Python Control Flow

Jennifer Helsby, Eric Potash
Computation for Public Policy
Lecture 3: January 12, 2016
`computationforpolicy.github.io`

Announcements

- Assignment #1 is now due on **Friday by 5pm**
- Bill is having office hours **Thursday 3-5pm**! Hands-on help
- Alternative VM if you have an older machine is available
- Functions pushed to Thursday

Today

- Python control flow
- Example analysis in IPython notebook (will help a lot for HW!)

Notebooks for today

- Part 1: Control flow

https://is.gd/gXQV2j

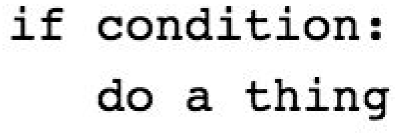
- Part 2: Example data analysis

https://is.gd/jqtils

Pseudocode

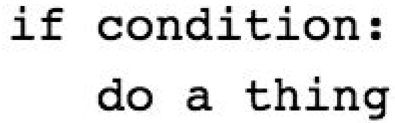
- Informal way of writing down what is to be done in your program
- Excellent first step when programming
- Formalize at later point

if statements

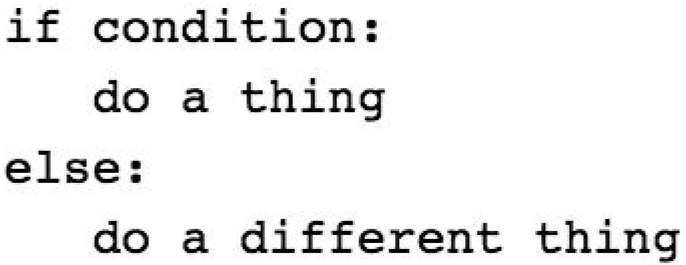


```
if condition:
   do a thing
```

if statements

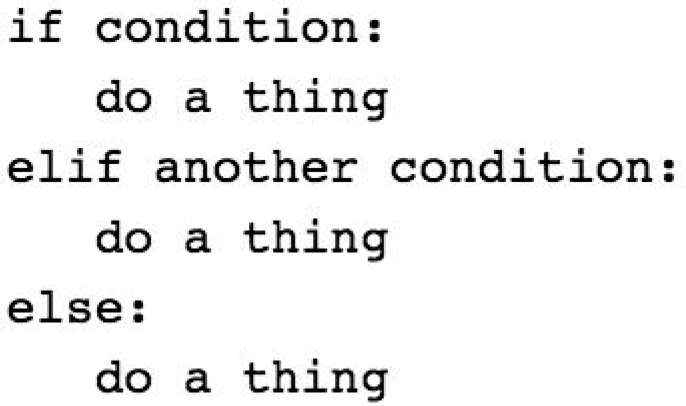


```
if condition:
    do a thing
```



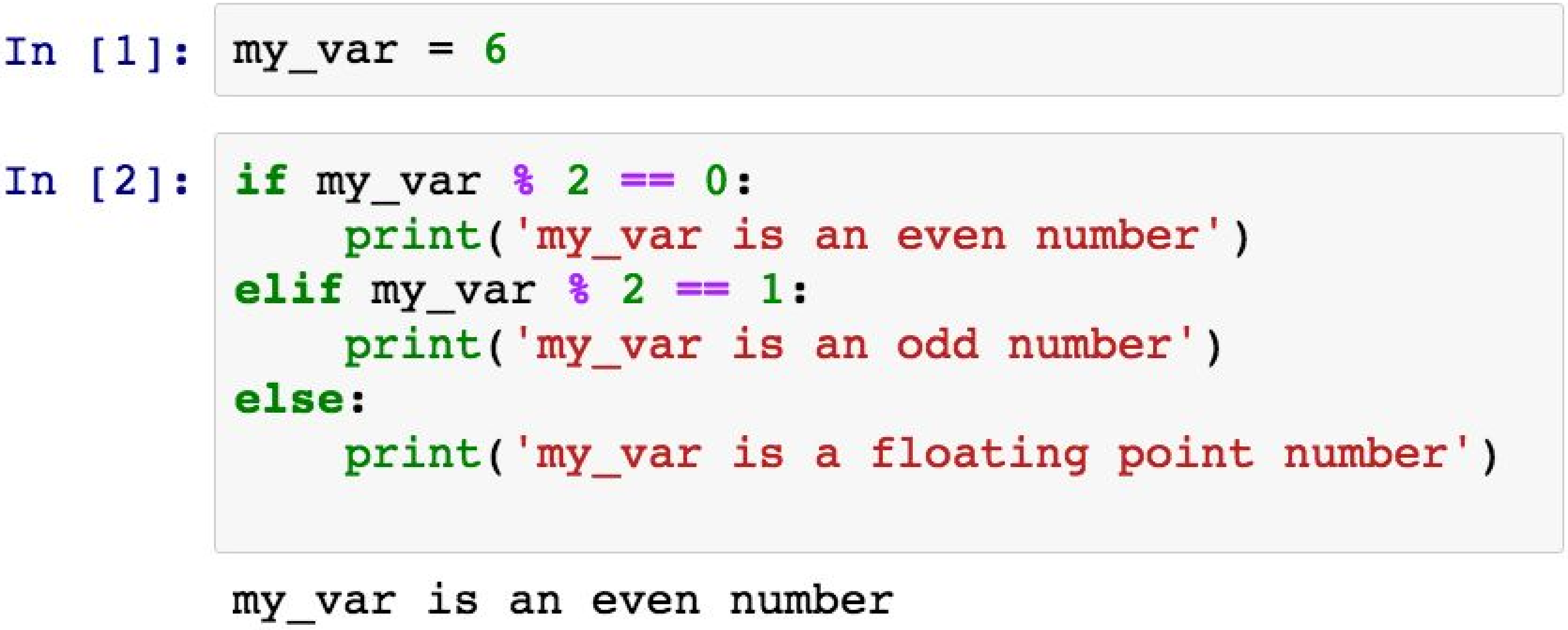
```
if condition:
    do a thing
else:
    do a different thing
```

if statements



```
if condition:
   do a thing
elif another condition:
   do a thing
else:
   do a thing
```

if statements: print if my_var is even or odd

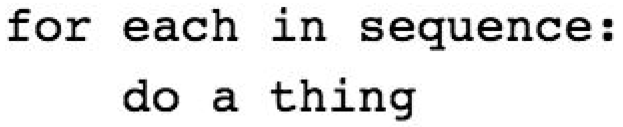


```
In [1]: my_var = 6
```

```
In [2]: if my_var % 2 == 0:
            print('my_var is an even number')
        elif my_var % 2 == 1:
            print('my_var is an odd number')
        else:
            print('my_var is a floating point number')
```

