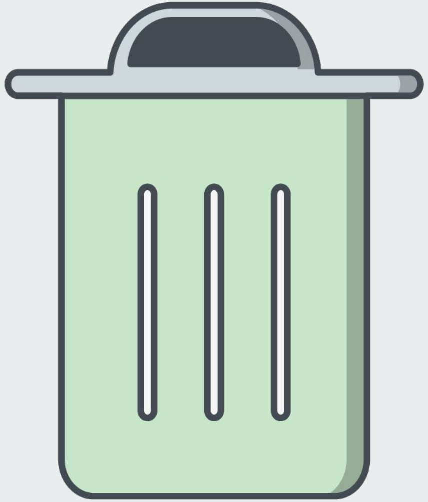
‘Del’

Keyword In classes

Presented by:
Incendero

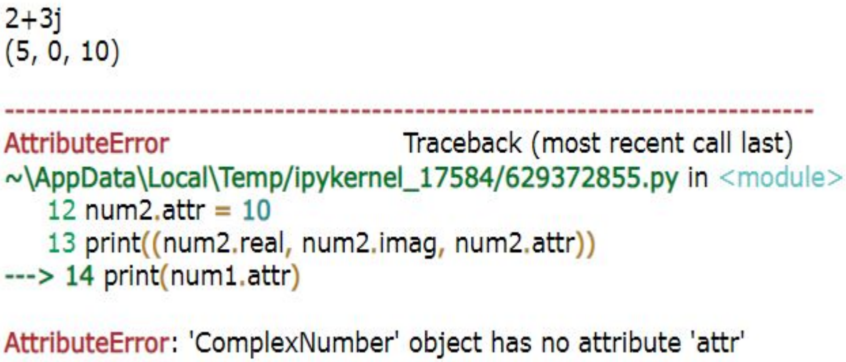


```python
class ComplexNumber:
    def __init__(self, r=0, i=0):
        self.real = r
        self.imag = i

    def get_data(self):
        print(f'{self.real}+{self.imag}j')

num1 = ComplexNumber(2, 3)
num1.get_data()
num2 = ComplexNumber(5)
num2.attr = 10
print((num2.real, num2.imag, num2.attr))
print(num1.attr)
```

Output:



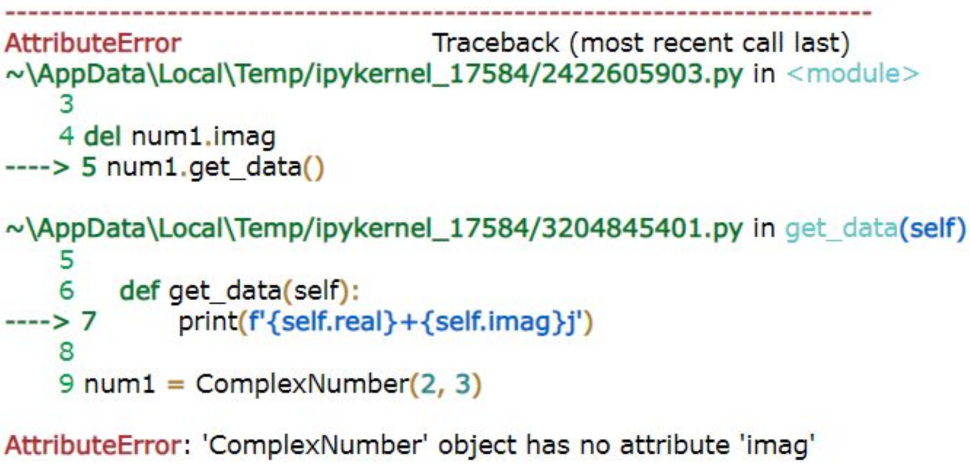
```
2+3j
(5, 0, 10)

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_17584/629372855.py in <module>
     12 num2.attr = 10
     13 print((num2.real, num2.imag, num2.attr))
---> 14 print(num1.attr)

AttributeError: 'ComplexNumber' object has no attribute 'attr'
```

```
# Deleting Attributes and Objects
#Any attribute of an object can be deleted anytime, using the del
statement. Try the following on the Python shell to see the
output.

del num1.imag
num1.get_data()
```



