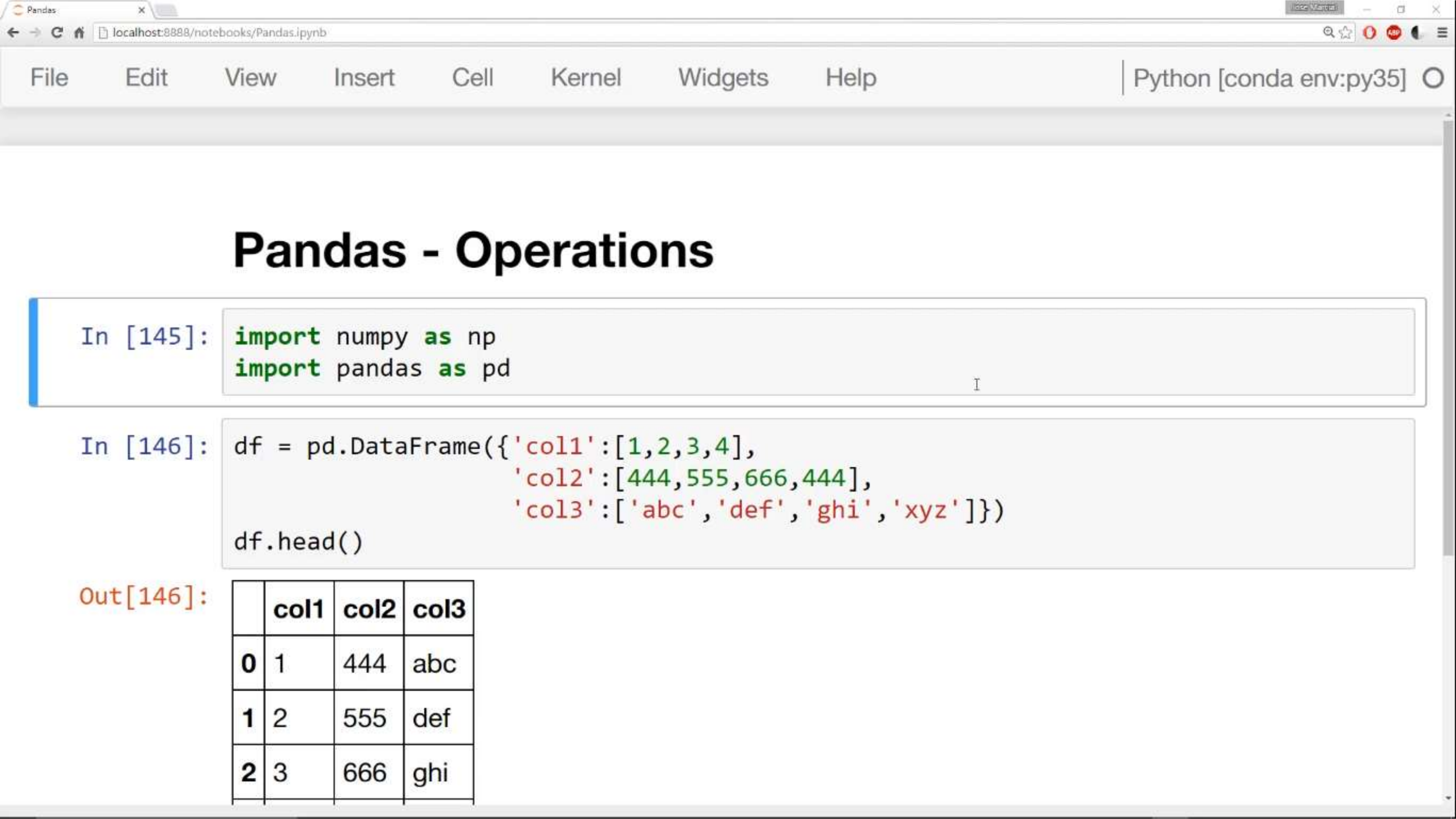
Python for Data Science

Operations

PIERIAN DATA
www.pieriandata.com


Pandas
localhost:8888/notebooks/Pandas.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:py35]

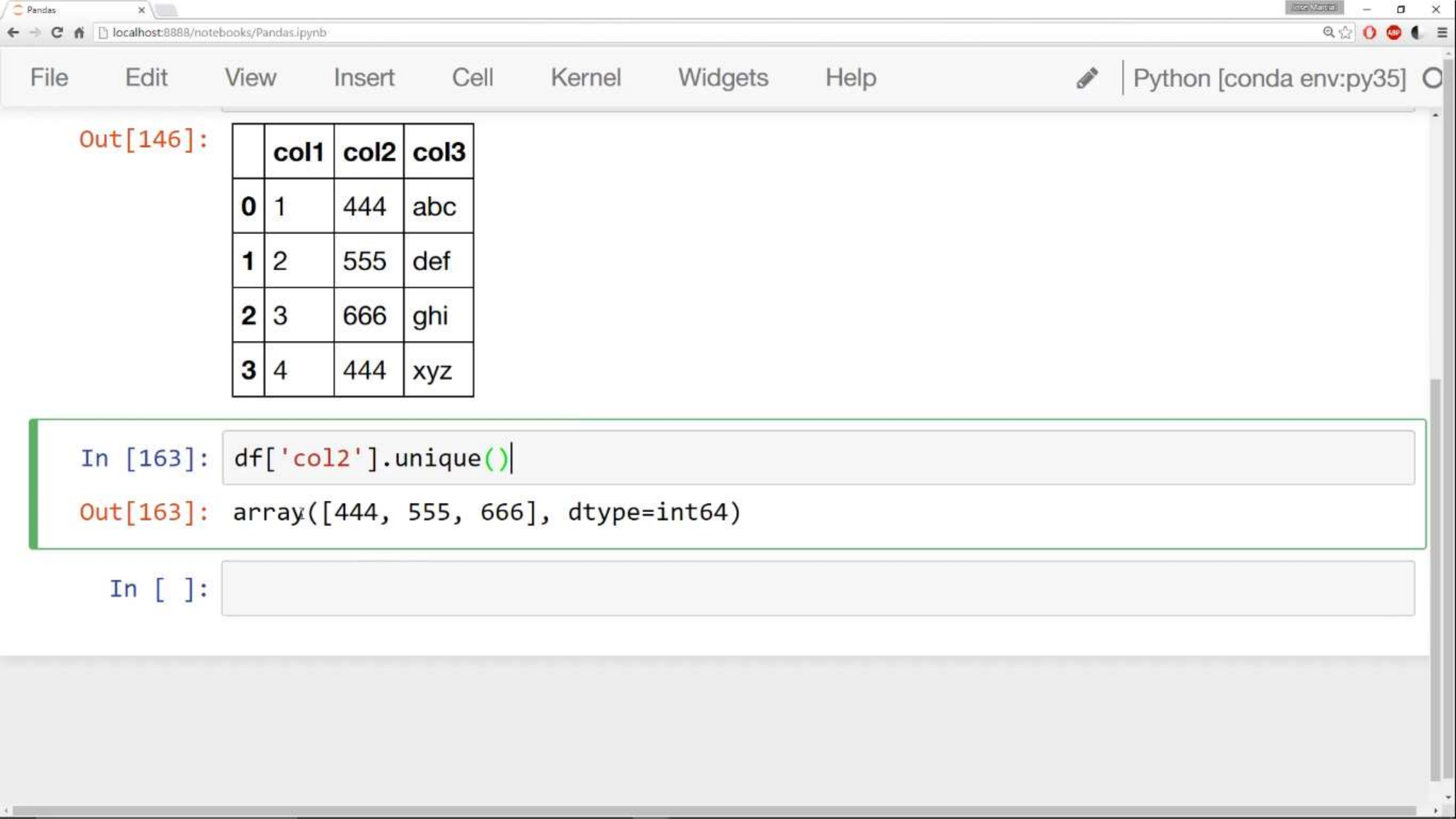
Pandas - Operations

In [145]:
```
import numpy as np
import pandas as pd
```

In [146]:
```
df = pd.DataFrame({'col1':[1,2,3,4],
                   'col2':[444,555,666,444],
                   'col3':['abc','def','ghi','xyz']})
df.head()
```

Out[146]:

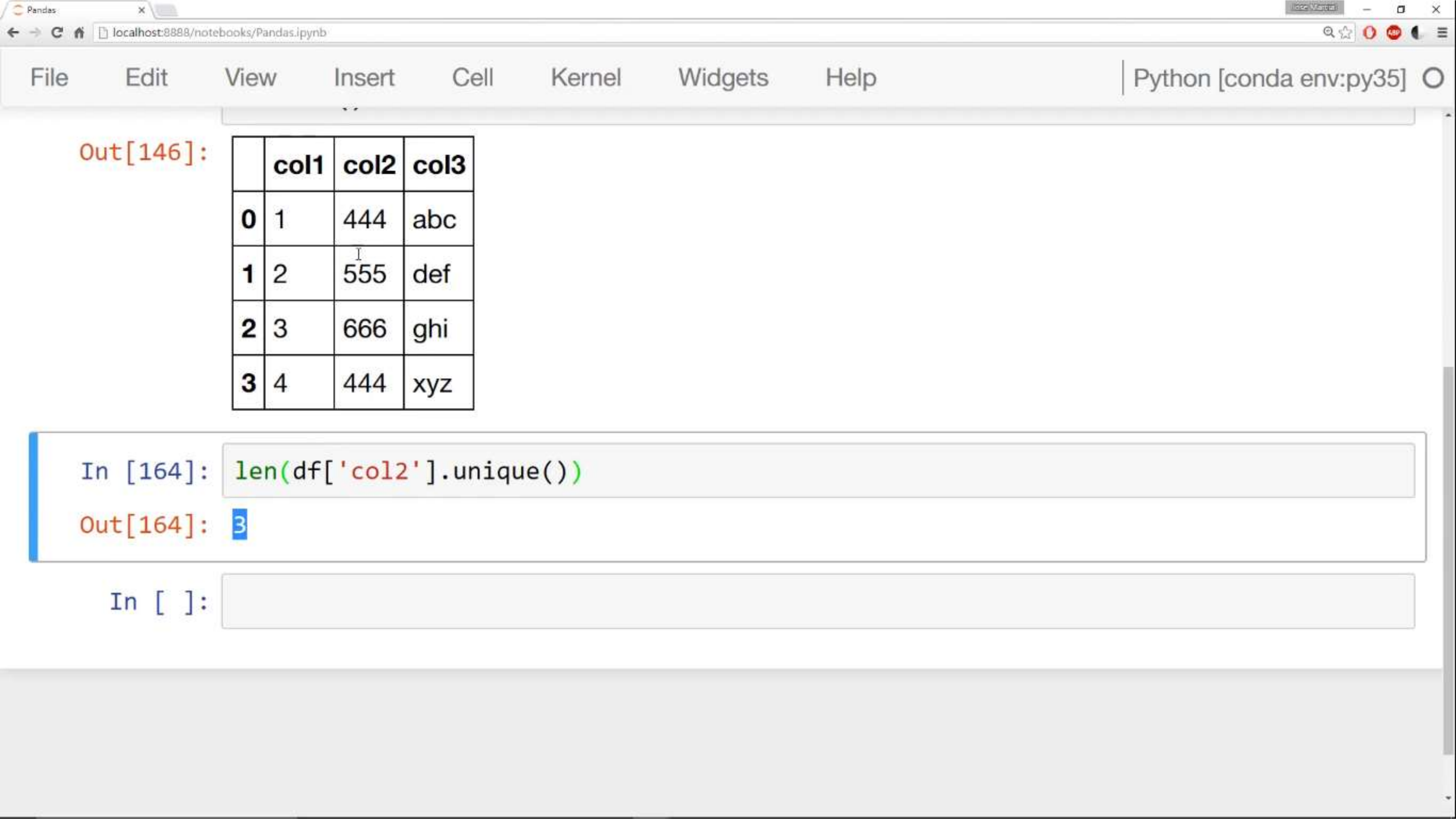
	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi