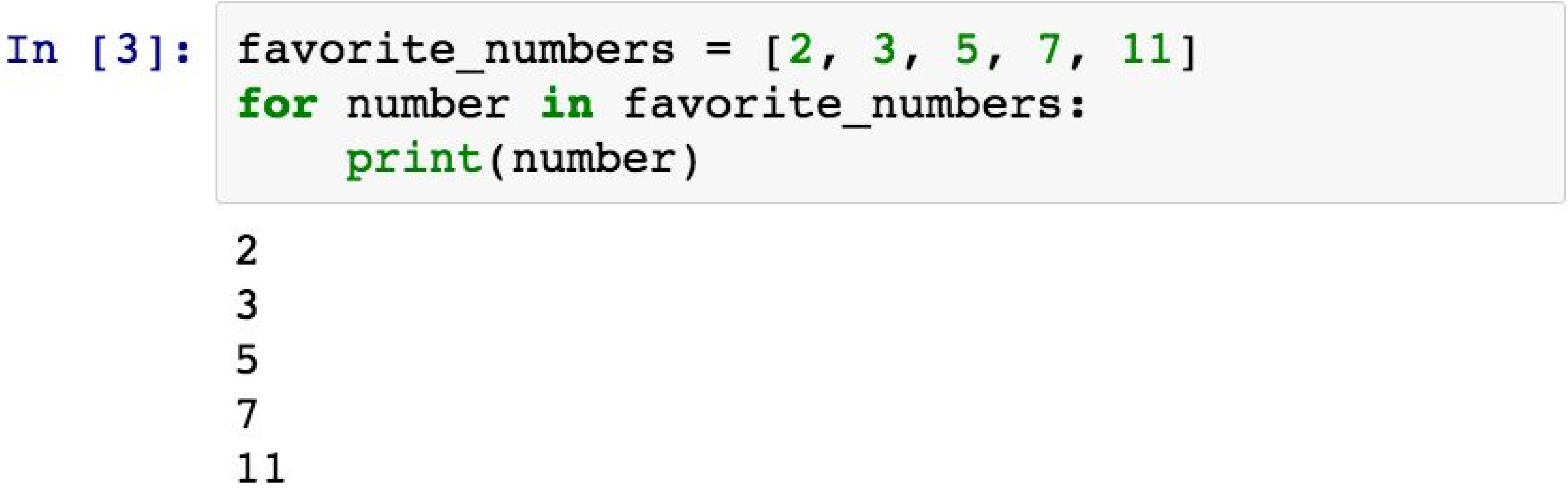
```
my_var is an even number
```

for loops



```
for each in sequence:
    do a thing
```

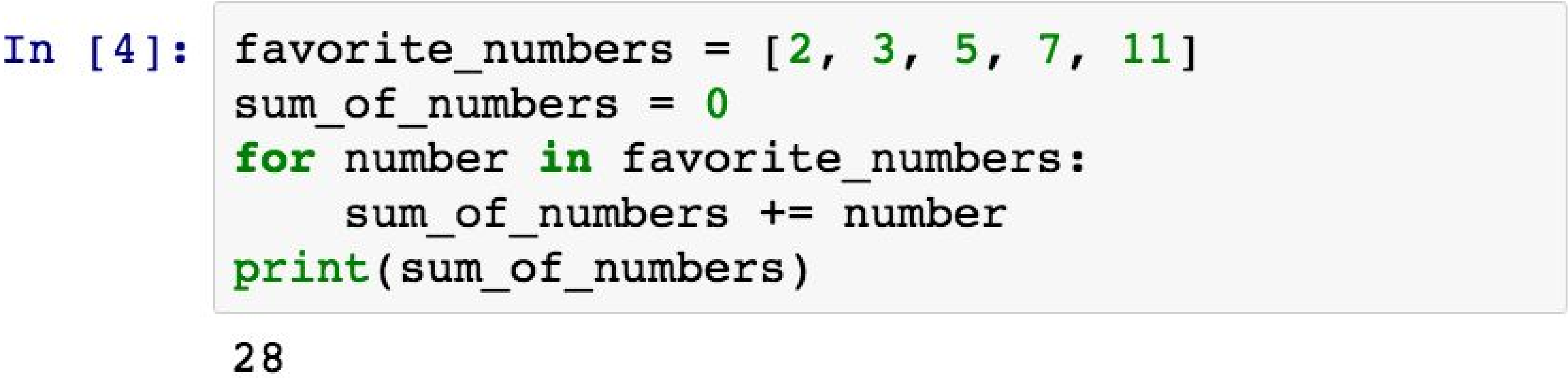
for loops



```python
favorite_numbers = [2, 3, 5, 7, 11]
for number in favorite_numbers:
    print(number)
```

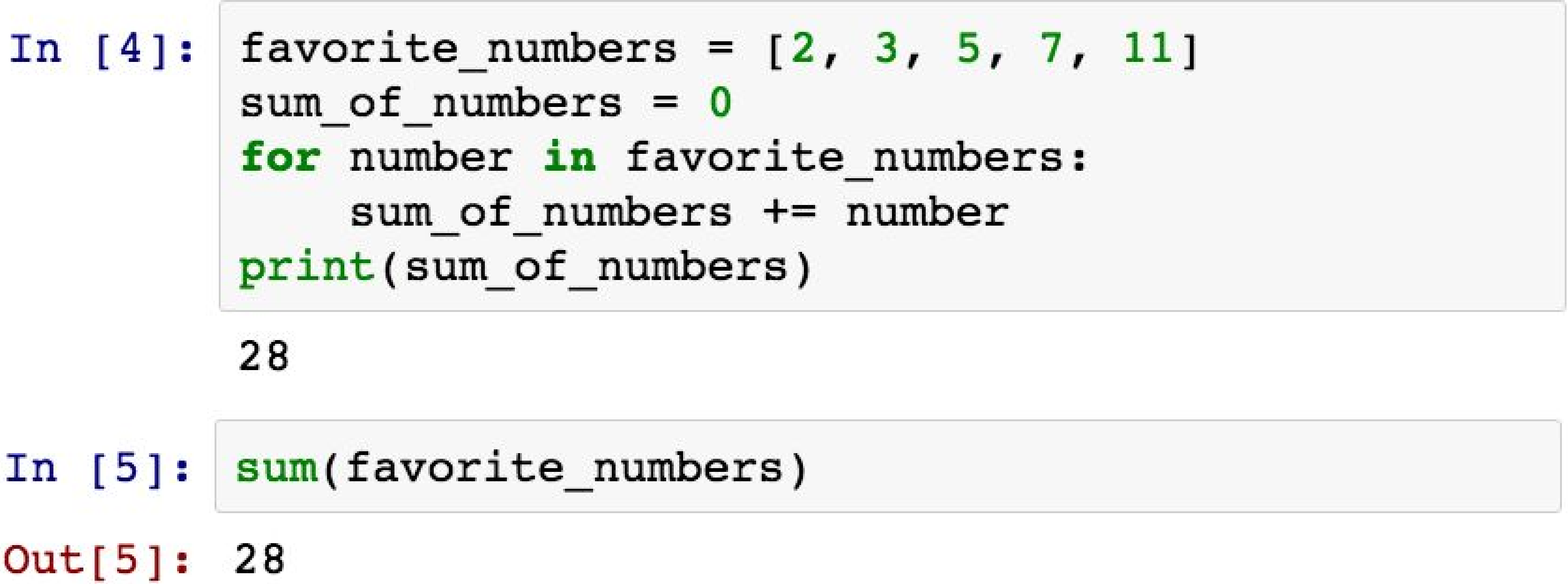
```
2
3
5
7
11
```

for loops



```
In [4]: favorite_numbers = [2, 3, 5, 7, 11]
        sum_of_numbers = 0
        for number in favorite_numbers:
            sum_of_numbers += number
        print(sum_of_numbers)

        28
```

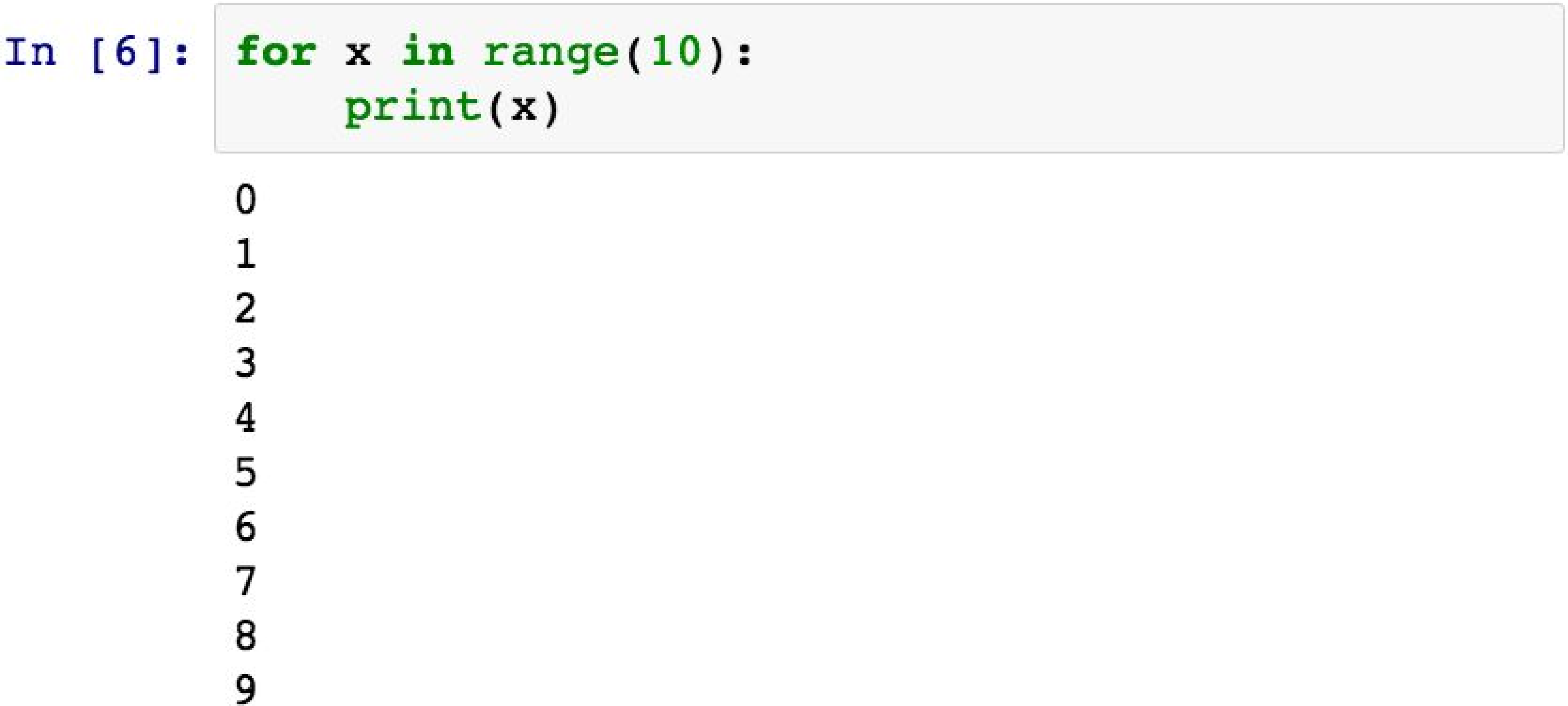
for loops



```
In [4]: favorite_numbers = [2, 3, 5, 7, 11]
        sum_of_numbers = 0
        for number in favorite_numbers:
            sum_of_numbers += number
        print(sum_of_numbers)

        28

In [5]: sum(favorite_numbers)

Out[5]: 28
```

range()