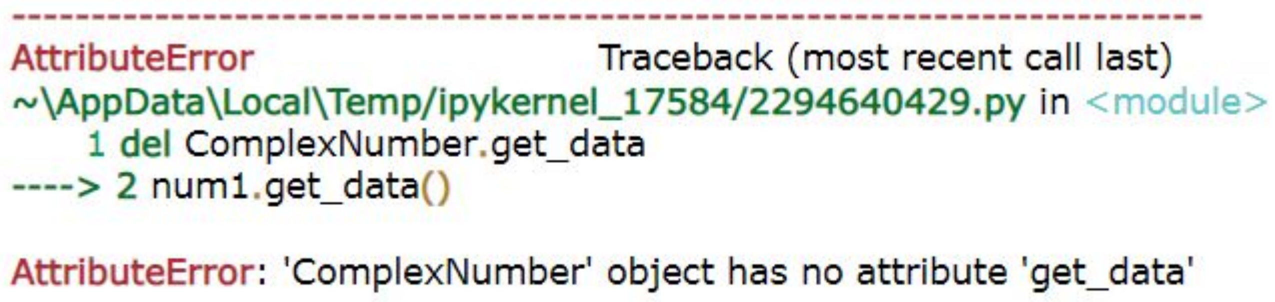
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_17584/2422605903.py in <module>
      3 
      4 del num1.imag
----> 5 num1.get_data()

~\AppData\Local\Temp/ipykernel_17584/3204845401.py in get_data(self)
      5 
      6     def get_data(self):
----> 7         print(f'{self.real}+{self.imag}j')
      8 
      9 num1 = ComplexNumber(2, 3)

AttributeError: 'ComplexNumber' object has no attribute 'imag'
```

```
del ComplexNumber.get_data
```

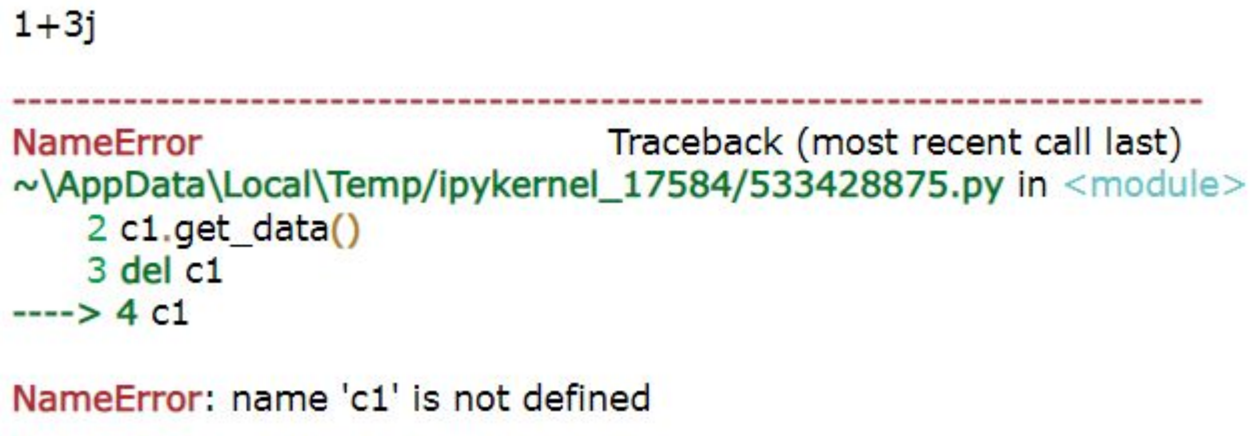
```
num1.get_data()
```



```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_17584/2294640429.py in <module>
      1 del ComplexNumber.get_data
----> 2 num1.get_data()

AttributeError: 'ComplexNumber' object has no attribute 'get_data'
```

```python
c1 = ComplexNumber(1,3)
c1.get_data()
del c1
c1
```



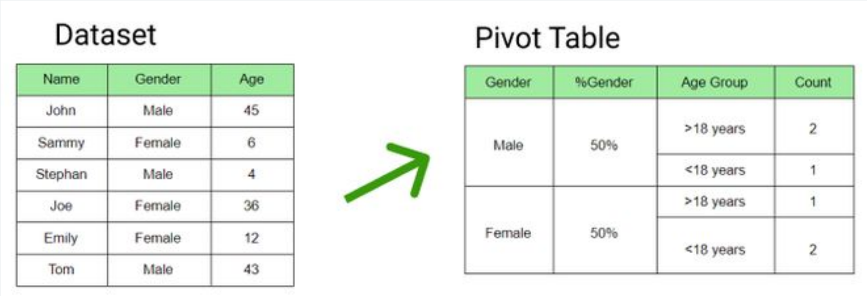
```
1+3j

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_17584/533428875.py in <module>
      2 c1.get_data()
      3 del c1
----> 4 c1