Out[146]:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

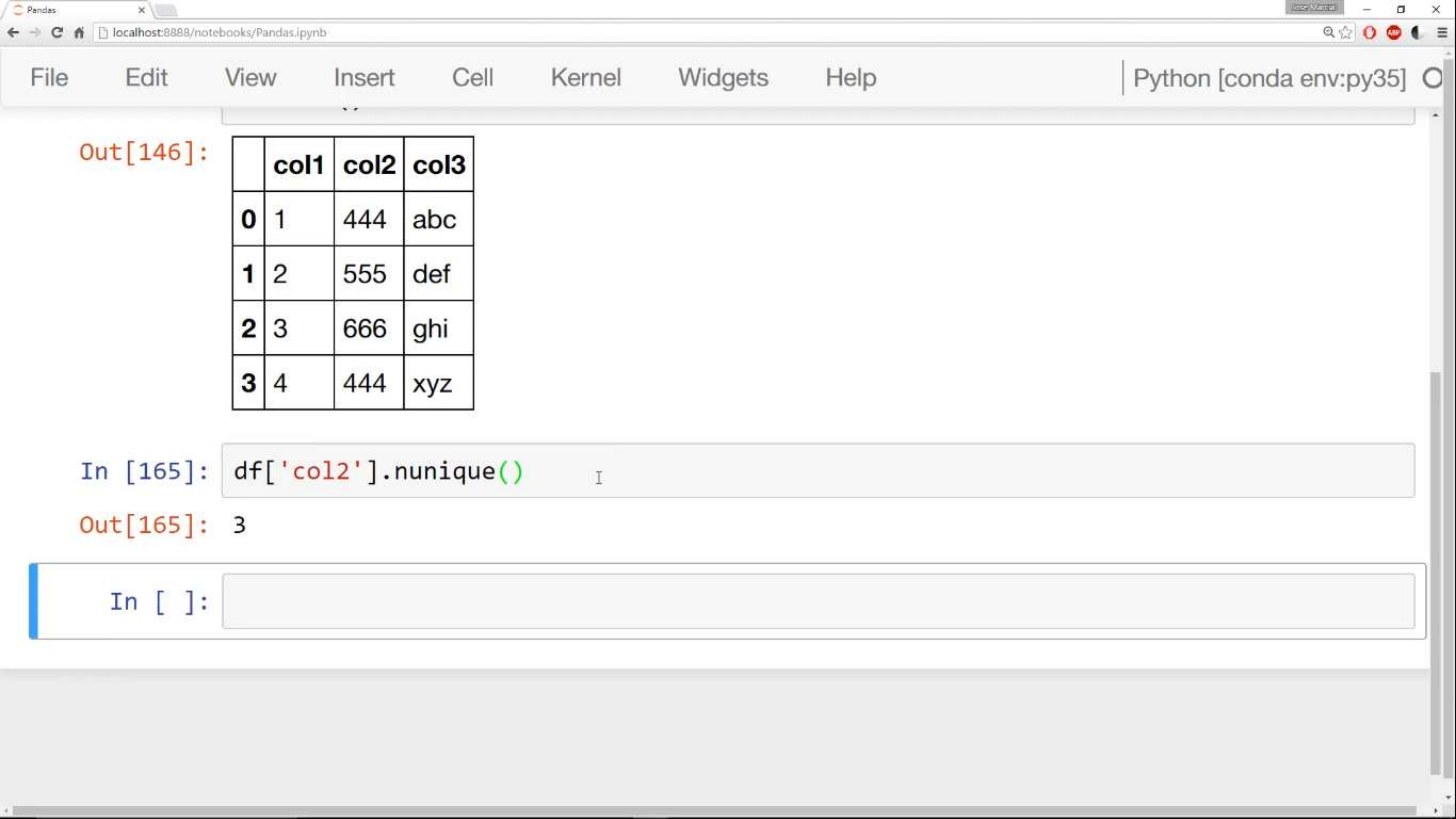
```
In [163]: df['col2'].unique()

Out[163]: array([444, 555, 666], dtype=int64)

In [ ]:
```



Out[146]:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

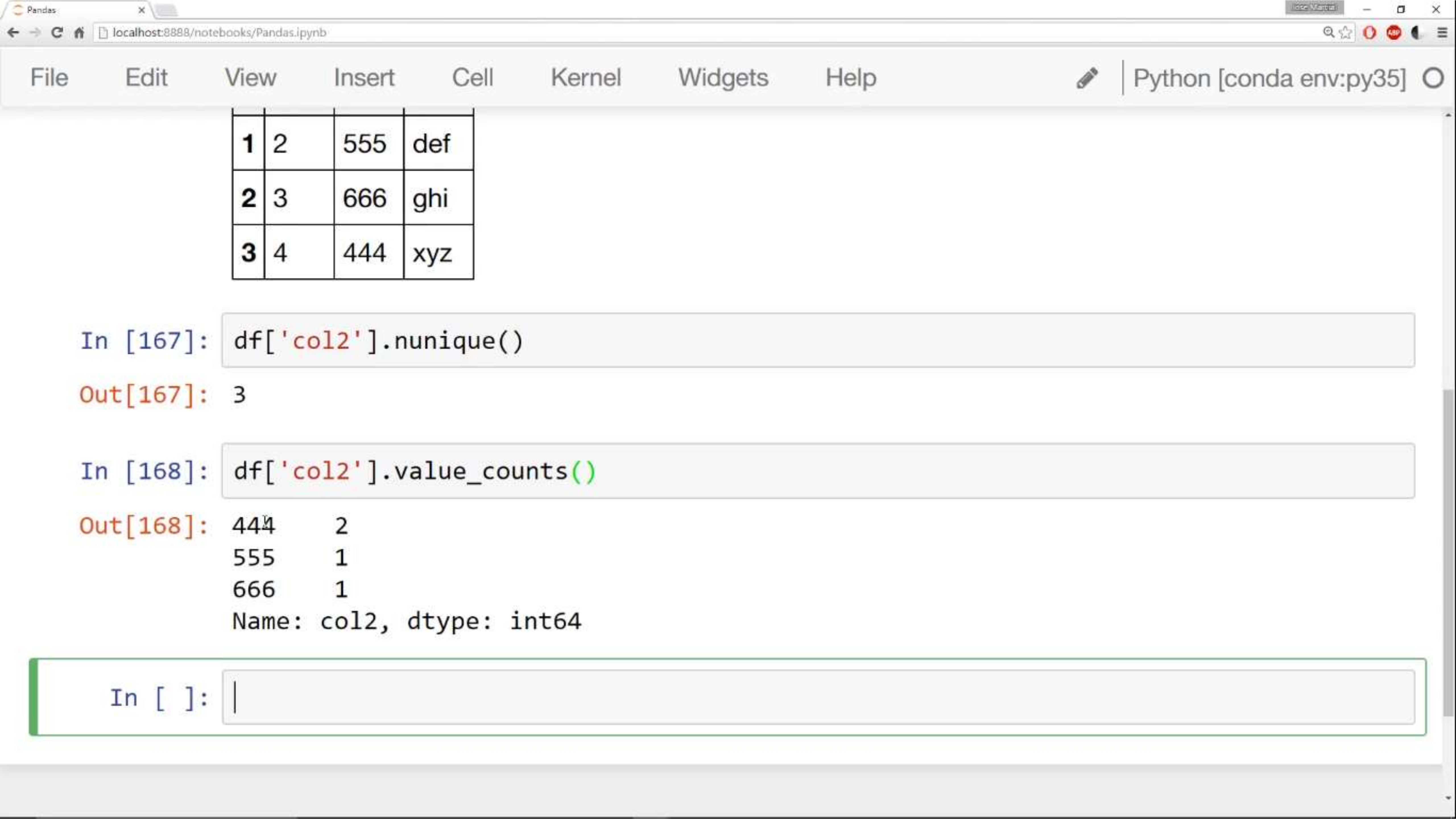
```
In [164]: len(df['col2'].unique())

Out[164]: 3

In [ ]:
```



```
Out[146]:
```

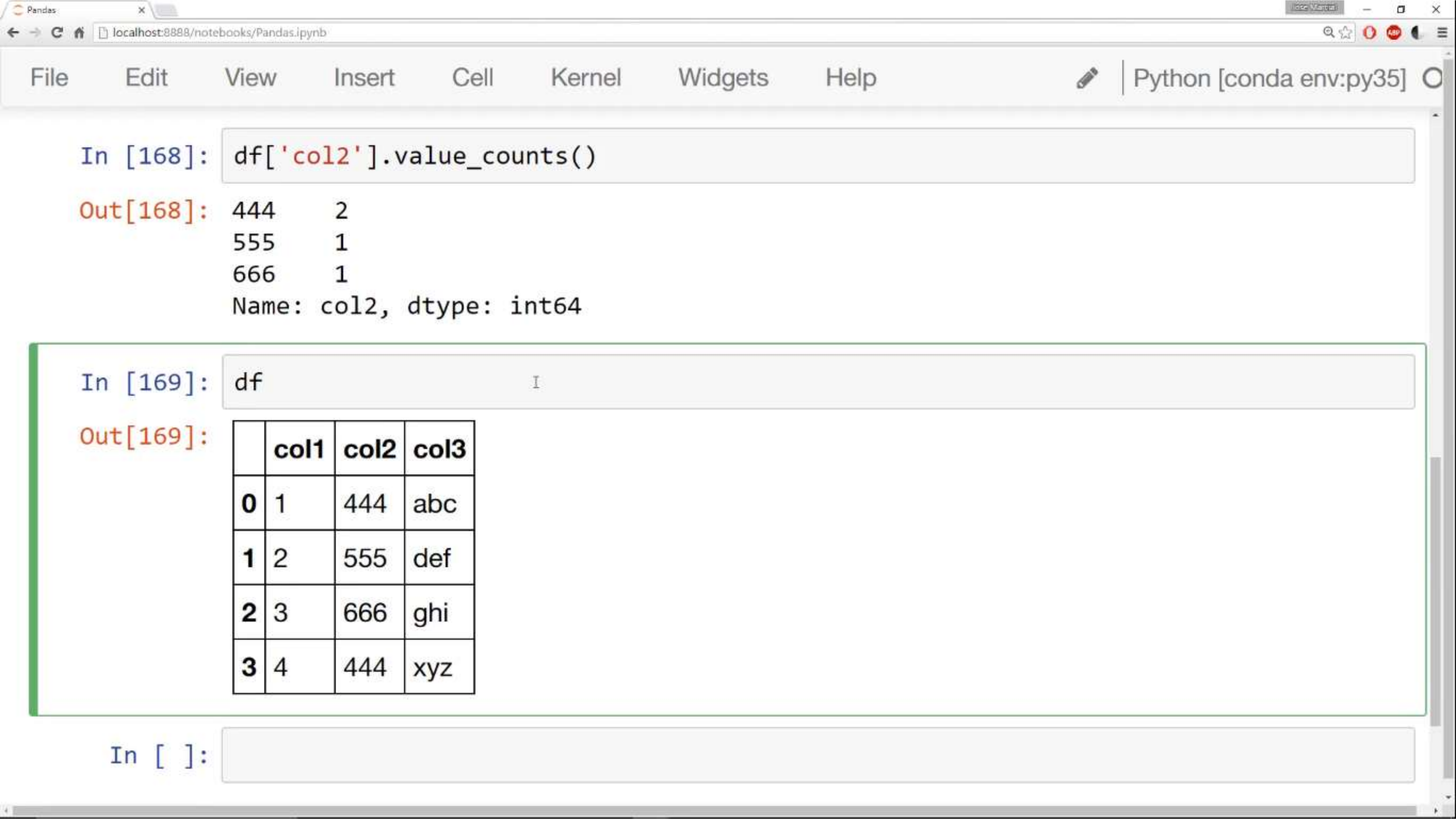
	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

```
In [165]: df['col2'].nunique()

Out[165]: 3

In [ ]:
```


Pandas
localhost:8888/notebooks/Pandas.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:py35]

1	2	555	def
2	3	666	ghi
3	4	444	xyz

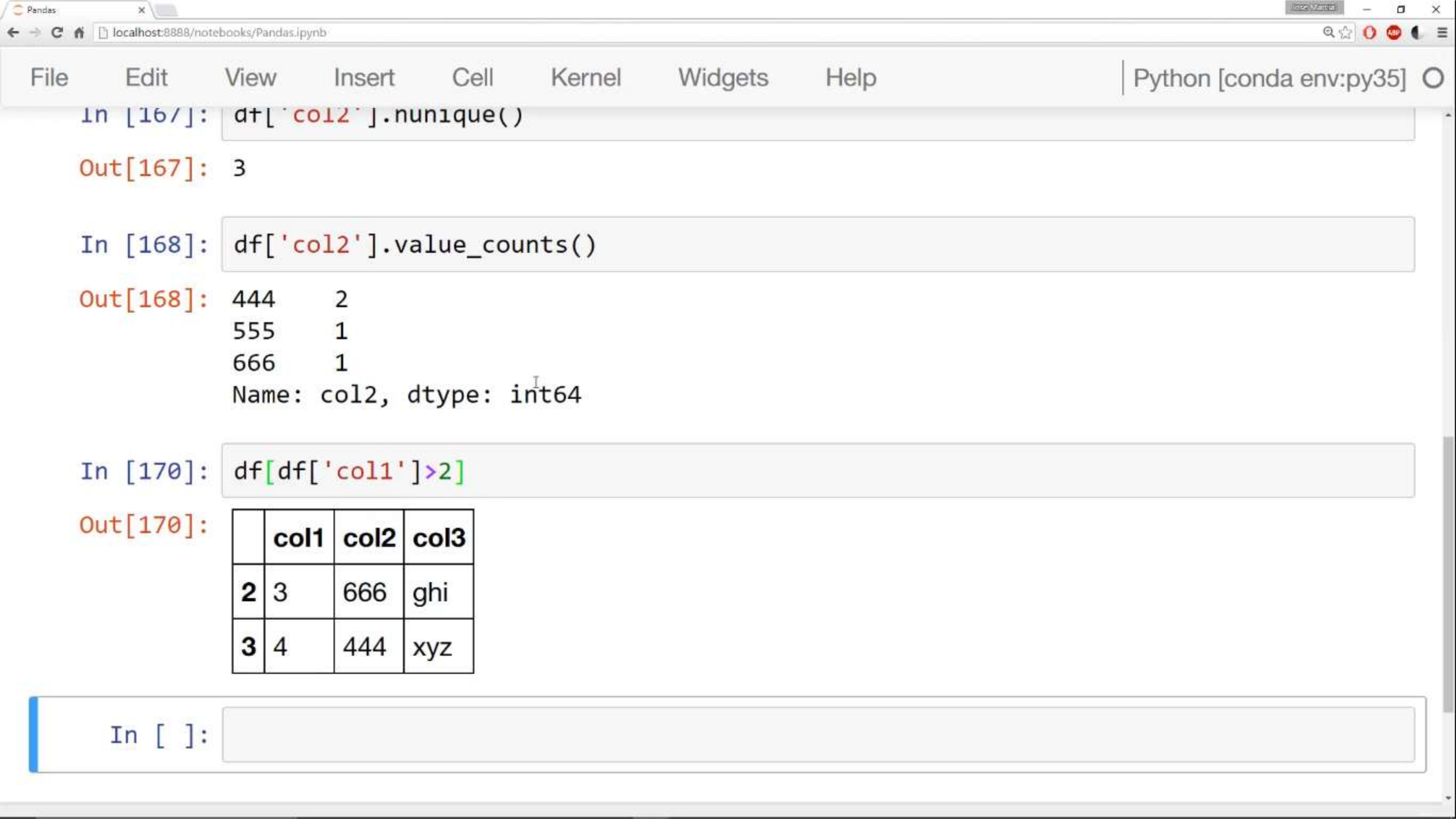
```
In [167]: df['col2'].nunique()

Out[167]: 3

In [168]: df['col2'].value_counts()

Out[168]: 444    2
          555    1
          666    1
          Name: col2, dtype: int64

In [ ]:
```