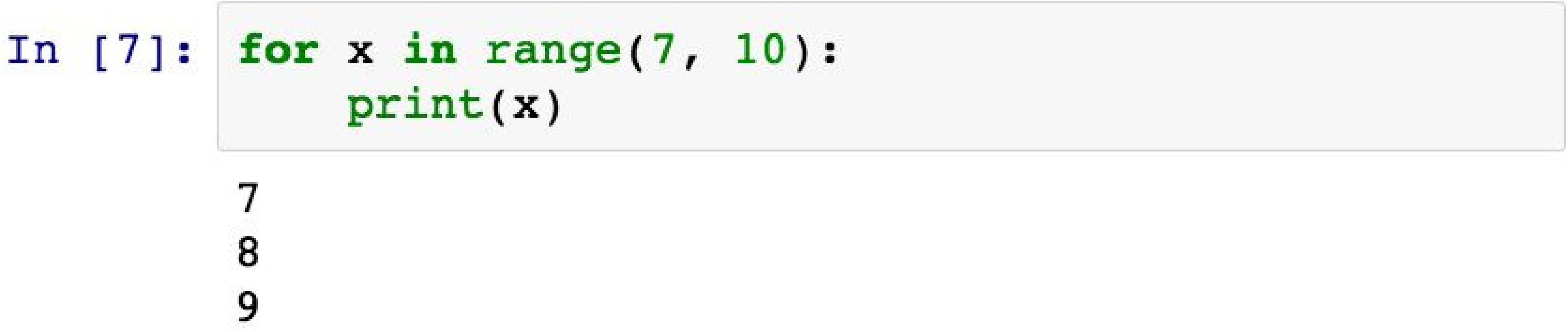
Range starts at 0



```
In [6]: for x in range(10):
            print(x)

        0
        1
        2
        3
        4
        5
        6
        7
        8
        9
```

range()

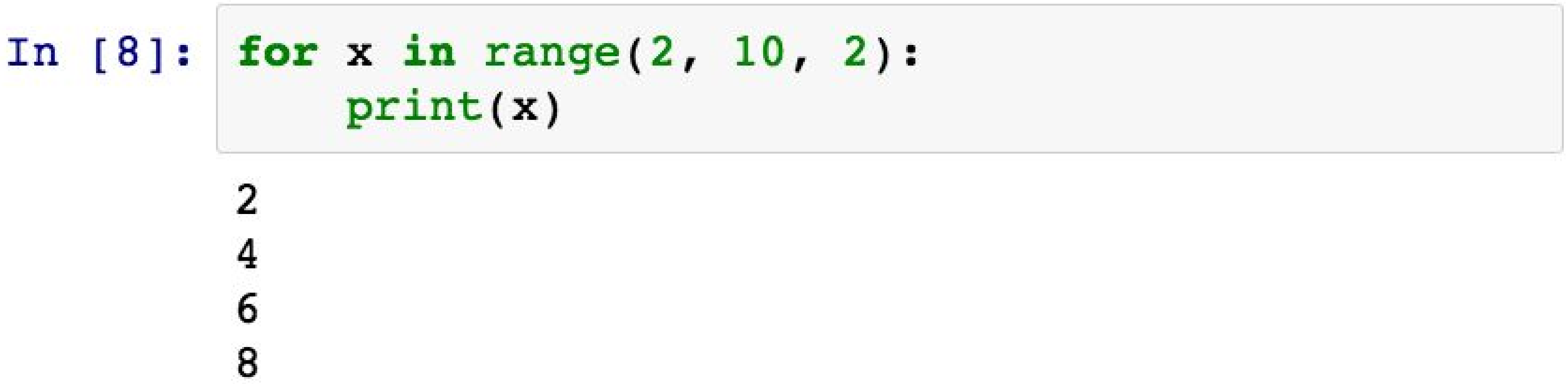
Range starts at 0, but can also start at a value passed to it



```
In [7]: for x in range(7, 10):
            print(x)

        7
        8
        9
```

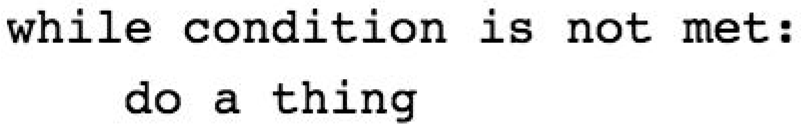
range()

The step can be adjusted, here we step by 2:



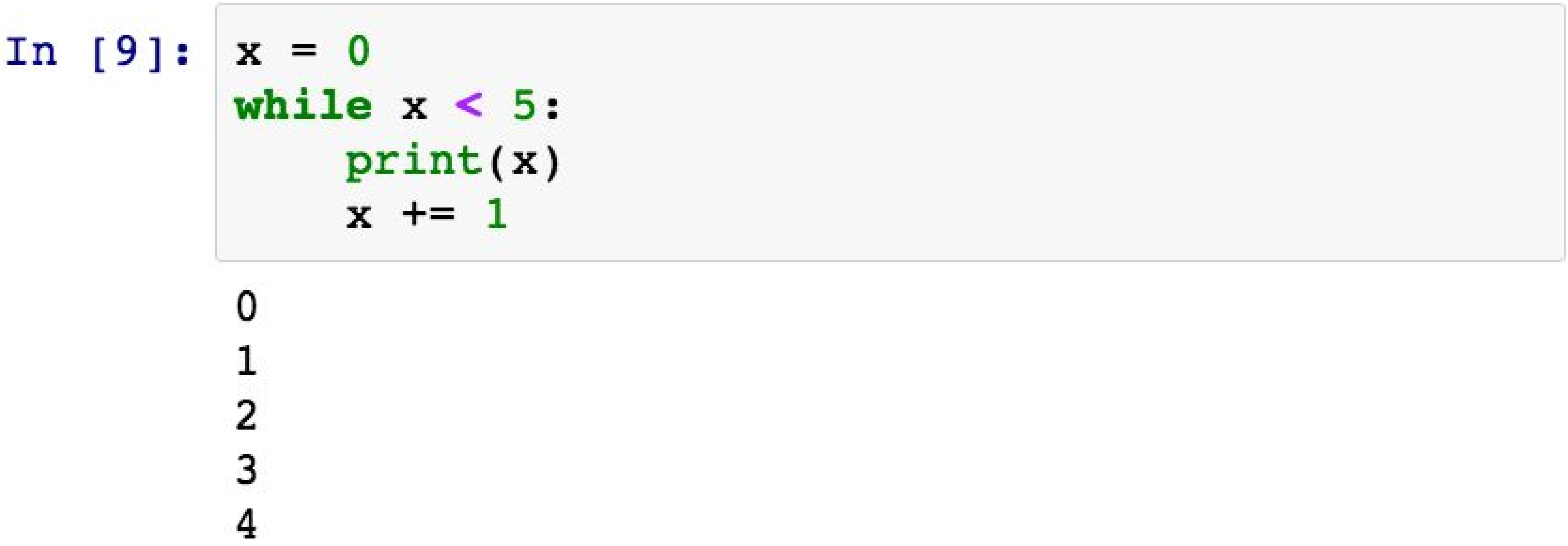
```
In [8]: for x in range(2, 10, 2):
            print(x)

        2
        4
        6
        8
```

while loops



```
while condition is not met:
    do a thing
```

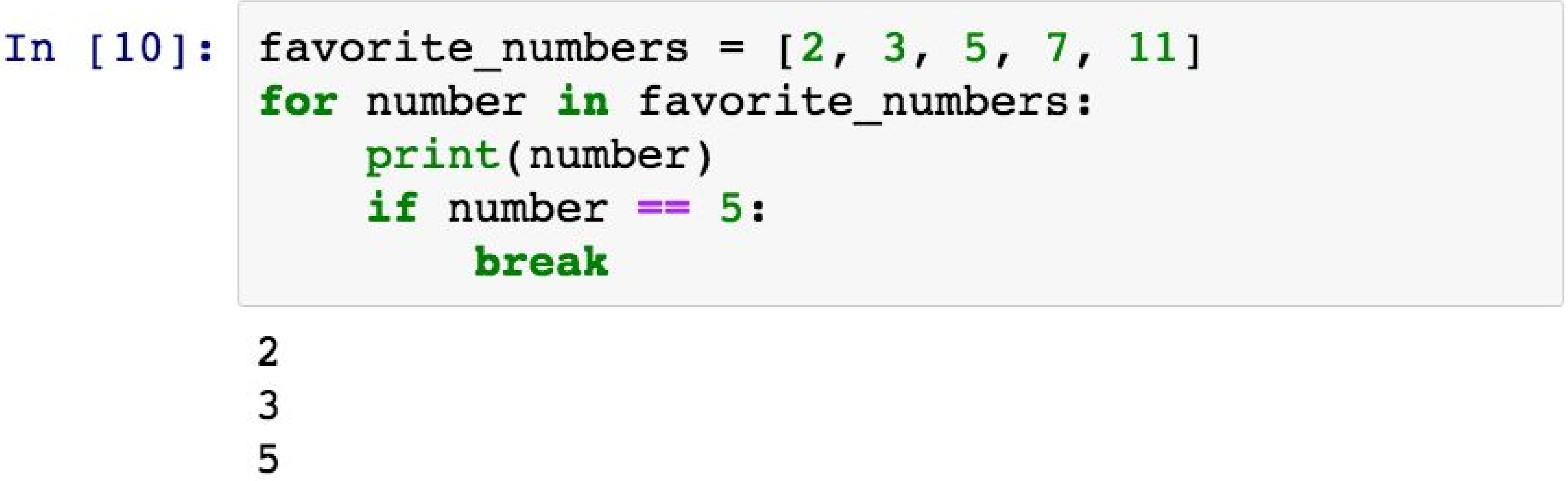
while loops



```
x = 0
while x < 5:
    print(x)
    x += 1
```

```
0
1
2
3
4
```