NameError: name 'c1' is not defined
```

Pivot_table

Function in Pandas



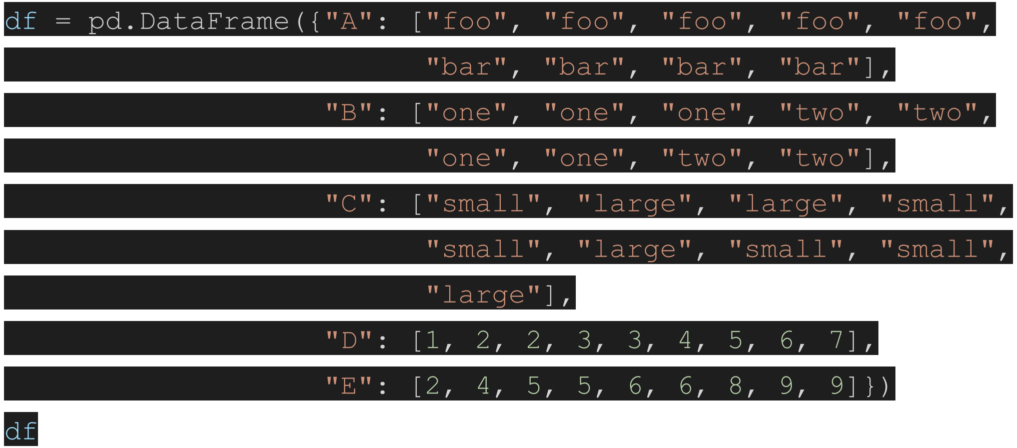
Dataset

Name	Gender	Age
John	Male	45
Sammy	Female	6
Stephan	Male	4
Joe	Female	36
Emily	Female	12
Tom	Male	43

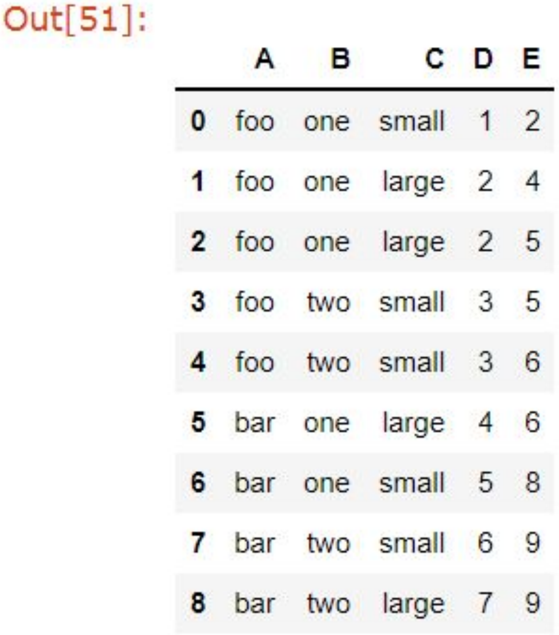
Pivot Table

Gender	%Gender	Age Group	Count
Male	50%	>18 years	2
		<18 years	1
Female	50%	>18 years	1
		<18 years	2

Presented by:
Incendero



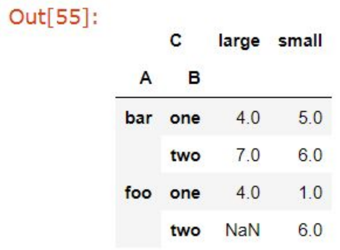
```python
df = pd.DataFrame({"A": ["foo", "foo", "foo", "foo", "foo",
                         "bar", "bar", "bar", "bar"],
                   "B": ["one", "one", "one", "two", "two",
                         "one", "one", "two", "two"],
                   "C": ["small", "large", "large", "small",
                         "small", "large", "small", "small",
                         "large"],
                   "D": [1, 2, 2, 3, 3, 4, 5, 6, 7],
                   "E": [2, 4, 5, 5, 6, 6, 8, 9, 9]})
df
```



Out[51]:

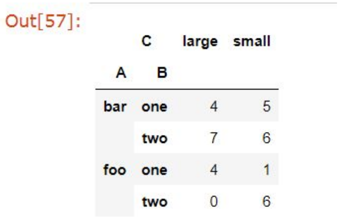
	A	B	C	D	E
0	foo	one	small	1	2
1	foo	one	large	2	4
2	foo	one	large	2	5
3	foo	two	small	3	5
4	foo	two	small	3	6
5	bar	one	large	4	6
6	bar	one	small	5	8
7	bar	two	small	6	9
8	bar	two	large	7	9