Pandas
localhost:8888/notebooks/Pandas.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:py35]

```
In [168]: df['col2'].value_counts()

Out[168]: 444    2
          555    1
          666    1
          Name: col2, dtype: int64

In [169]: df

Out[169]:
```

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

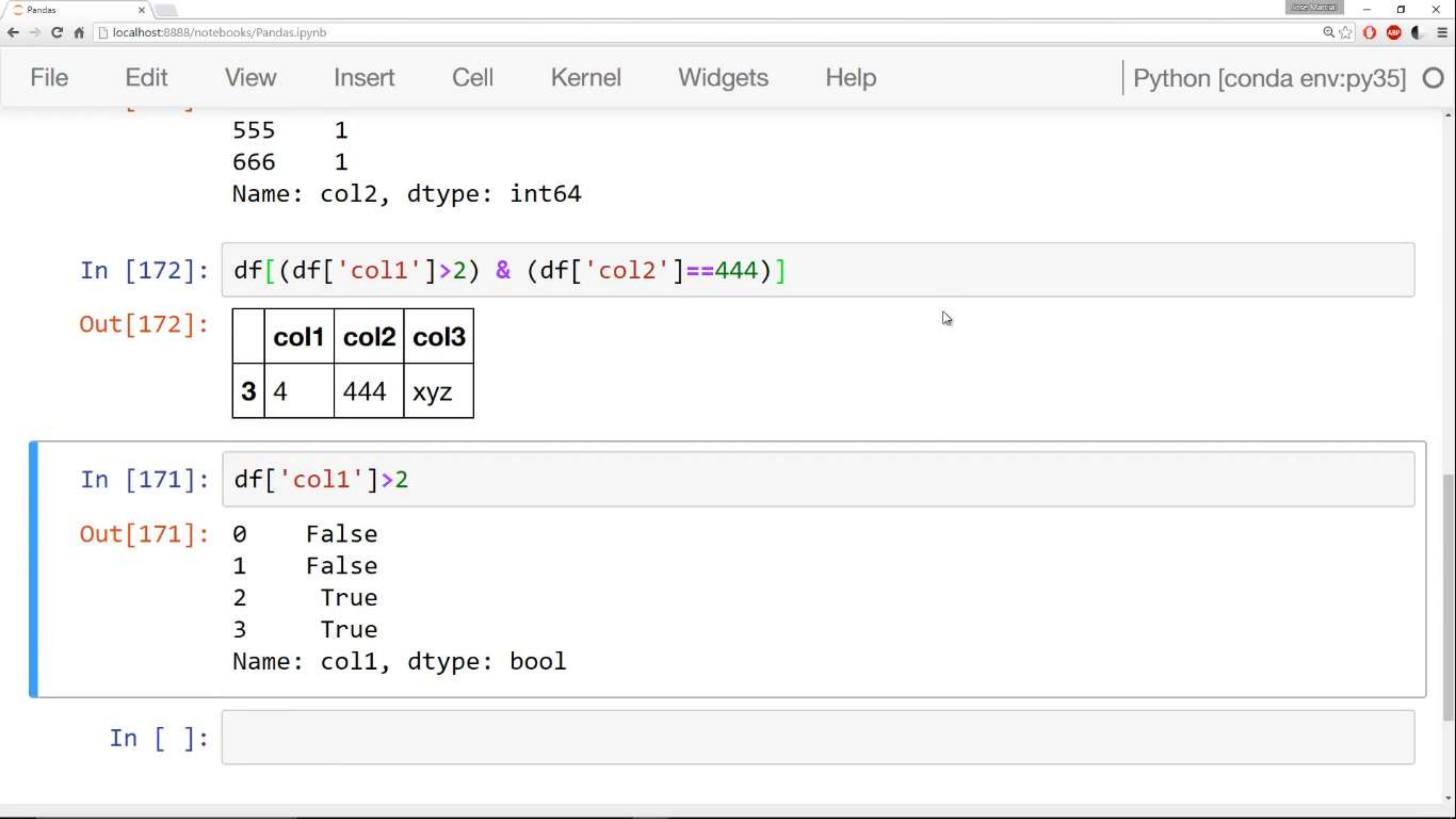
```
In [ ]:
```



```
In [167]: df['col2'].nunique()

Out[167]: 3

In [168]: df['col2'].value_counts()

Out[168]: 444    2
          555    1
          666    1
          Name: col2, dtype: int64

In [170]: df[df['col1']>2]

Out[170]:
```

	col1	col2	col3
2	3	666	ghi
3	4	444	xyz

```
In [ ]:
```



```
555    1
666    1
Name: col2, dtype: int64
```

In [172]:
```
df[(df['col1']>2) & (df['col2']==444)]
```

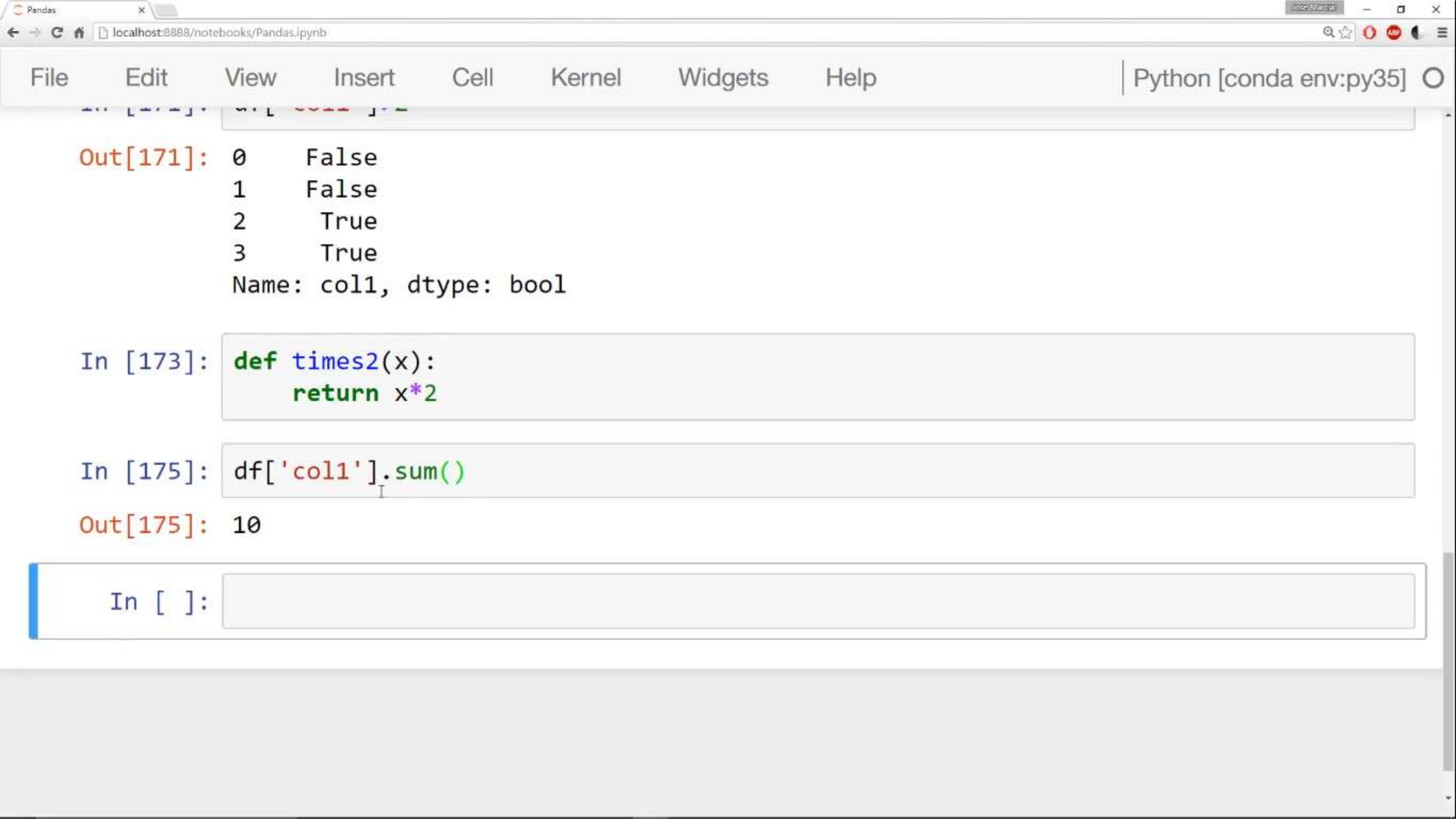
Out[172]:

	col1	col2	col3
3	4	444	xyz

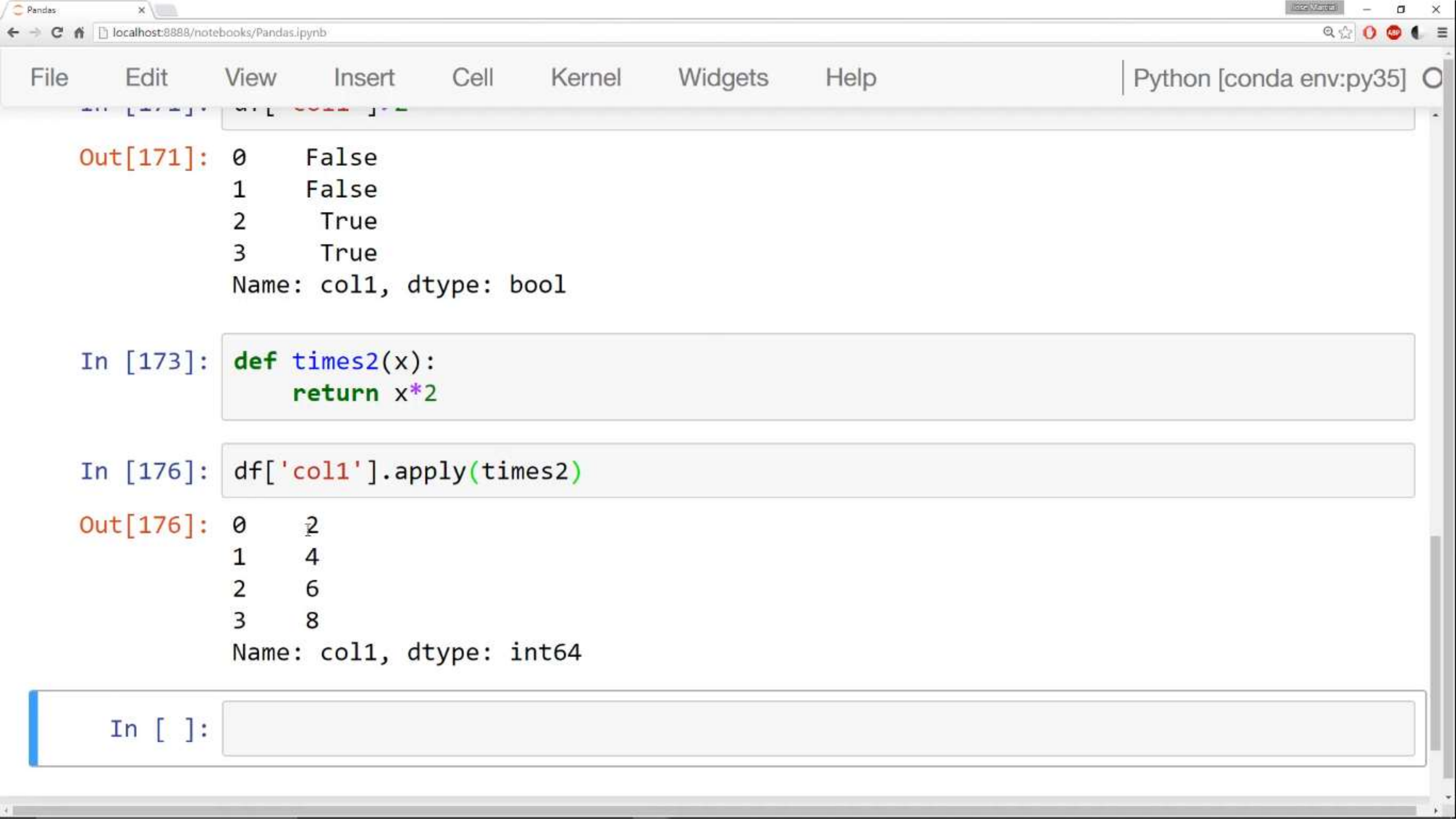
In [171]:
```
df['col1']>2
```

Out[171]:
```
0    False
1    False
2     True
3     True
Name: col1, dtype: bool
```