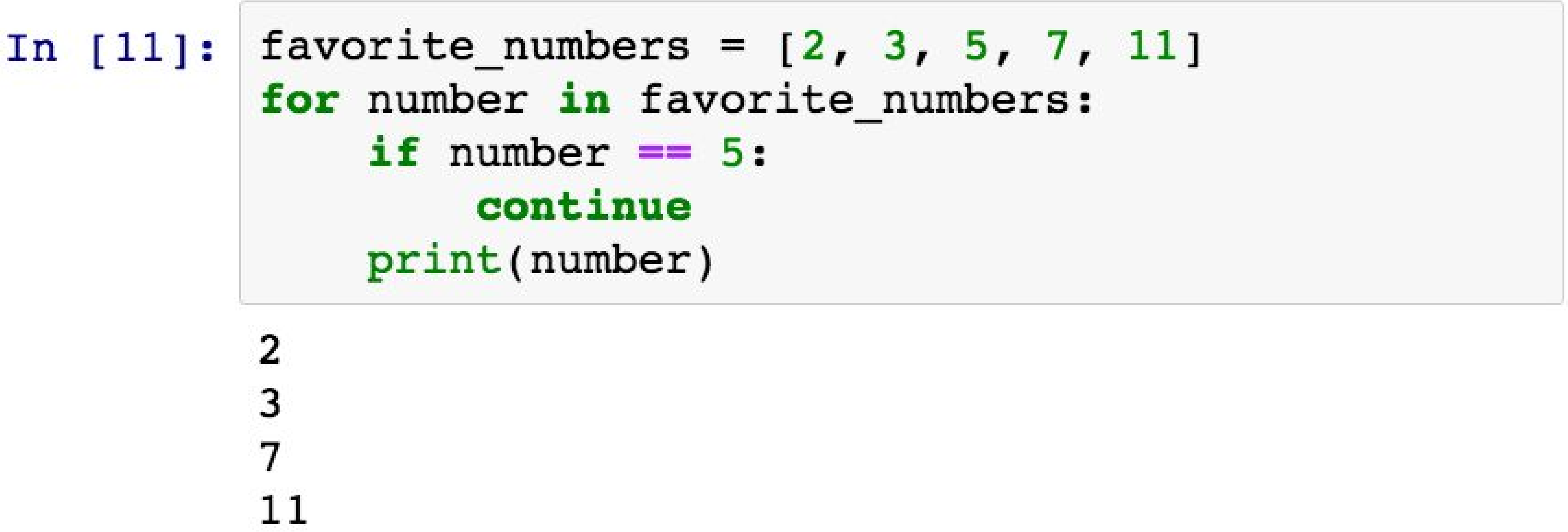
break



```
In [10]: favorite_numbers = [2, 3, 5, 7, 11]
         for number in favorite_numbers:
             print(number)
             if number == 5:
                 break
```

```
2
3
5
```

continue

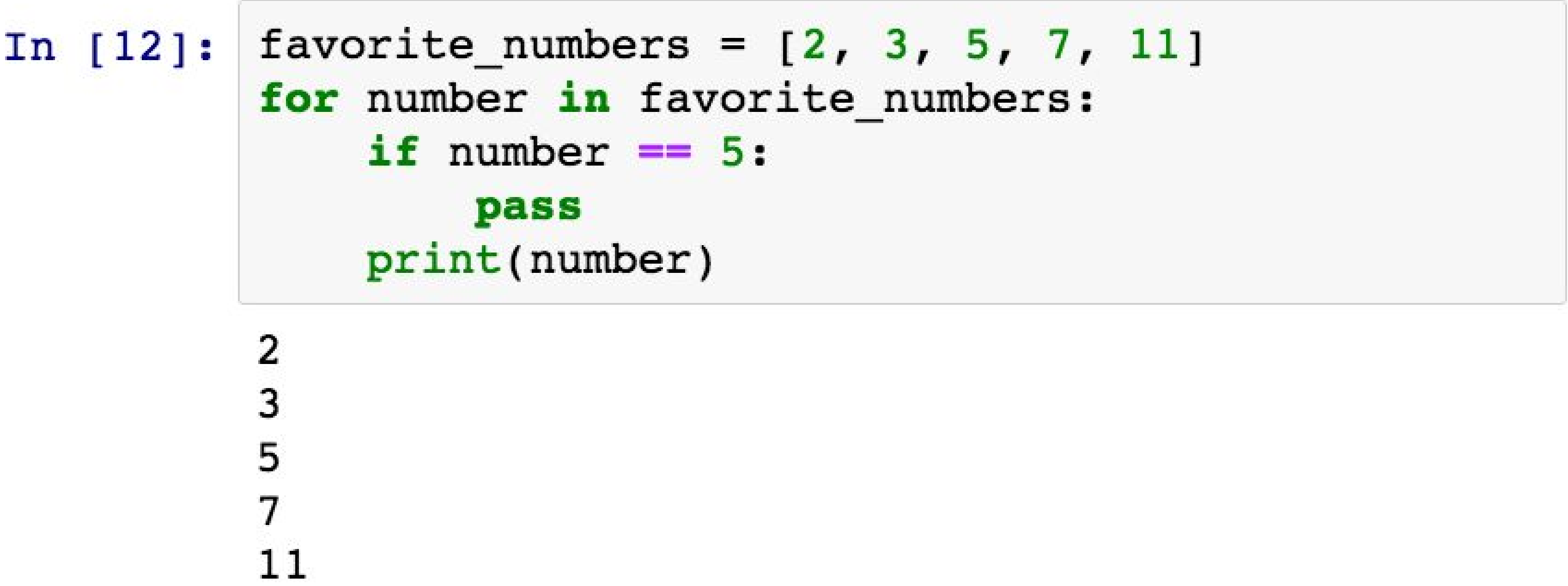


In [11]:

```python
favorite_numbers = [2, 3, 5, 7, 11]
for number in favorite_numbers:
    if number == 5:
        continue
    print(number)
```

```
2
3
7
11
```

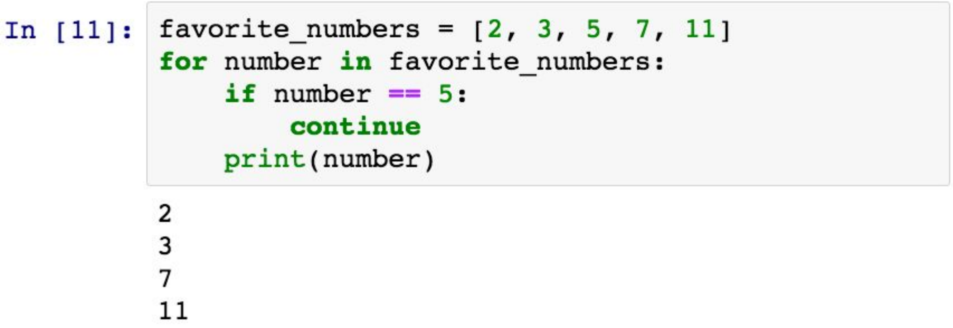
pass



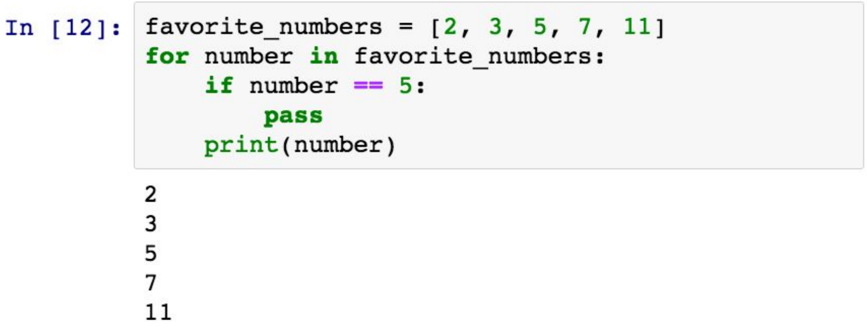
```
In [12]: favorite_numbers = [2, 3, 5, 7, 11]
         for number in favorite_numbers:
             if number == 5:
                 pass
             print(number)
```

```
2
3
5
7
11
```

continue vs. pass



```
In [11]: favorite_numbers = [2, 3, 5, 7, 11]
         for number in favorite_numbers:
             if number == 5:
                 continue
             print(number)

         2
         3
         7
         11
```



```
In [12]: favorite_numbers = [2, 3, 5, 7, 11]
         for number in favorite_numbers:
             if number == 5:
                 pass
             print(number)

         2
         3
         5
         7
         11
```

Basic Data Analysis

https://is.gd/jqtils

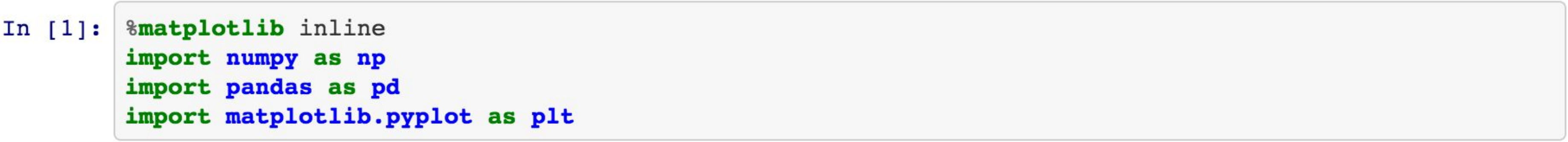
Getting Started

- Download the data file from: https://is.gd/O0YGLc
- Start the IPython notebook:

```
$ ipython notebook
```

- Navigate to and open the IPython notebook example you downloaded from: https://is.gd/jqtils
- Follow along!