```
table = pd.pivot_table(df, values='D', index=['A',
'B'],
                    columns=['C'], aggfunc=np.sum)
table
```



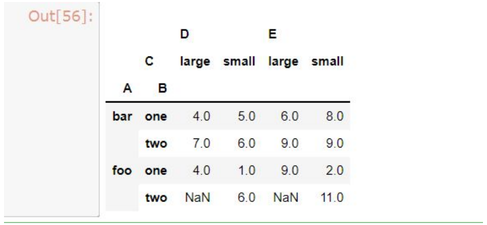
```
Out[55]:
         C   large  small
  A    B
bar  one      4.0    5.0
     two      7.0    6.0
foo  one      4.0    1.0
     two      NaN    6.0
```

To make Nan as zero, add parameter fill_value = 0

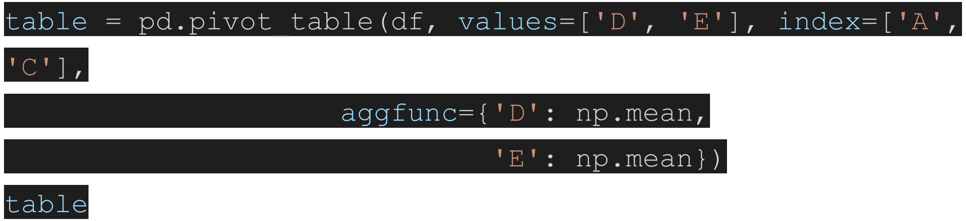


```
Out[57]:
        C  large  small
  A    B
bar  one      4      5
     two      7      6
foo  one      4      1
     two      0      6
```

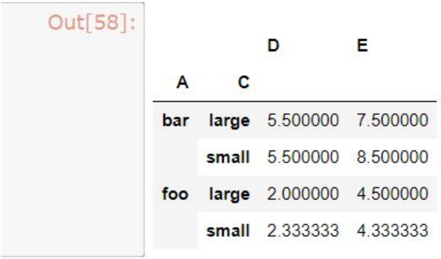
```
table = pd.pivot_table(df, values=['D','E'], index=['A',
'B'],
                    columns=['C'], aggfunc=np.sum)
table
```



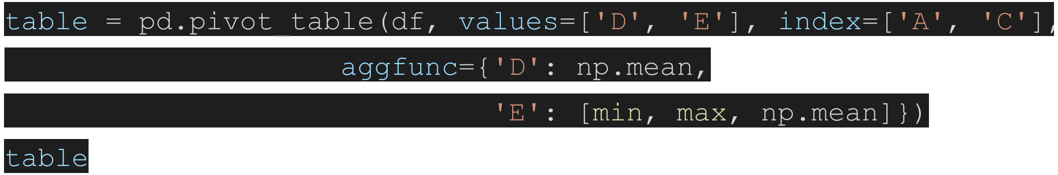
```
Out[56]:
              D             E
        C  large  small  large  small
  A    B
bar  one    4.0    5.0    6.0    8.0
     two    7.0    6.0    9.0    9.0
foo  one    4.0    1.0    9.0    2.0
     two    NaN    6.0    NaN   11.0
```



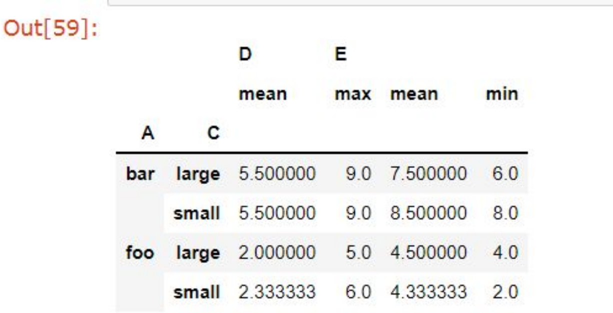
```
table = pd.pivot_table(df, values=['D', 'E'], index=['A',
'C'],
                    aggfunc={'D': np.mean,
                             'E': np.mean})
table
```



```
Out[58]:
                    D          E
 A      C
bar   large   5.500000   7.500000
      small   5.500000   8.500000
foo   large   2.000000   4.500000
      small   2.333333   4.333333
```



```
table = pd.pivot_table(df, values=['D', 'E'], index=['A', 'C'],
                    aggfunc={'D': np.mean,
                             'E': [min, max, np.mean]})
table
```



```
Out[59]:
                    D          E
                  mean       max   mean       min
 A      C
bar   large   5.500000   9.0   7.500000   6.0
      small   5.500000   9.0   8.500000   8.0
foo   large   2.000000   5.0   4.500000   4.0
      small   2.333333   6.0   4.333333   2.0
```

THANK YOU