In [ ]:



```
In [171]: df['col1']>2

Out[171]: 0    False
          1    False
          2     True
          3     True
          Name: col1, dtype: bool

In [173]: def times2(x):
              return x*2

In [175]: df['col1'].sum()

Out[175]: 10

In [ ]:
```



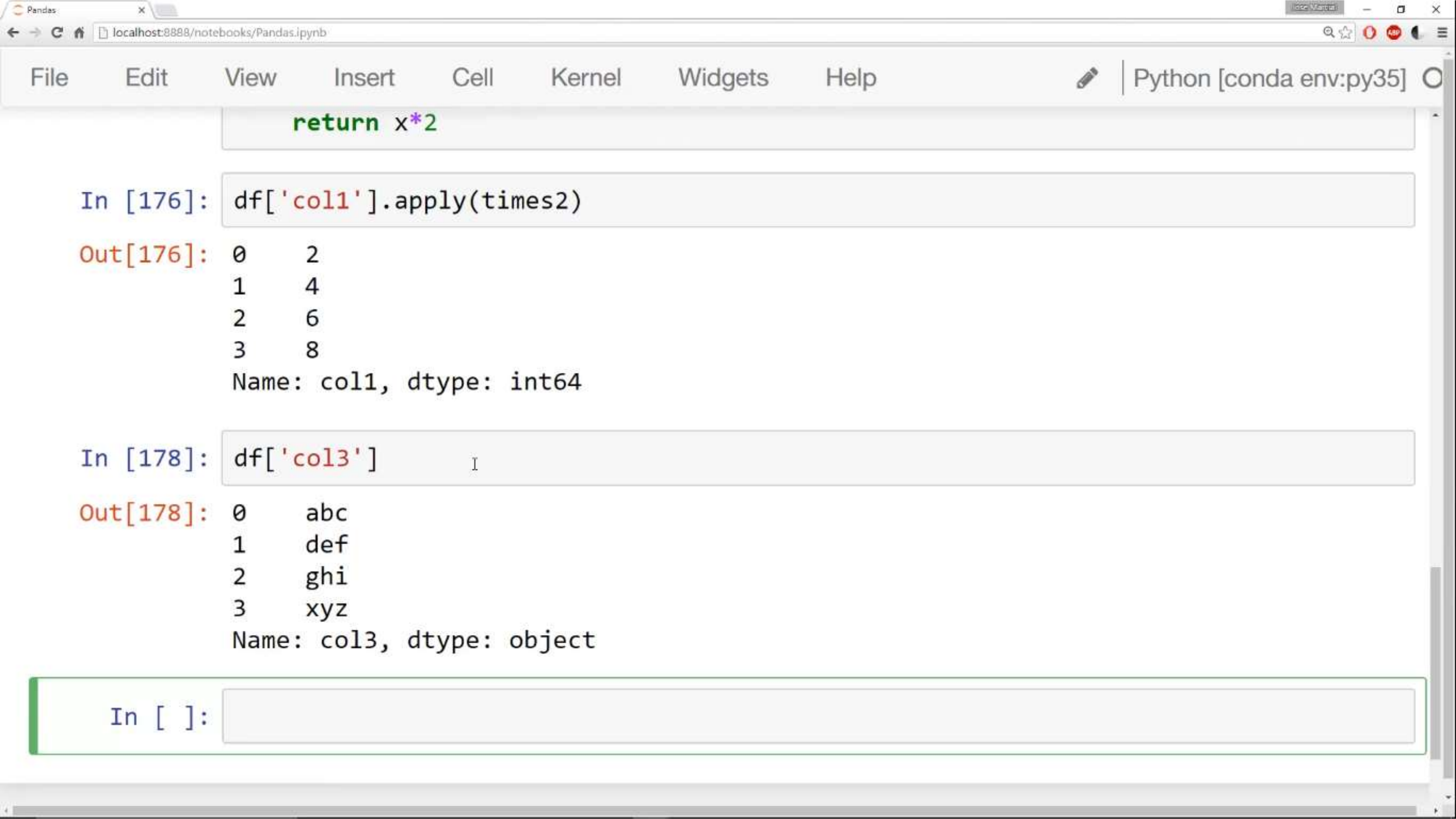
```
Out[171]: 0    False
          1    False
          2     True
          3     True
          Name: col1, dtype: bool
```

```
In [173]: def times2(x):
              return x*2
```

```
In [176]: df['col1'].apply(times2)
```

```
Out[176]: 0    2
          1    4
          2    6
          3    8
          Name: col1, dtype: int64
```

```
In [ ]:
```



```
    return x*2
```

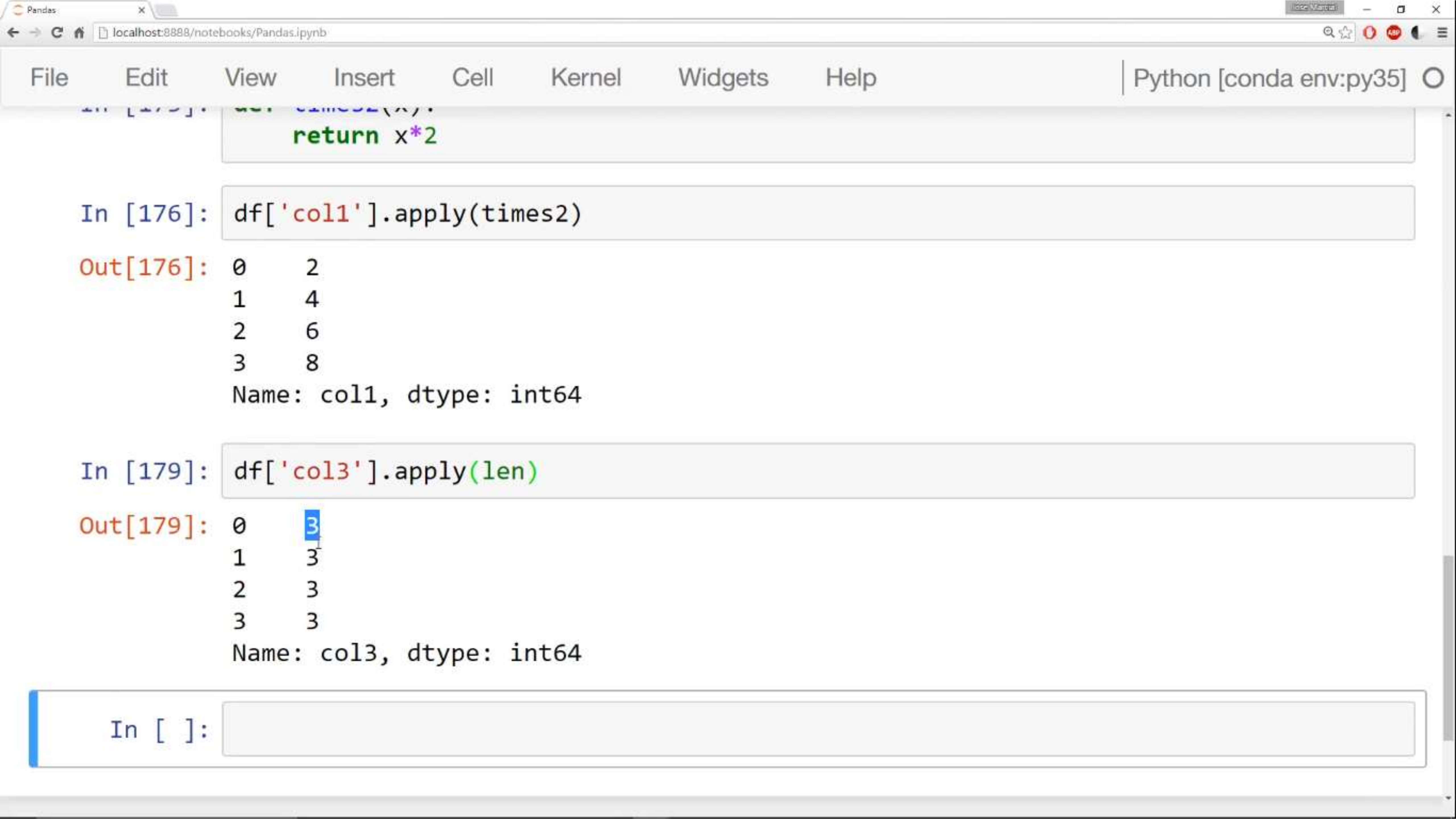
In [176]:
```
df['col1'].apply(times2)
```

Out[176]:
```
0    2
1    4
2    6
3    8
Name: col1, dtype: int64
```

In [178]:
```
df['col3']
```

Out[178]:
```
0    abc
1    def
2    ghi
3    xyz
Name: col3, dtype: object
```

In [ ]:


Pandas
localhost:8888/notebooks/Pandas.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:py35]

```
        return x*2
```

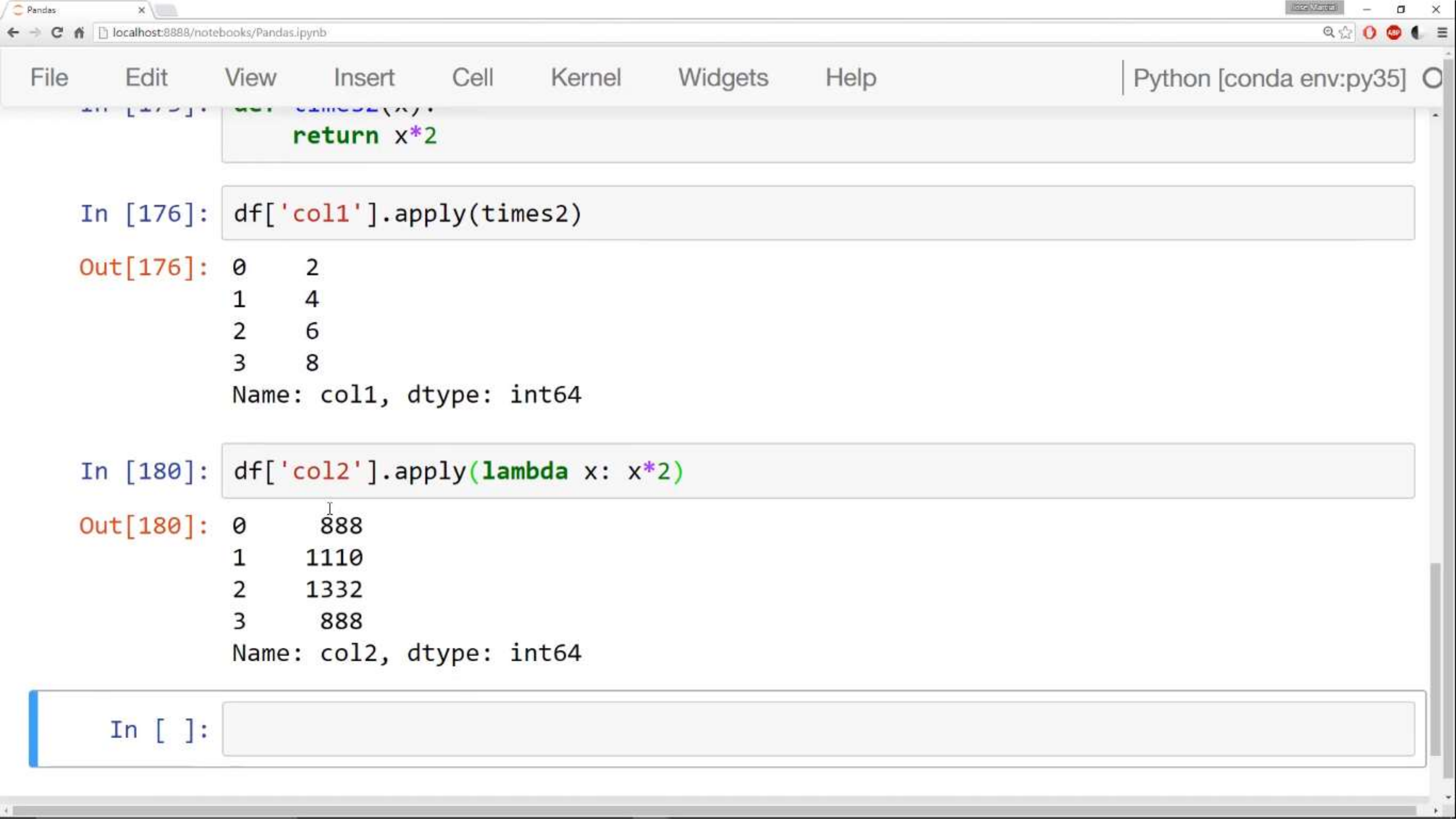
In [176]:
```
df['col1'].apply(times2)
```