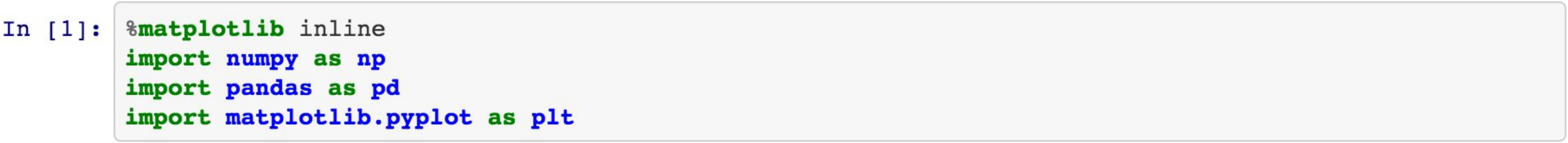
Imports



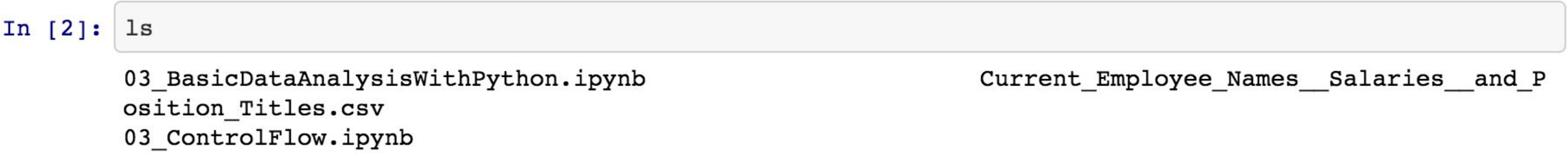
In [1]:

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Imports

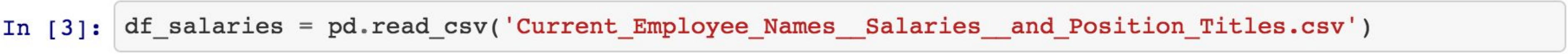


```
In [1]: %matplotlib inline
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```



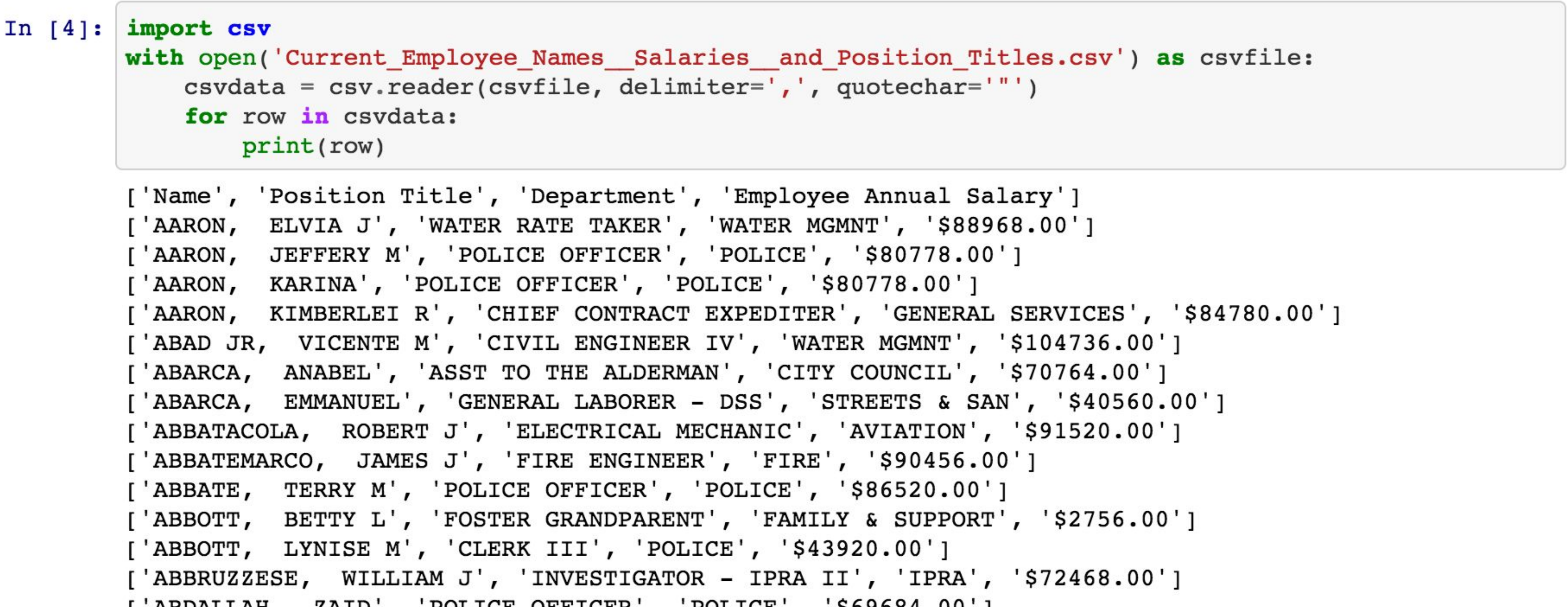
```
In [2]: ls

        03_BasicDataAnalysisWithPython.ipynb                          Current_Employee_Names__Salaries__and_P
        osition_Titles.csv
        03_ControlFlow.ipynb
```

Read in Data



```
In [3]: df_salaries = pd.read_csv('Current_Employee_Names__Salaries__and_Position_Titles.csv')
```

Read in Data

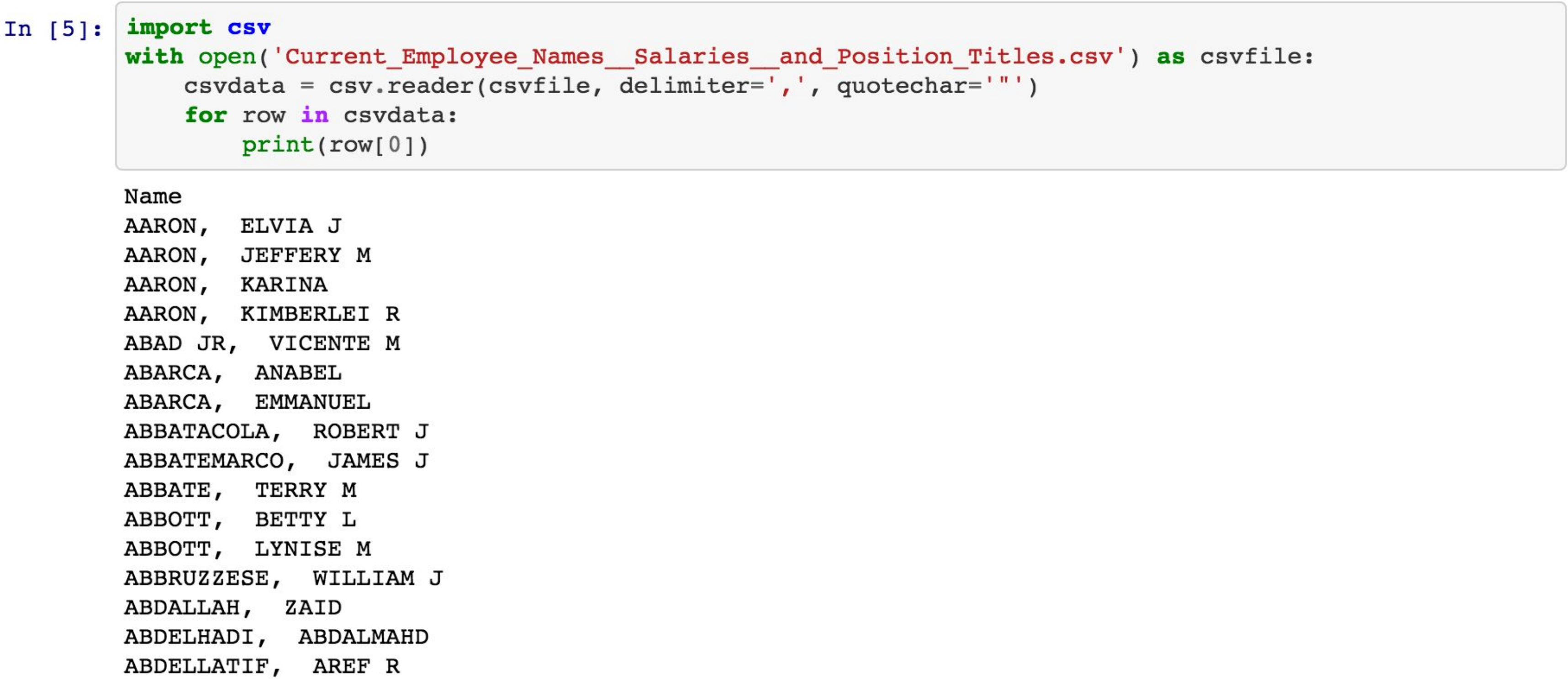


In [4]:

```
import csv
with open('Current_Employee_Names__Salaries__and_Position_Titles.csv') as csvfile:
    csvdata = csv.reader(csvfile, delimiter=',', quotechar='"')
    for row in csvdata:
        print(row)
```