Out[176]:
```
0    2
1    4
2    6
3    8
Name: col1, dtype: int64
```

In [179]:
```
df['col3'].apply(len)
```

Out[179]:
```
0    3
1    3
2    3
3    3
Name: col3, dtype: int64
```

In [ ]:



```
        return x*2
```

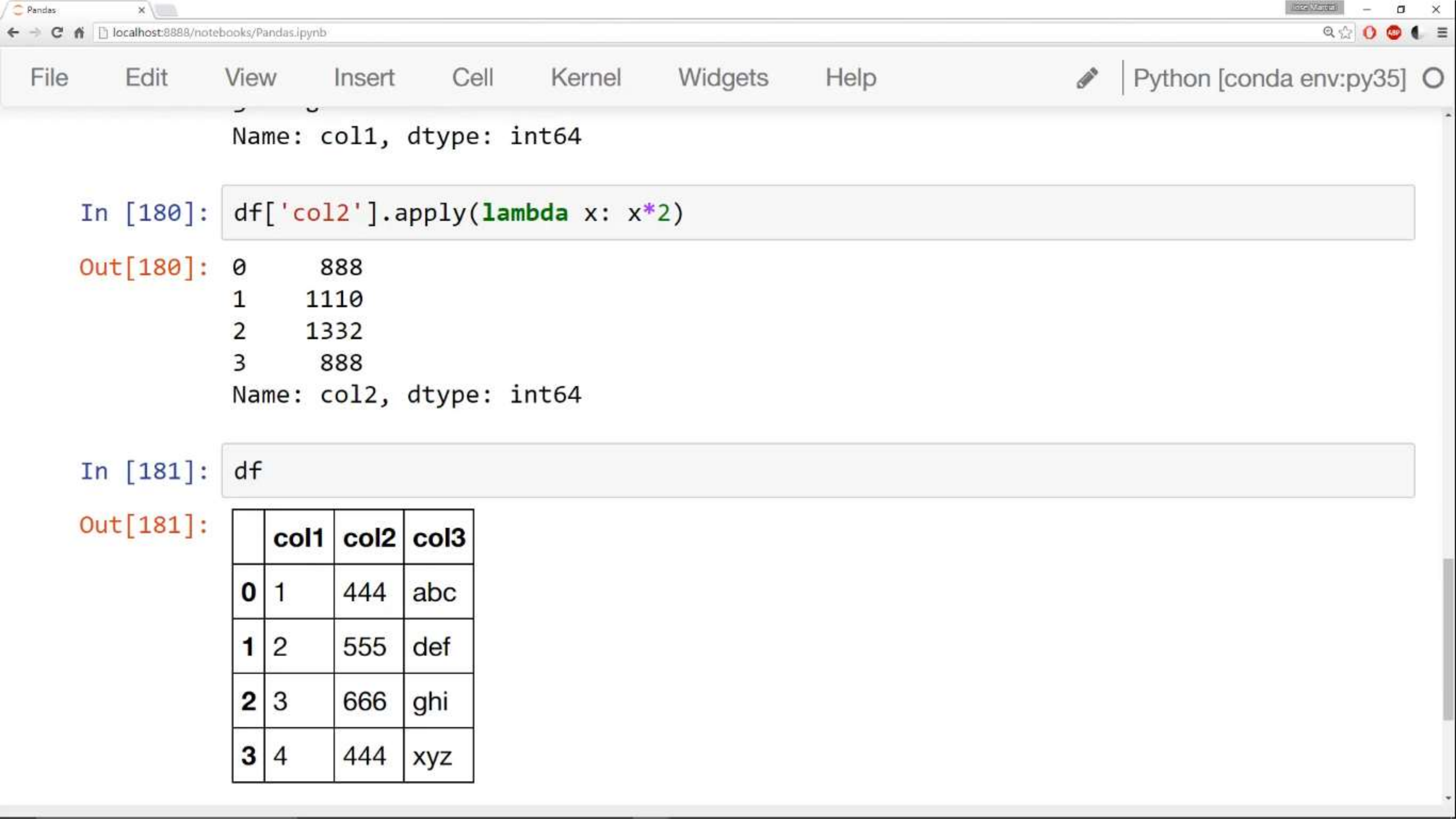
In [176]:
```
df['col1'].apply(times2)
```

Out[176]:
```
0    2
1    4
2    6
3    8
Name: col1, dtype: int64
```

In [180]:
```
df['col2'].apply(lambda x: x*2)
```

Out[180]:
```
0     888
1    1110
2    1332
3     888
Name: col2, dtype: int64
```

In [ ]:



```
Name: col1, dtype: int64
```

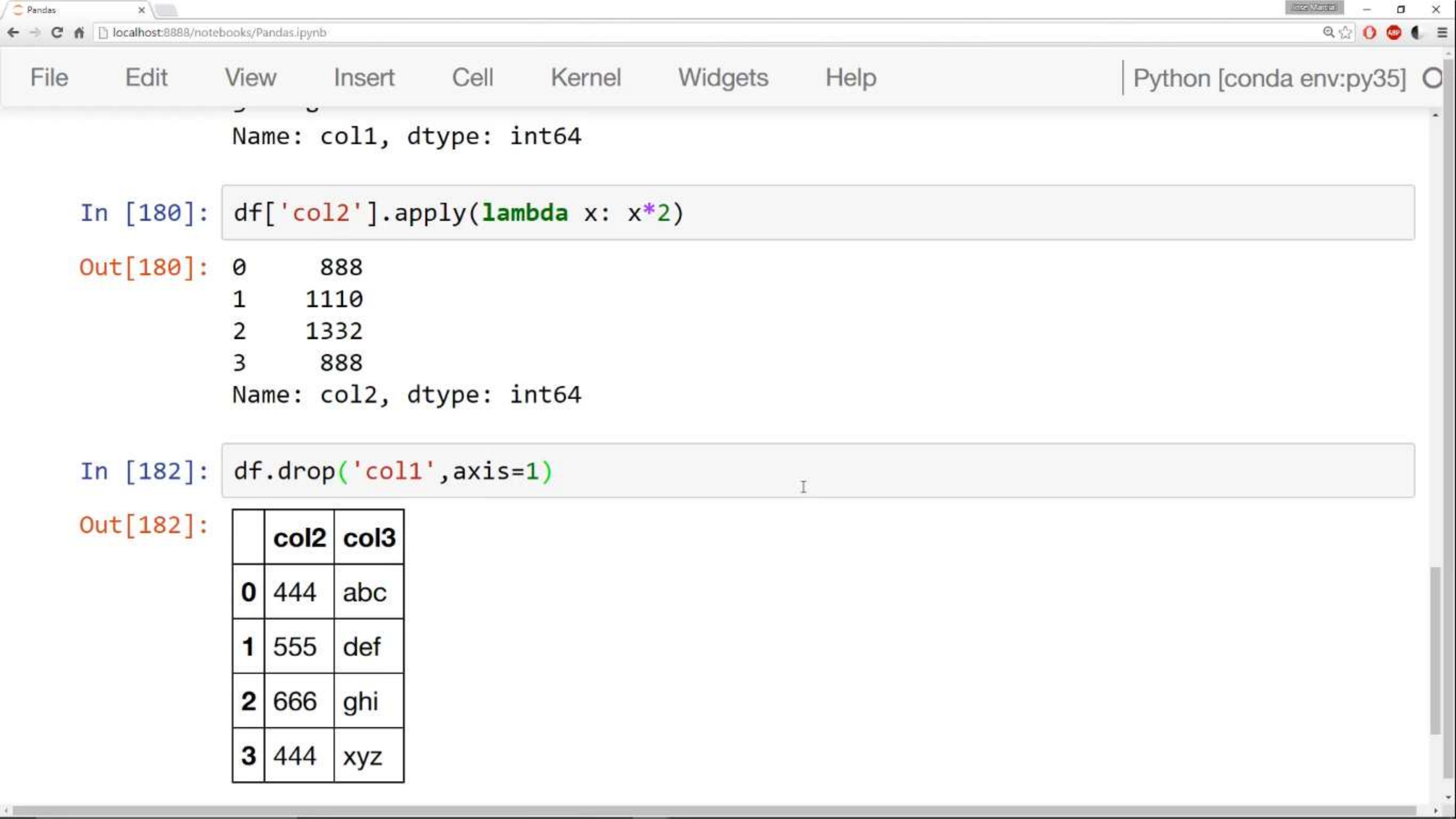
In [180]:
```
df['col2'].apply(lambda x: x*2)
```

Out[180]:
```
0     888
1    1110
2    1332
3     888
Name: col2, dtype: int64
```

In [181]:
```
df
```

Out[181]:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz



localhost:8888/notebooks/Pandas.ipynb

```
Name: col1, dtype: int64
```

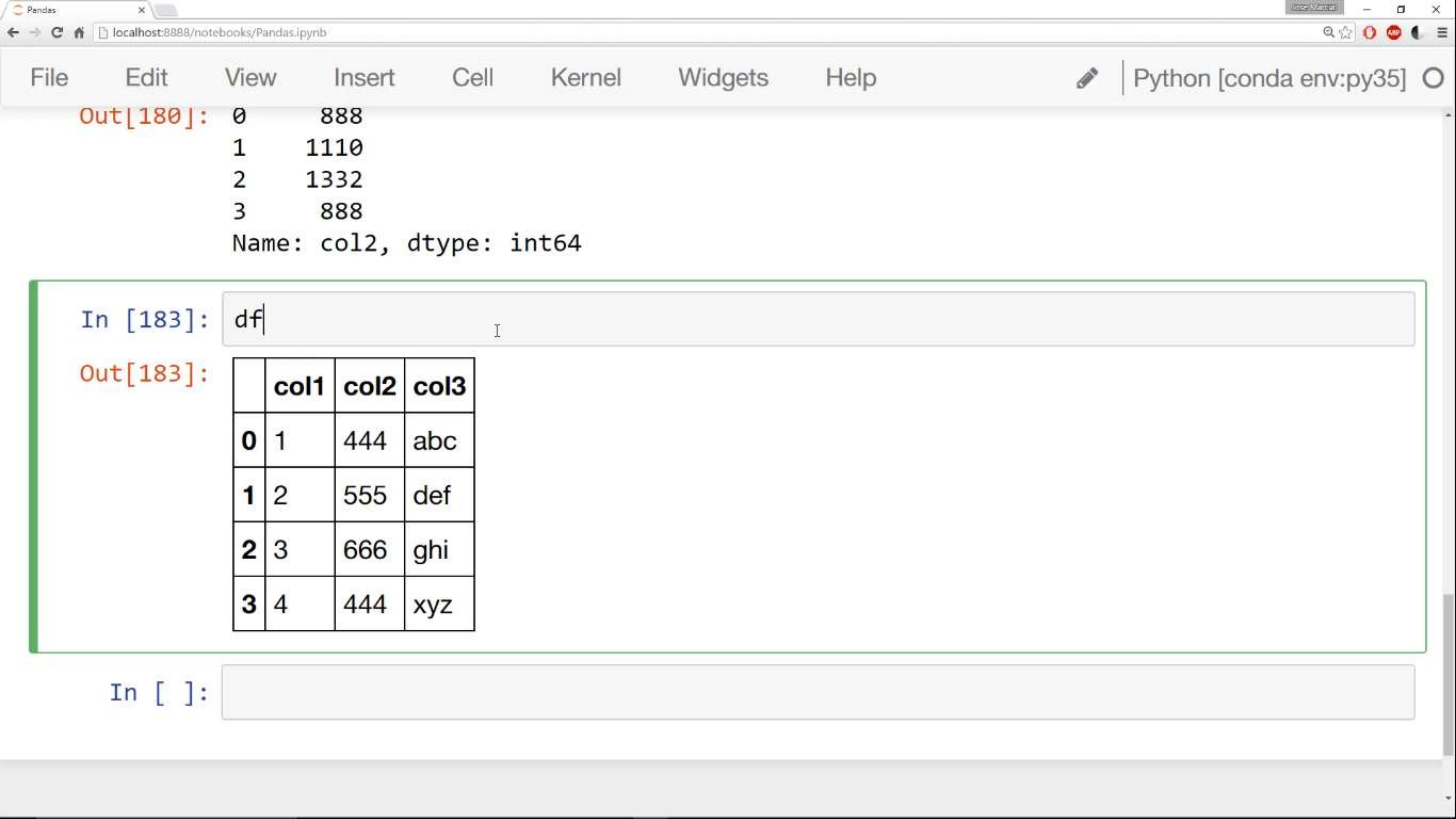
In [180]:
```
df['col2'].apply(lambda x: x*2)
```

Out[180]:
```
0     888
1    1110
2    1332
3     888
Name: col2, dtype: int64
```

In [182]:
```
df.drop('col1',axis=1)
```

Out[182]:

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz



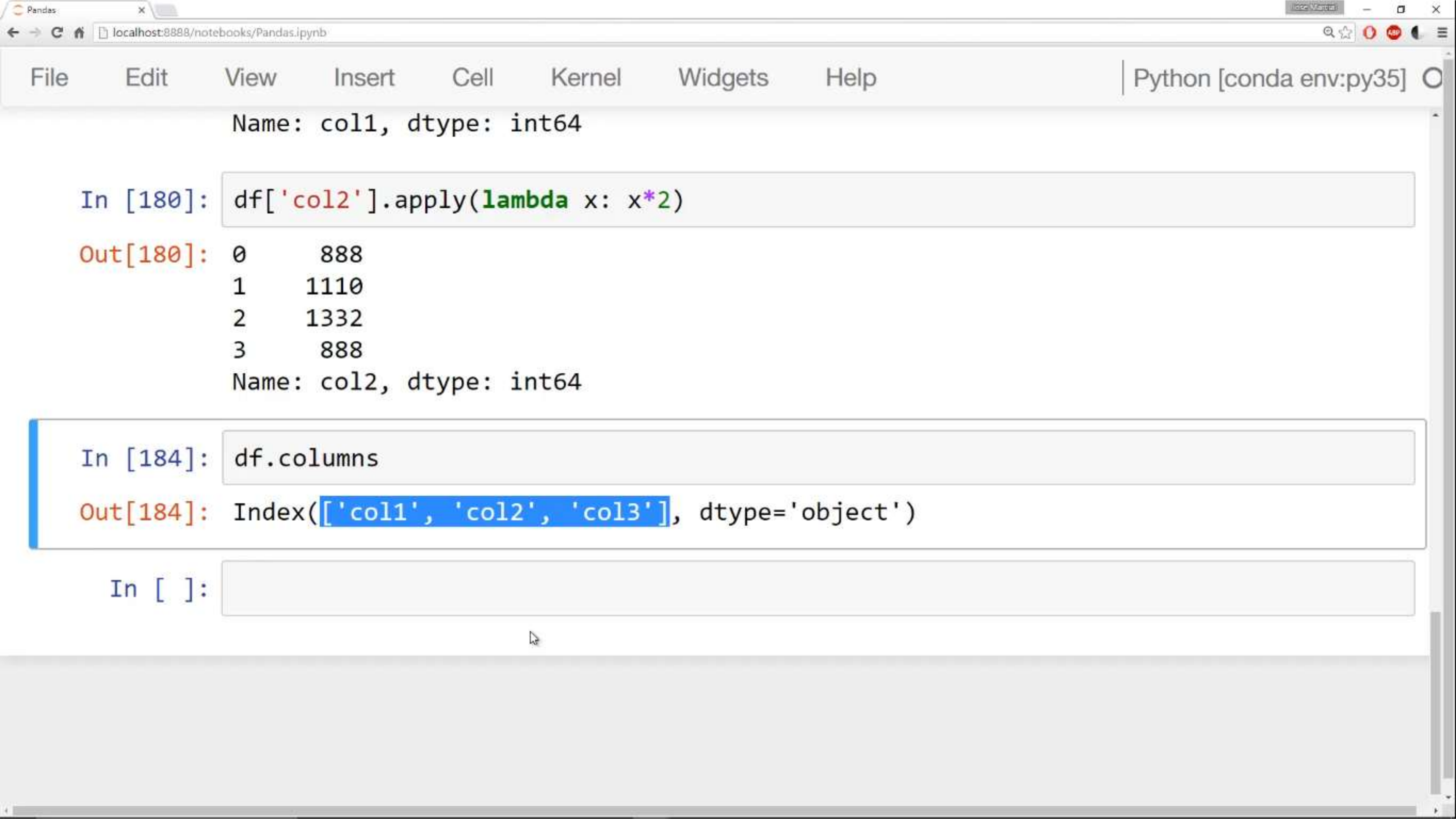
```
Out[180]: 0     888
          1    1110
          2    1332
          3     888
          Name: col2, dtype: int64
```

```
In [183]: df
```

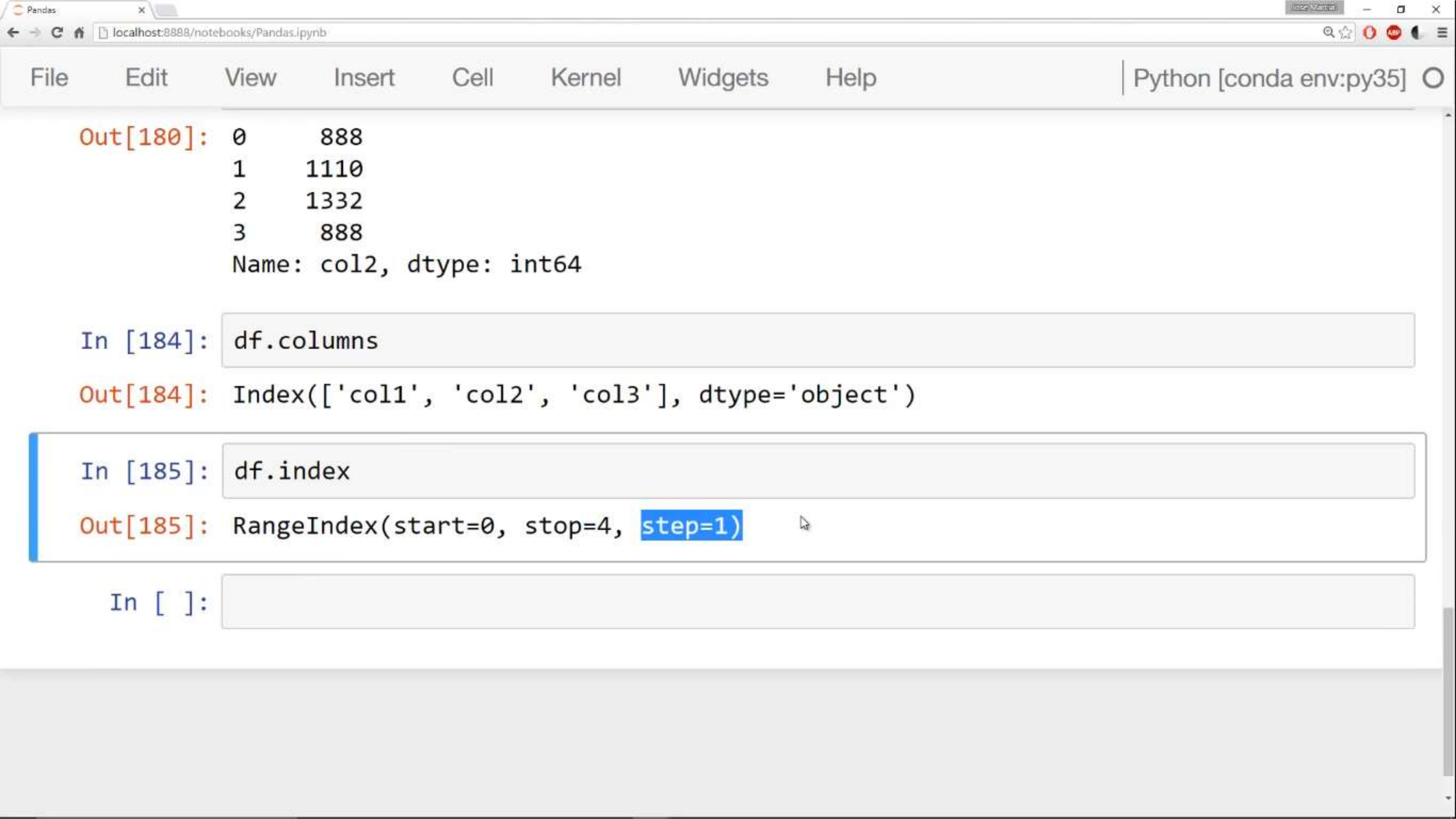
Out[183]:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

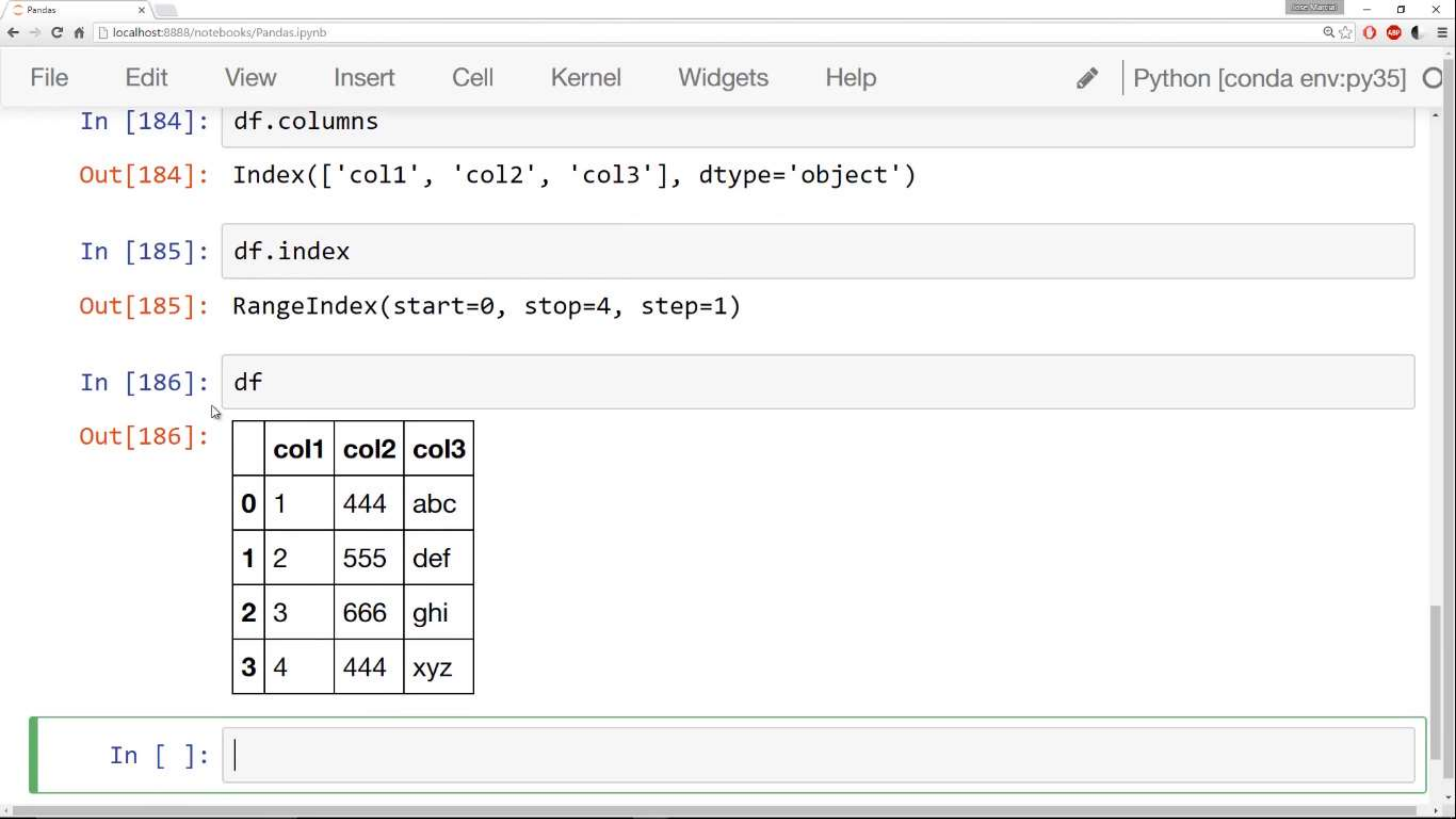
```
In [ ]:
```


Pandas
localhost:8888/notebooks/Pandas.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:py35]

```
Name: col1, dtype: int64

In [180]: df['col2'].apply(lambda x: x*2)

Out[180]: 0     888
          1    1110
          2    1332
          3     888
          Name: col2, dtype: int64

In [184]: df.columns

Out[184]: Index(['col1', 'col2', 'col3'], dtype='object')

In [ ]:
```



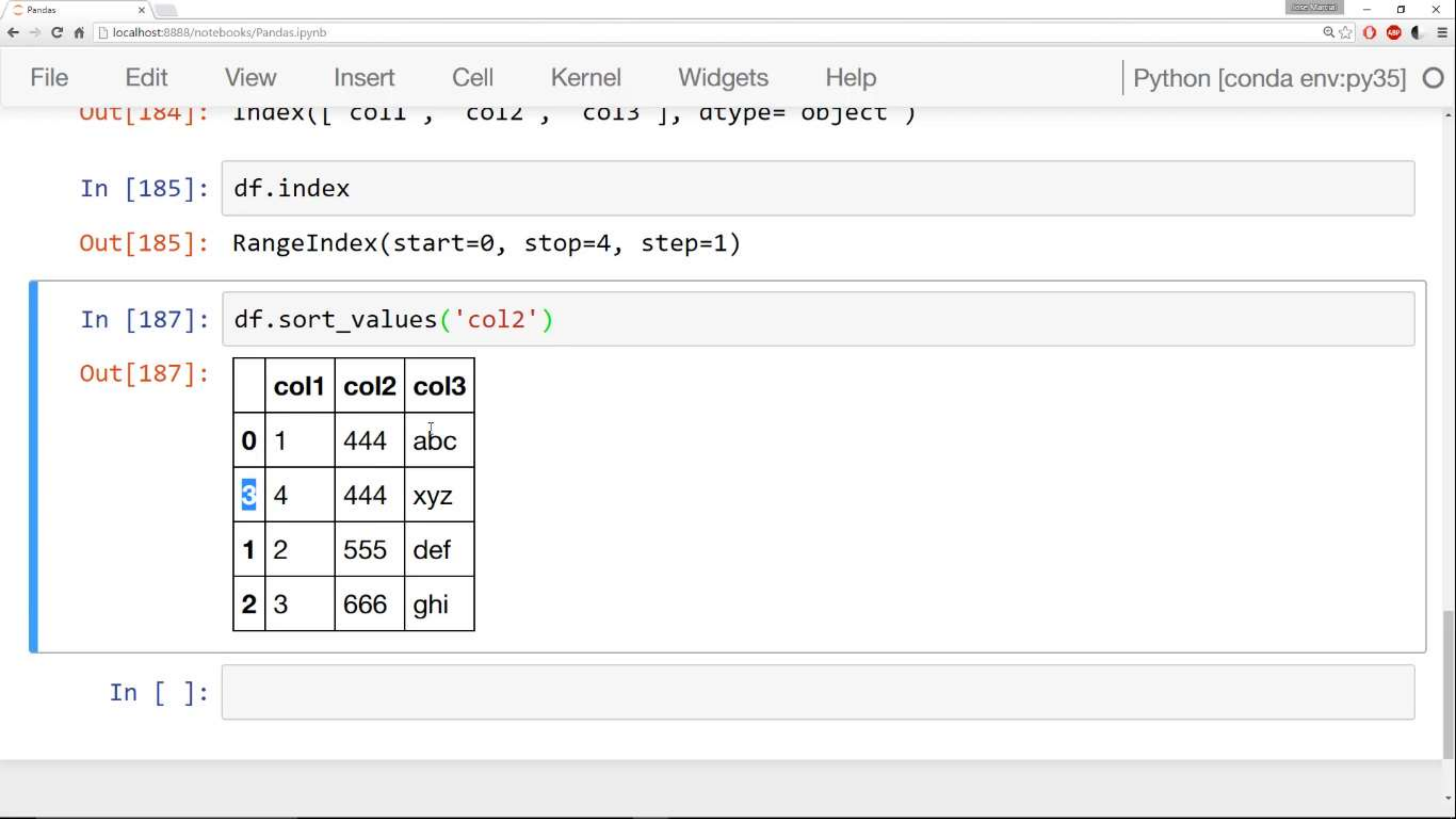
```
Out[180]: 0     888
          1    1110
          2    1332
          3     888
          Name: col2, dtype: int64

In [184]: df.columns

Out[184]: Index(['col1', 'col2', 'col3'], dtype='object')

In [185]: df.index

Out[185]: RangeIndex(start=0, stop=4, step=1)

In [ ]:
```



```
In [184]: df.columns

Out[184]: Index(['col1', 'col2', 'col3'], dtype='object')

In [185]: df.index

Out[185]: RangeIndex(start=0, stop=4, step=1)

In [186]: df

Out[186]:
```