```
['Name', 'Position Title', 'Department', 'Employee Annual Salary']
['AARON,  ELVIA J', 'WATER RATE TAKER', 'WATER MGMNT', '$88968.00']
['AARON,  JEFFERY M', 'POLICE OFFICER', 'POLICE', '$80778.00']
['AARON,  KARINA', 'POLICE OFFICER', 'POLICE', '$80778.00']
['AARON,  KIMBERLEI R', 'CHIEF CONTRACT EXPEDITER', 'GENERAL SERVICES', '$84780.00']
['ABAD JR,  VICENTE M', 'CIVIL ENGINEER IV', 'WATER MGMNT', '$104736.00']
['ABARCA,  ANABEL', 'ASST TO THE ALDERMAN', 'CITY COUNCIL', '$70764.00']
['ABARCA,  EMMANUEL', 'GENERAL LABORER - DSS', 'STREETS & SAN', '$40560.00']
['ABBATACOLA,  ROBERT J', 'ELECTRICAL MECHANIC', 'AVIATION', '$91520.00']
['ABBATEMARCO,  JAMES J', 'FIRE ENGINEER', 'FIRE', '$90456.00']
['ABBATE,  TERRY M', 'POLICE OFFICER', 'POLICE', '$86520.00']
['ABBOTT,  BETTY L', 'FOSTER GRANDPARENT', 'FAMILY & SUPPORT', '$2756.00']
['ABBOTT,  LYNISE M', 'CLERK III', 'POLICE', '$43920.00']
['ABBRUZZESE,  WILLIAM J', 'INVESTIGATOR - IPRA II', 'IPRA', '$72468.00']
['ABDALLAH,  ZAID', 'POLICE OFFICER', 'POLICE', '$69684.00']
```

Read in Only Names



```
In [5]: import csv
        with open('Current_Employee_Names__Salaries__and_Position_Titles.csv') as csvfile:
            csvdata = csv.reader(csvfile, delimiter=',', quotechar='"')
            for row in csvdata:
                print(row[0])
```

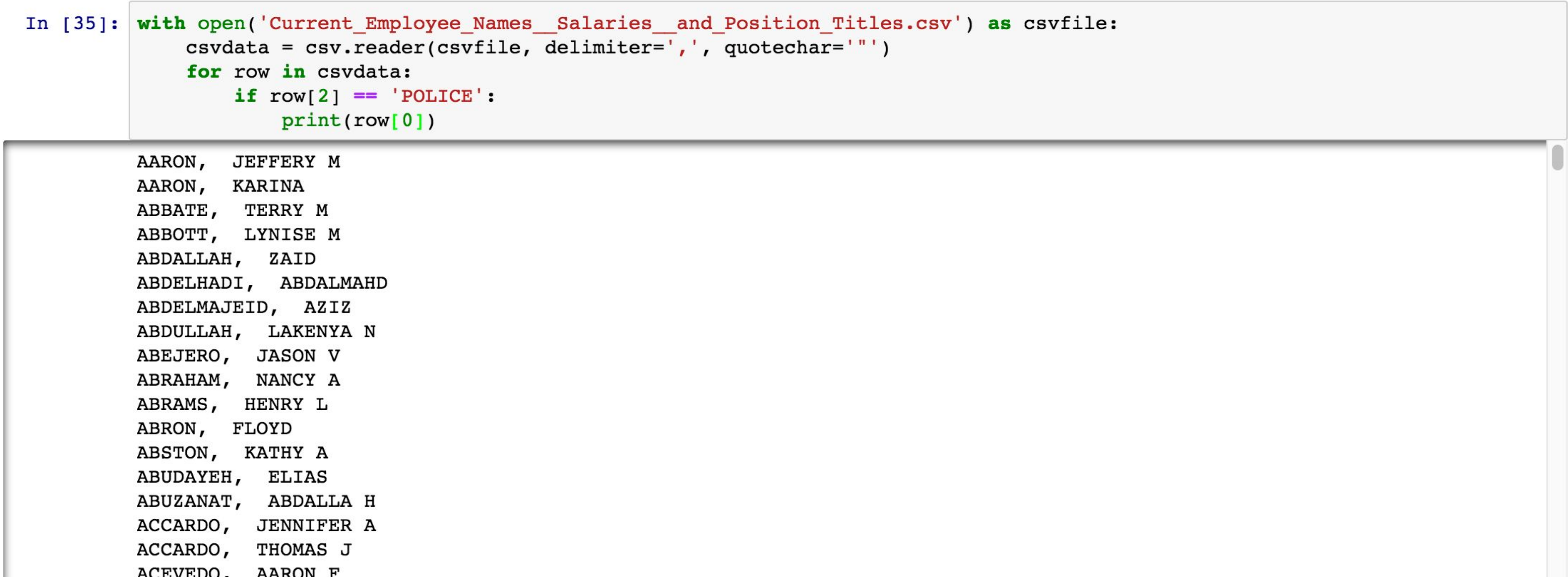
```
Name
AARON,  ELVIA J
AARON,  JEFFERY M
AARON,  KARINA
AARON,  KIMBERLEI R
ABAD JR,  VICENTE M
ABARCA,  ANABEL
ABARCA,  EMMANUEL
ABBATACOLA,  ROBERT J
ABBATEMARCO,  JAMES J
ABBATE,  TERRY M
ABBOTT,  BETTY L
ABBOTT,  LYNISE M
ABBRUZZESE,  WILLIAM J
ABDALLAH,  ZAID
ABDELHADI,  ABDALMAHD
ABDELLATIF,  AREF R
```

Class Challenge 1

Print only names of those rows where the Department column is ‘POLICE’

Class Challenge 1

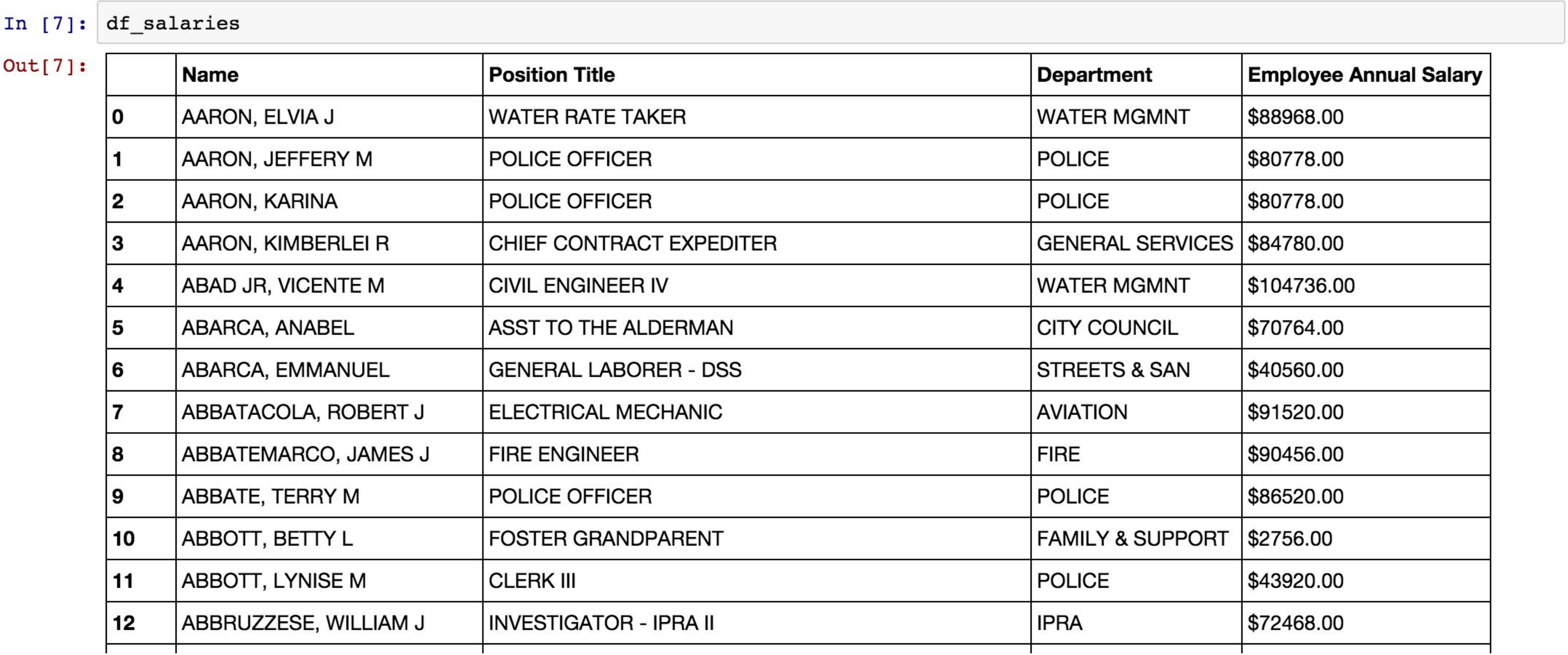
Print only names of those rows where the Department column is ‘POLICE’



```
In [35]: with open('Current_Employee_Names__Salaries__and_Position_Titles.csv') as csvfile:
             csvdata = csv.reader(csvfile, delimiter=',', quotechar='"')
             for row in csvdata:
                 if row[2] == 'POLICE':
                     print(row[0])
```

```
AARON,  JEFFERY M
AARON,  KARINA
ABBATE,  TERRY M
ABBOTT,  LYNISE M
ABDALLAH,  ZAID
ABDELHADI,  ABDALMAHD
ABDELMAJEID,  AZIZ
ABDULLAH,  LAKENYA N
ABEJERO,  JASON V
ABRAHAM,  NANCY A
ABRAMS,  HENRY L
ABRON,  FLOYD
ABSTON,  KATHY A
ABUDAYEH,  ELIAS
ABUZANAT,  ABDALLA H
ACCARDO,  JENNIFER A
ACCARDO,  THOMAS J
ACEVEDO,  AARON F
```

Pandas DataFrames

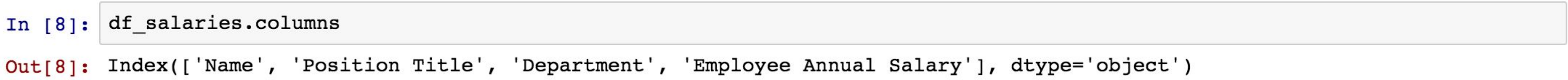


```
In [7]: df_salaries
```

Out[7]:

	Name	Position Title	Department	Employee Annual Salary
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00
5	ABARCA, ANABEL	ASST TO THE ALDERMAN	CITY COUNCIL	$70764.00
6	ABARCA, EMMANUEL	GENERAL LABORER - DSS	STREETS & SAN	$40560.00
7	ABBATACOLA, ROBERT J	ELECTRICAL MECHANIC	AVIATION	$91520.00
8	ABBATEMARCO, JAMES J	FIRE ENGINEER	FIRE	$90456.00
9	ABBATE, TERRY M	POLICE OFFICER	POLICE	$86520.00
10	ABBOTT, BETTY L	FOSTER GRANDPARENT	FAMILY & SUPPORT	$2756.00
11	ABBOTT, LYNISE M	CLERK III	POLICE	$43920.00
12	ABBRUZZESE, WILLIAM J	INVESTIGATOR - IPRA II	IPRA	$72468.00

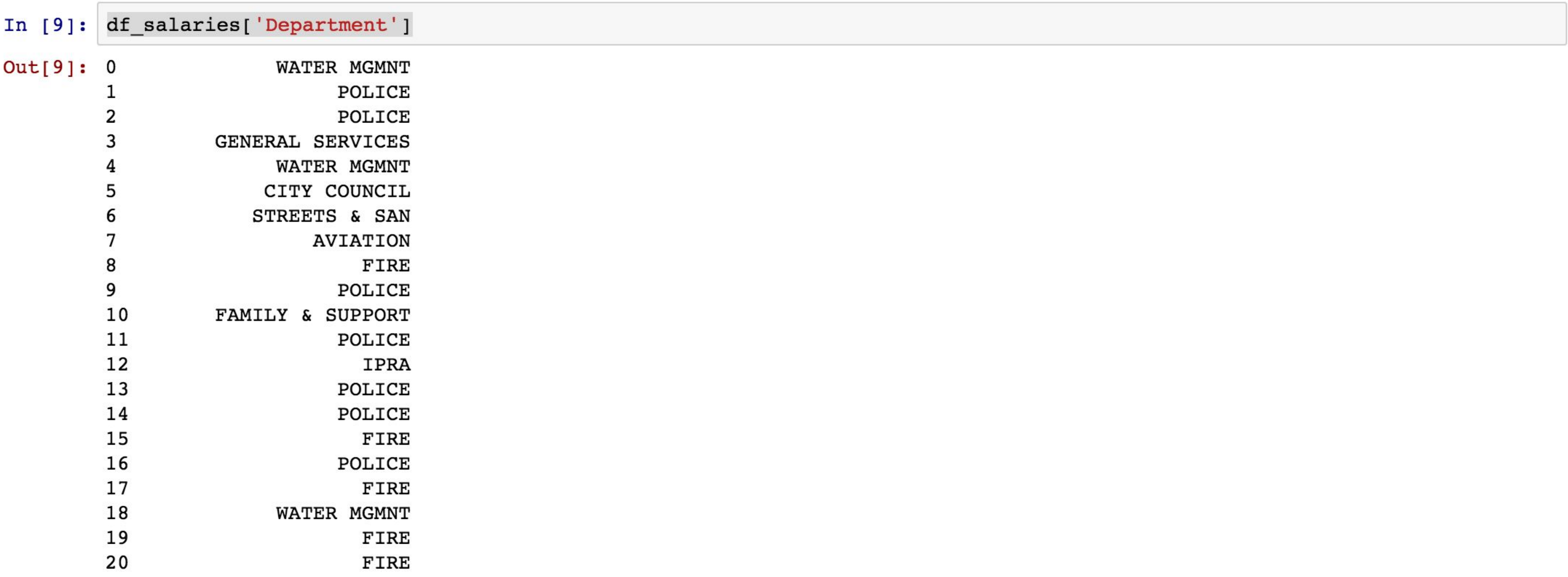
Pandas DataFrames



```
In [8]: df_salaries.columns

Out[8]: Index(['Name', 'Position Title', 'Department', 'Employee Annual Salary'], dtype='object')
```

Pandas DataFrames

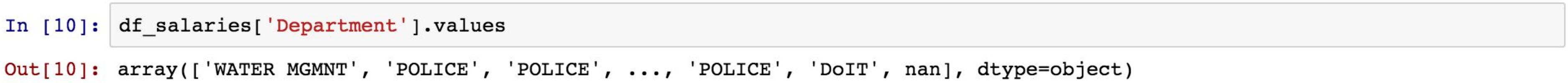
Access column with square brackets



```
In [9]: df_salaries['Department']

Out[9]: 0              WATER MGMNT
        1                   POLICE
        2                   POLICE
        3         GENERAL SERVICES
        4              WATER MGMNT
        5             CITY COUNCIL
        6           STREETS & SAN
        7                 AVIATION
        8                     FIRE
        9                   POLICE
        10        FAMILY & SUPPORT
        11                  POLICE
        12                    IPRA
        13                  POLICE
        14                  POLICE
        15                    FIRE
        16                  POLICE
        17                    FIRE
        18             WATER MGMNT
        19                    FIRE
        20                    FIRE
```

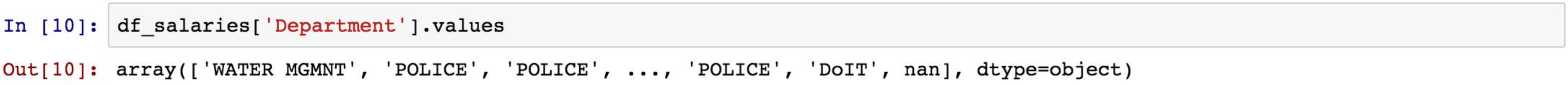
Pandas DataFrames

Get the values as numpy array with `.values`



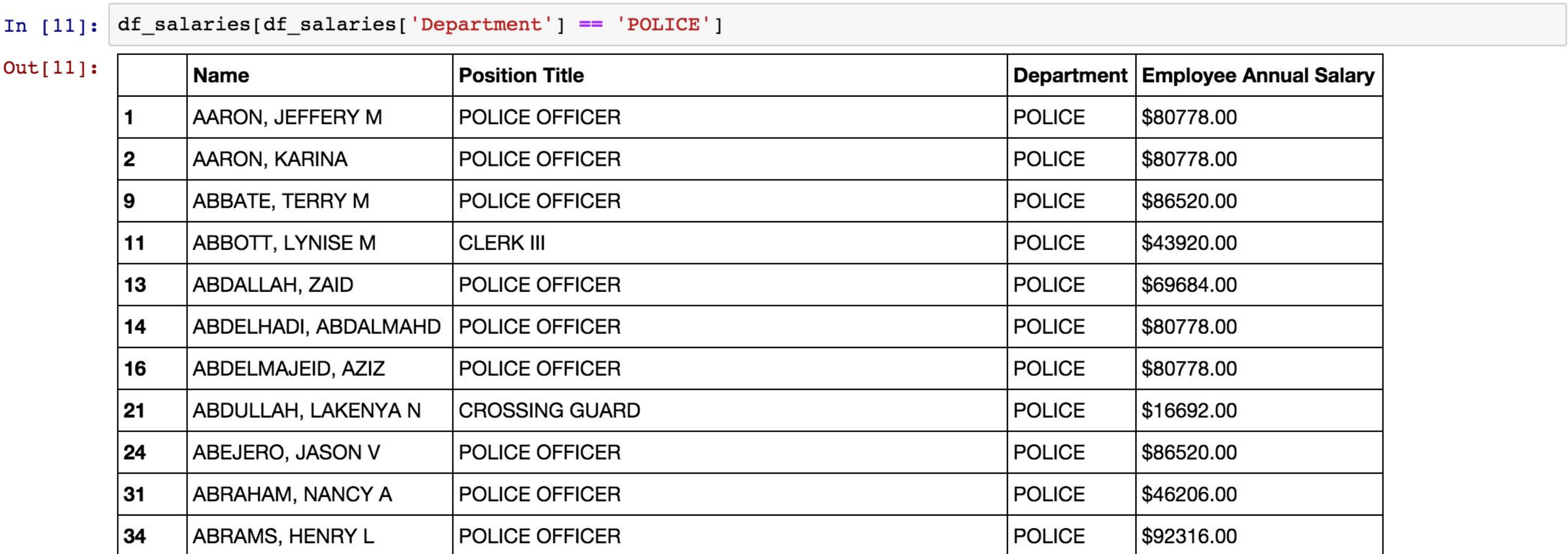
```
In [10]: df_salaries['Department'].values

Out[10]: array(['WATER MGMNT', 'POLICE', 'POLICE', ..., 'POLICE', 'DoIT', nan], dtype=object)
```

Pandas DataFrames