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

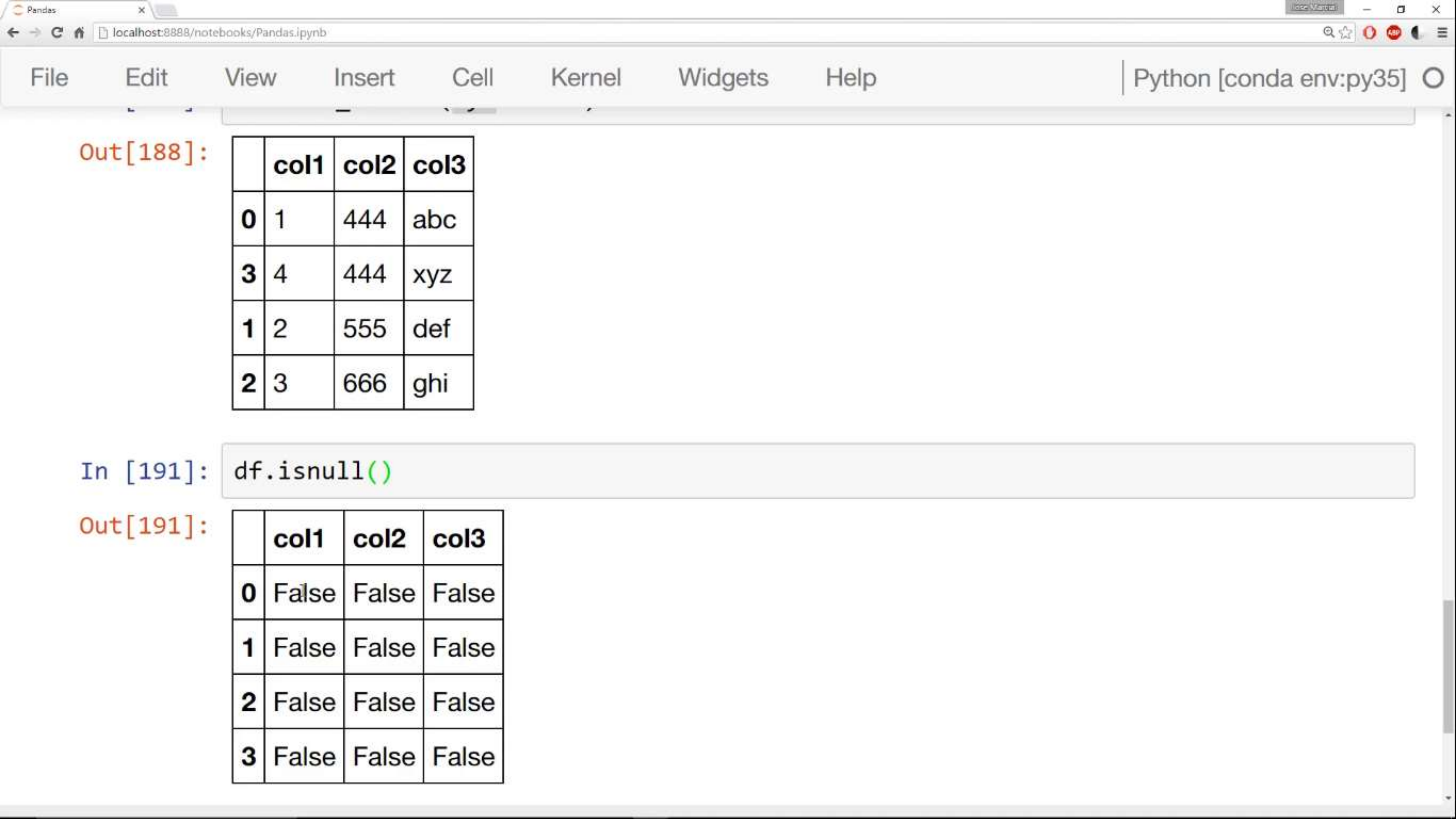
```
In [ ]:
```



```
Out[184]: Index(['col1', 'col2', 'col3'], dtype='object')

In [185]: df.index

Out[185]: RangeIndex(start=0, stop=4, step=1)

In [187]: df.sort_values('col2')

Out[187]:
```

	col1	col2	col3
0	1	444	abc
3	4	444	xyz
1	2	555	def
2	3	666	ghi

```
In [ ]:
```


Pandas
localhost:8888/notebooks/Pandas.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:py35]

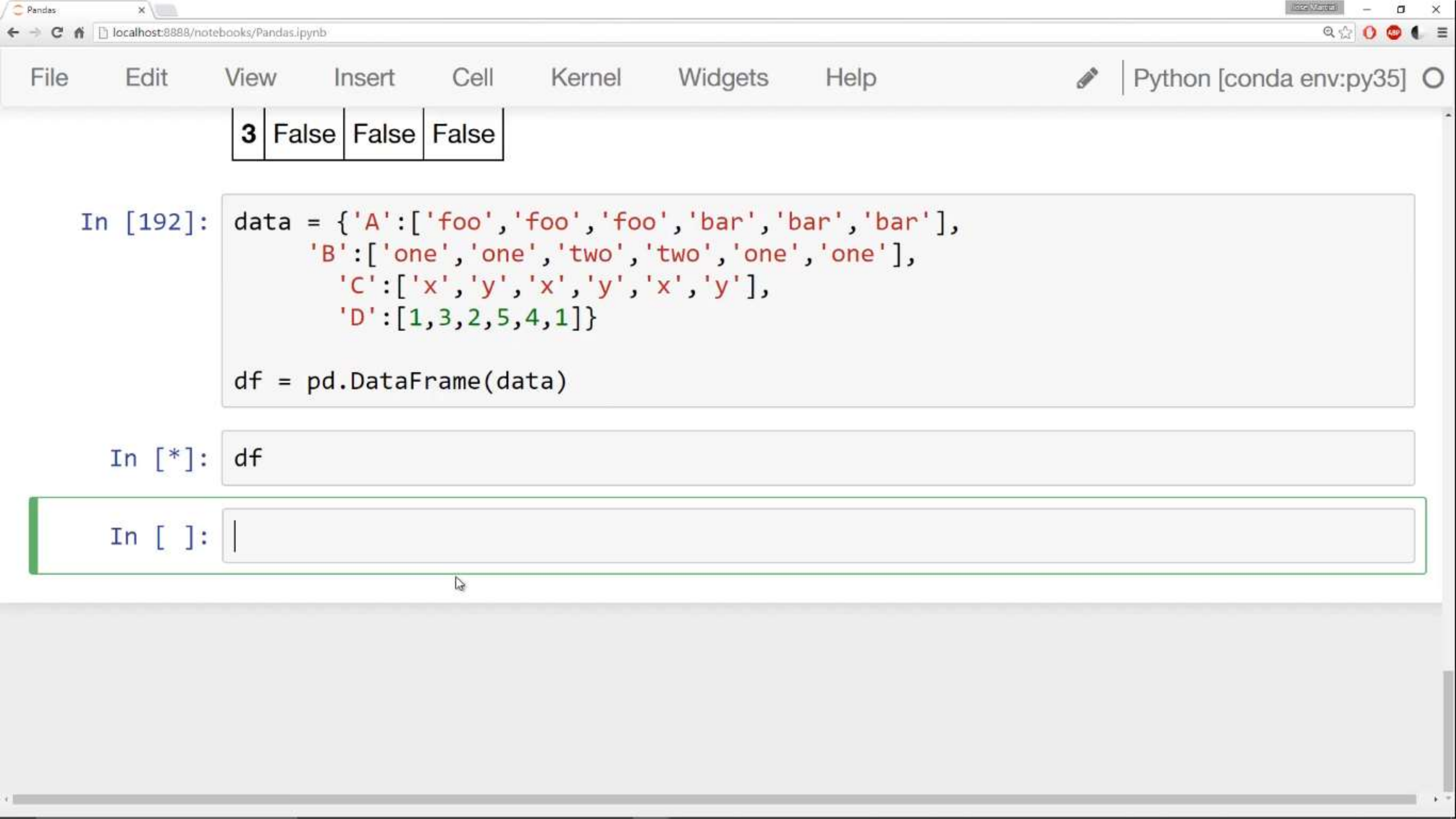
Out[188]:

	col1	col2	col3
0	1	444	abc
3	4	444	xyz
1	2	555	def
2	3	666	ghi

In [191]:
```
df.isnull()
```

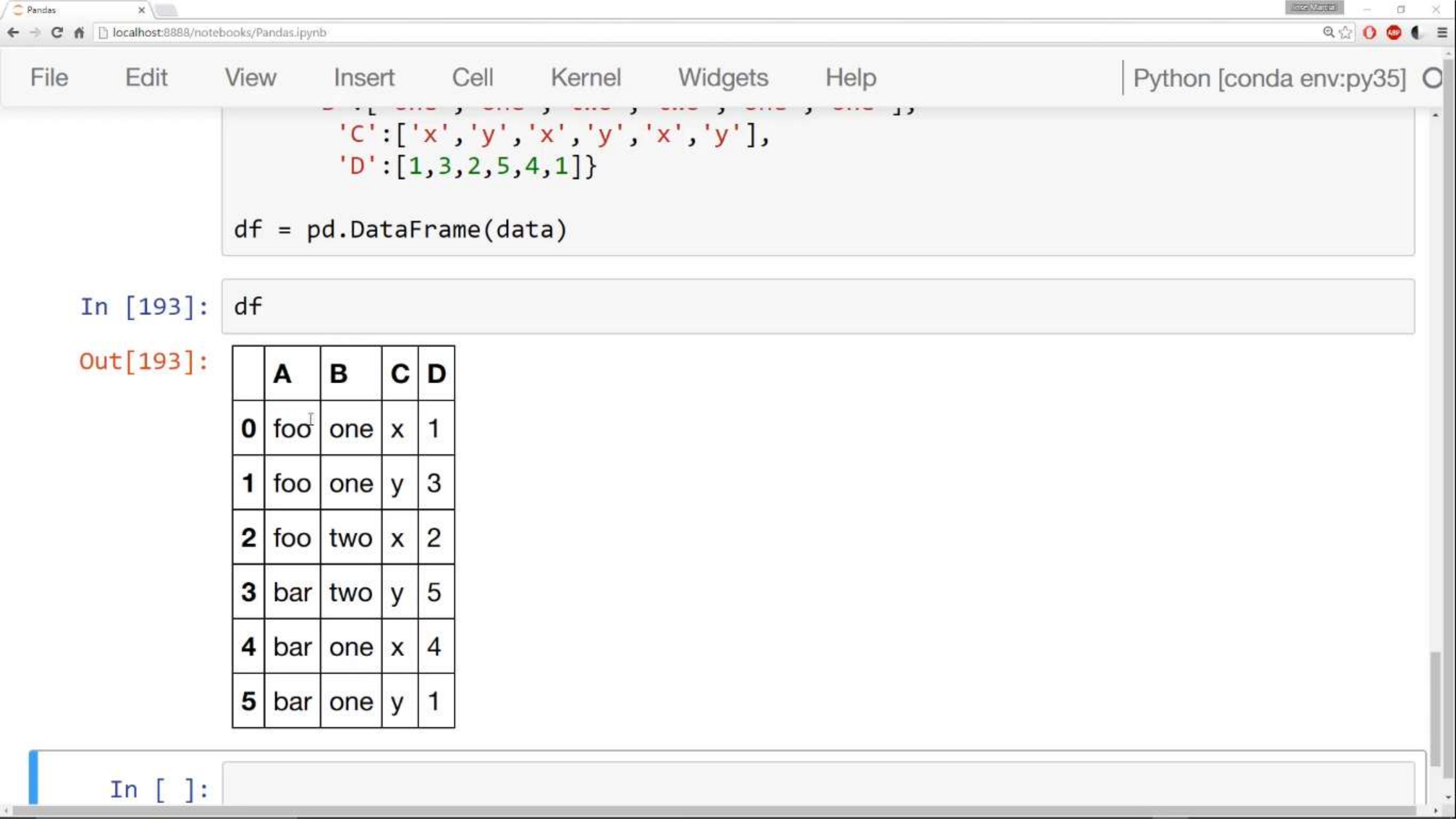
Out[191]:

	col1	col2	col3
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False



3	False	False	False

```
In [192]: data = {'A':['foo','foo','foo','bar','bar','bar'],
               'B':['one','one','two','two','one','one'],
                 'C':['x','y','x','y','x','y'],
                 'D':[1,3,2,5,4,1]}

          df = pd.DataFrame(data)

In [*]: df

In [ ]:
```



```
      'B':['one','one','two','two','one','one'],
       'C':['x','y','x','y','x','y'],
       'D':[1,3,2,5,4,1]}

df = pd.DataFrame(data)
```

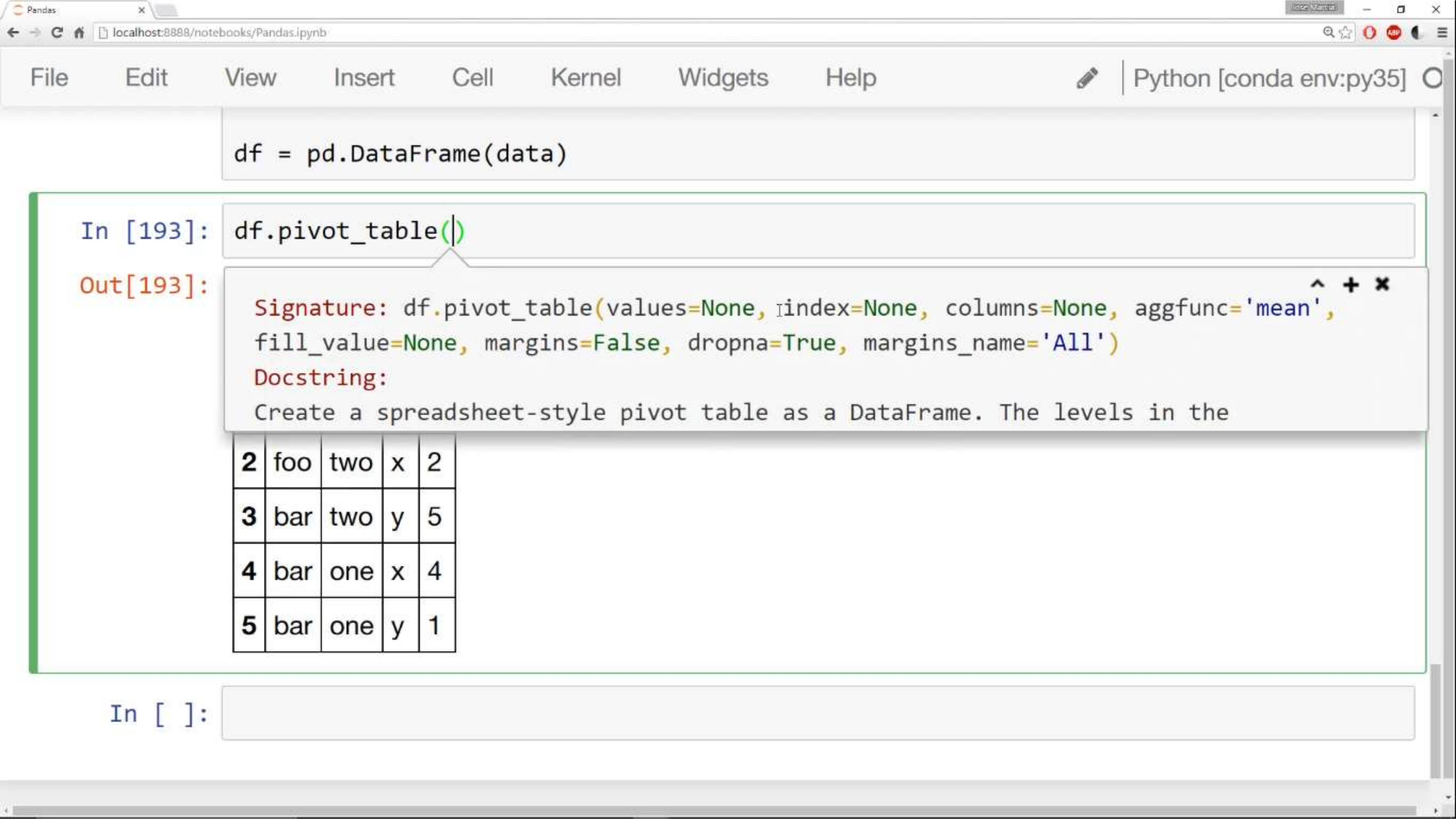
In [193]:

```
df
```

Out[193]:

	A	B	C	D
0	foo	one	x	1
1	foo	one	y	3
2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

In [ ]:



```
df = pd.DataFrame(data)
```

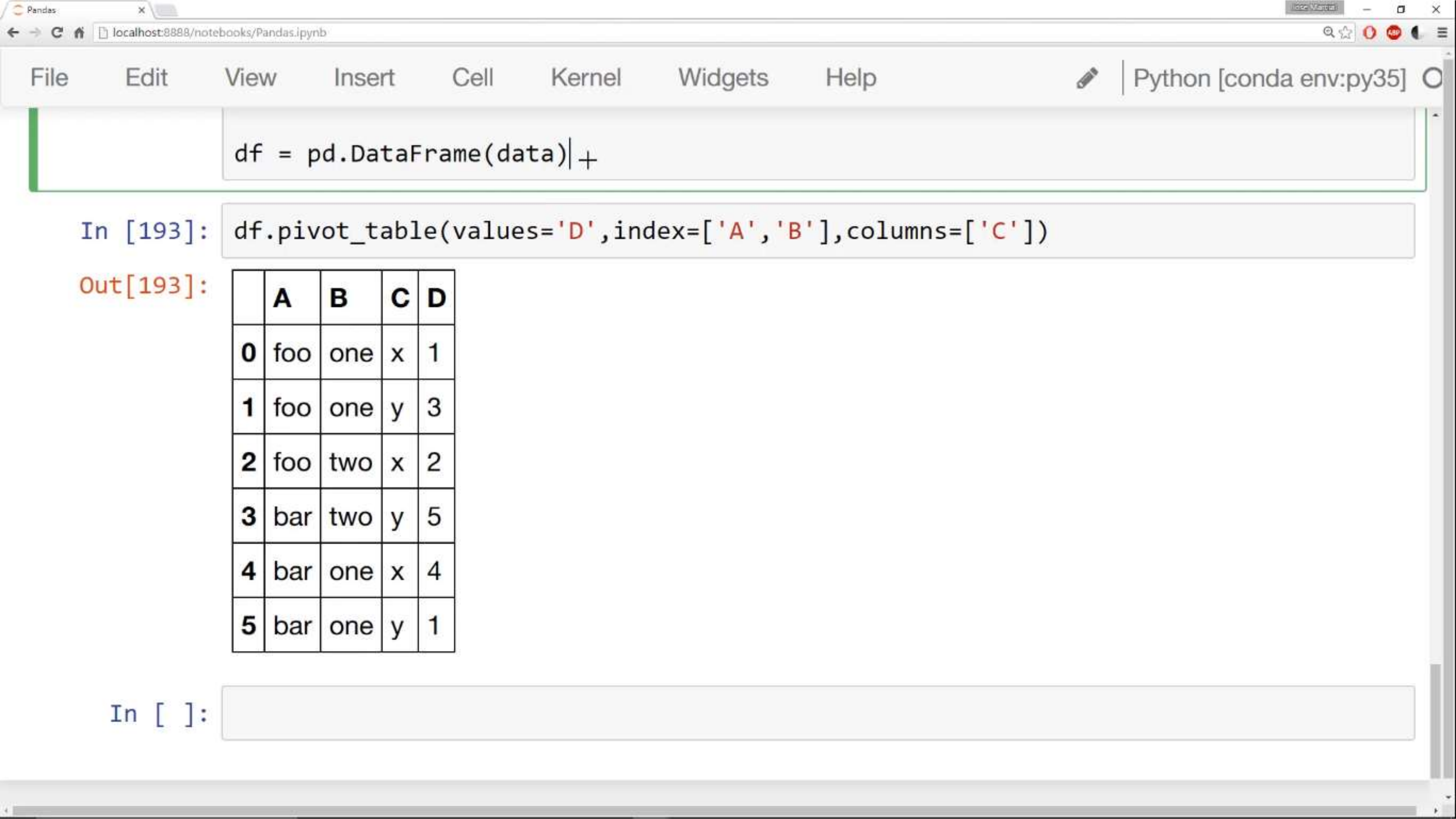
In [193]:
```
df.pivot_table()
```

Out[193]:

```
Signature: df.pivot_table(values=None, index=None, columns=None, aggfunc='mean',
fill_value=None, margins=False, dropna=True, margins_name='All')
Docstring:
Create a spreadsheet-style pivot table as a DataFrame. The levels in the
```

2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

In [ ]:



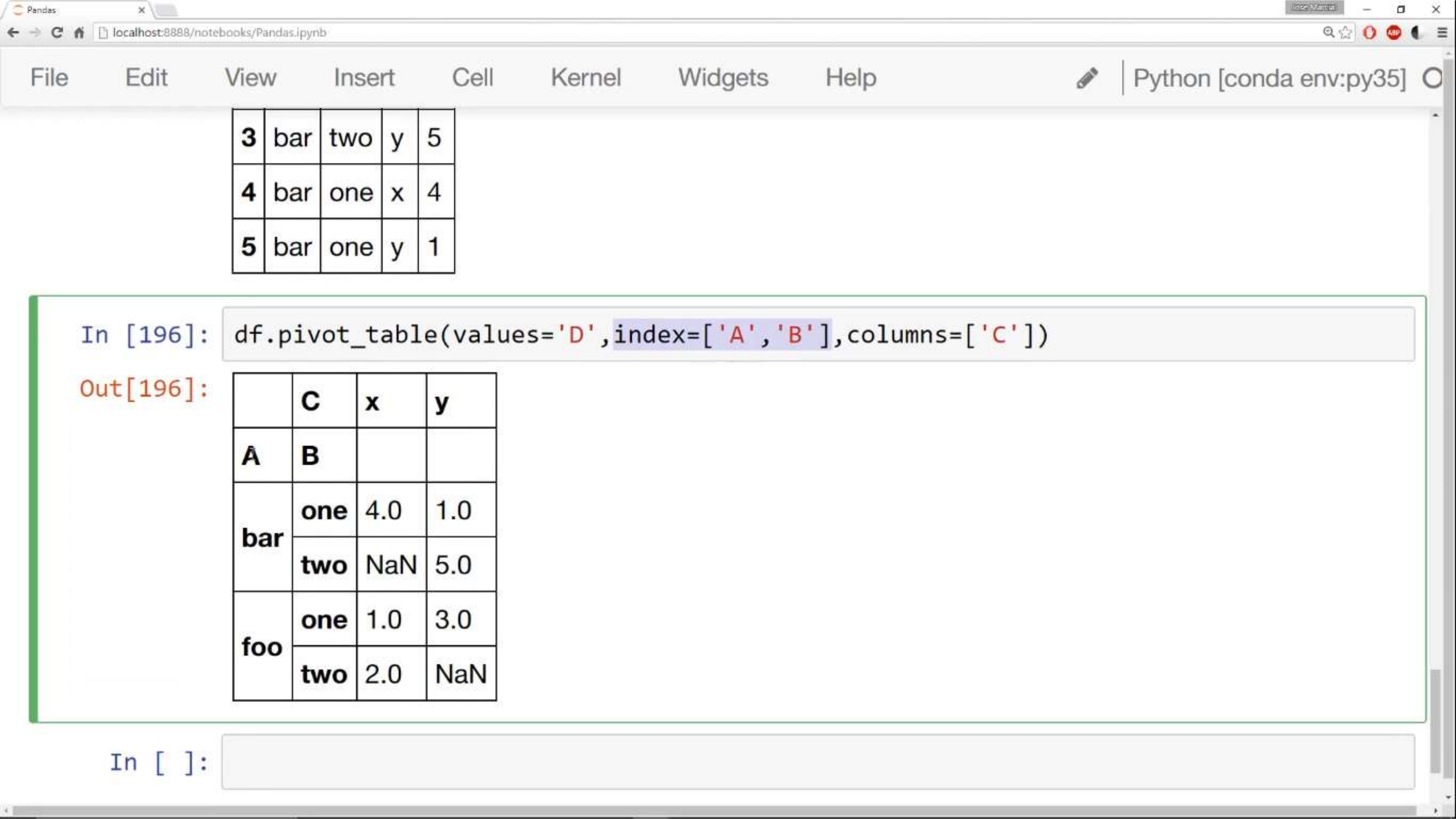
```
df = pd.DataFrame(data)
```

```
In [193]: df.pivot_table(values='D',index=['A','B'],columns=['C'])
```

Out[193]:

	A	B	C	D
0	foo	one	x	1
1	foo	one	y	3
2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

```
In [ ]:
```



3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

```
In [196]: df.pivot_table(values='D',index=['A','B'],columns=['C'])
```

Out[196]:

	C	**x**	**y**
A	**B**		
bar	**one**	4.0	1.0
	two	NaN	5.0
foo	**one**	1.0	3.0
	two	2.0	NaN

```
In [ ]:
```