```
In [10]: df_salaries['Department'].values

Out[10]: array(['WATER MGMNT', 'POLICE', 'POLICE', ..., 'POLICE', 'DoIT', nan], dtype=object)
```

Pandas: Select rows where Department=='POLICE'



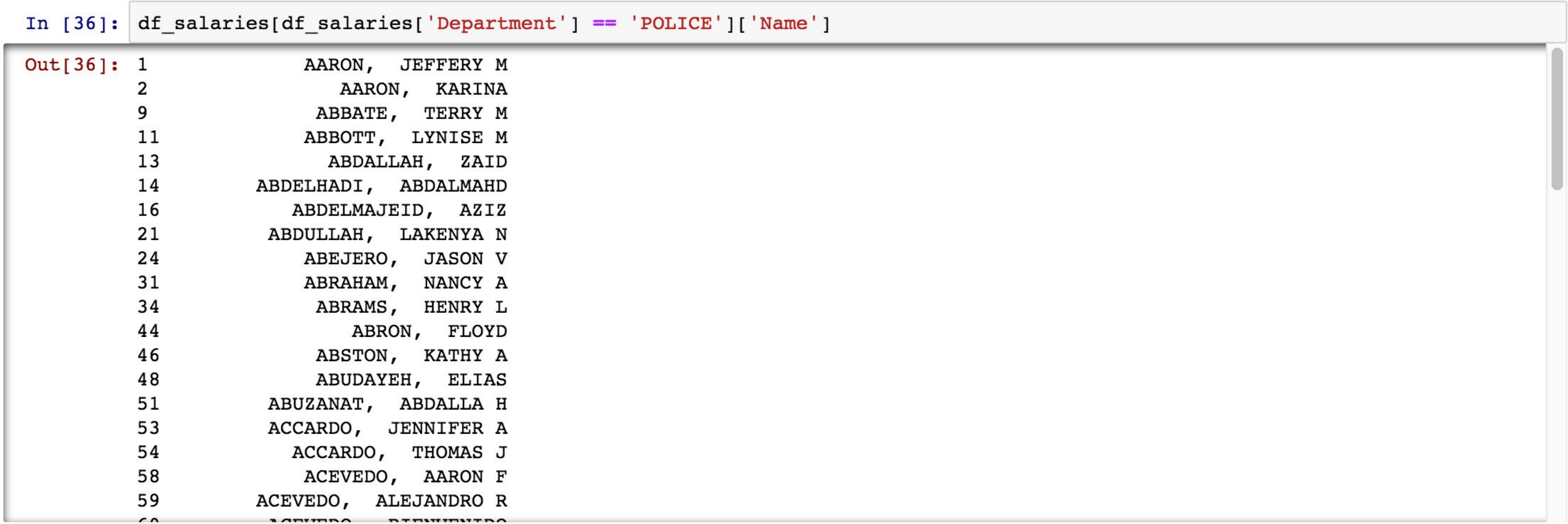
In [11]:

```
df_salaries[df_salaries['Department'] == 'POLICE']
```

Out[11]:

	Name	Position Title	Department	Employee Annual Salary
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00
9	ABBATE, TERRY M	POLICE OFFICER	POLICE	$86520.00
11	ABBOTT, LYNISE M	CLERK III	POLICE	$43920.00
13	ABDALLAH, ZAID	POLICE OFFICER	POLICE	$69684.00
14	ABDELHADI, ABDALMAHD	POLICE OFFICER	POLICE	$80778.00
16	ABDELMAJEID, AZIZ	POLICE OFFICER	POLICE	$80778.00
21	ABDULLAH, LAKENYA N	CROSSING GUARD	POLICE	$16692.00
24	ABEJERO, JASON V	POLICE OFFICER	POLICE	$86520.00
31	ABRAHAM, NANCY A	POLICE OFFICER	POLICE	$46206.00
34	ABRAMS, HENRY L	POLICE OFFICER	POLICE	$92316.00

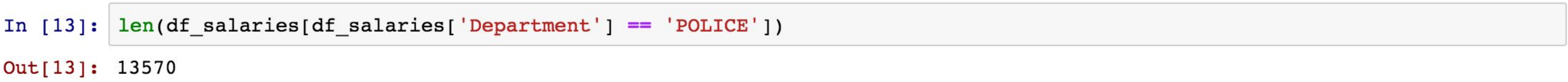
Pandas: Select names where Department=='POLICE'



```
In [36]: df_salaries[df_salaries['Department'] == 'POLICE']['Name']
```

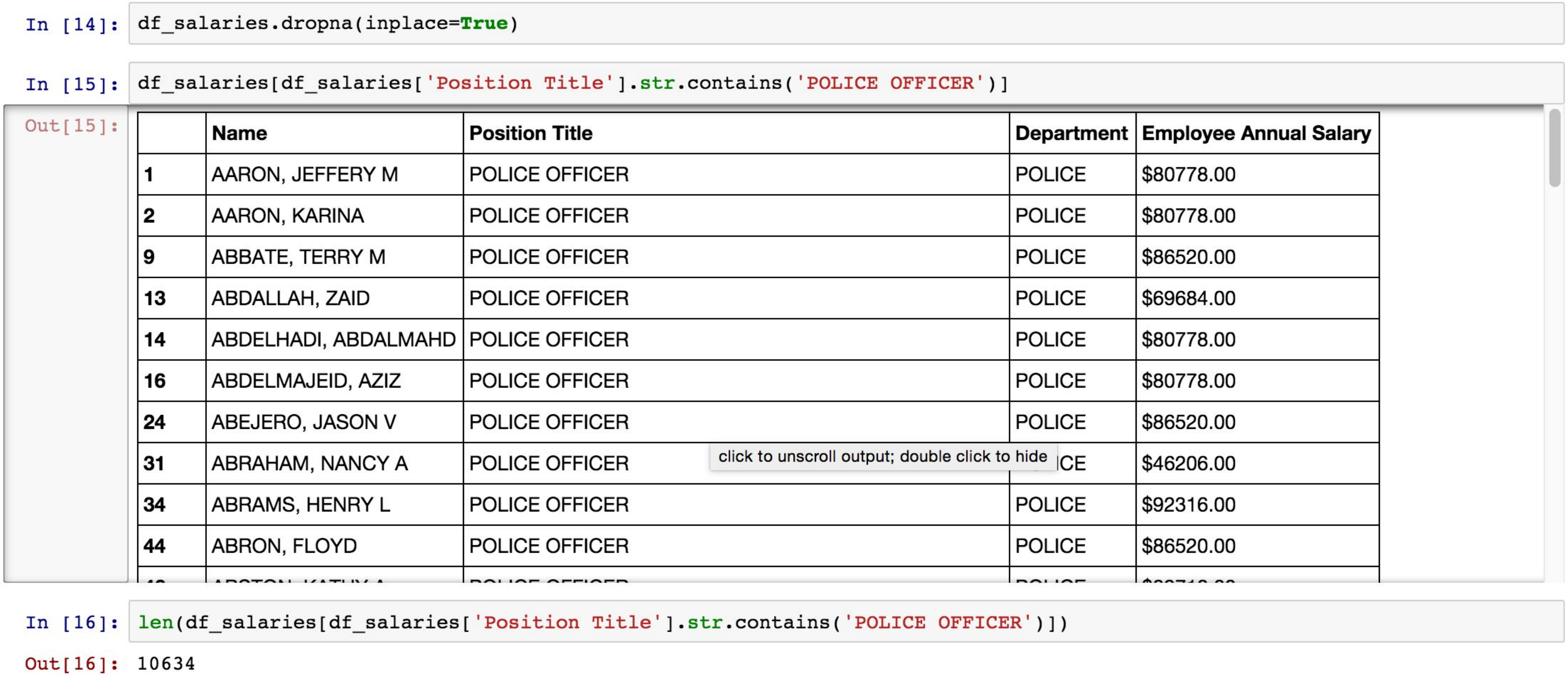
```
Out[36]: 1                AARON,  JEFFERY M
         2                   AARON,  KARINA
         9                ABBATE,  TERRY M
         11              ABBOTT,  LYNISE M
         13                ABDALLAH,  ZAID
         14         ABDELHADI,  ABDALMAHD
         16           ABDELMAJEID,  AZIZ
         21         ABDULLAH,  LAKENYA N
         24              ABEJERO,  JASON V
         31              ABRAHAM,  NANCY A
         34               ABRAMS,  HENRY L
         44                  ABRON,  FLOYD
         46               ABSTON,  KATHY A
         48               ABUDAYEH,  ELIAS
         51          ABUZANAT,  ABDALLA H
         53          ACCARDO,  JENNIFER A
         54            ACCARDO,  THOMAS J
         58             ACEVEDO,  AARON F
         59         ACEVEDO,  ALEJANDRO R
```

Pandas: How many entries where Department == ‘POLICE’?



```
In [13]: len(df_salaries[df_salaries['Department'] == 'POLICE'])

Out[13]: 13570
```

Pandas: How many police officers



```
In [14]: df_salaries.dropna(inplace=True)
```

```
In [15]: df_salaries[df_salaries['Position Title'].str.contains('POLICE OFFICER')]
```

Out[15]:

	Name	Position Title	Department	Employee Annual Salary
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00
9	ABBATE, TERRY M	POLICE OFFICER	POLICE	$86520.00
13	ABDALLAH, ZAID	POLICE OFFICER	POLICE	$69684.00
14	ABDELHADI, ABDALMAHD	POLICE OFFICER	POLICE	$80778.00
16	ABDELMAJEID, AZIZ	POLICE OFFICER	POLICE	$80778.00
24	ABEJERO, JASON V	POLICE OFFICER	POLICE	$86520.00
31	ABRAHAM, NANCY A	POLICE OFFICER	[illegible]ICE	$46206.00
34	ABRAMS, HENRY L	POLICE OFFICER	POLICE	$92316.00
44	ABRON, FLOYD	POLICE OFFICER	POLICE	$86520.00

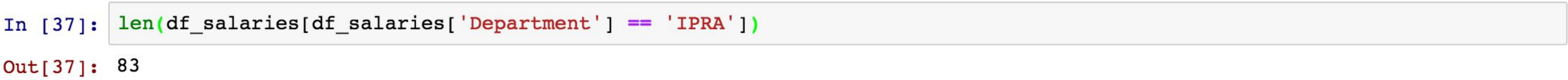
```
In [16]: len(df_salaries[df_salaries['Position Title'].str.contains('POLICE OFFICER')])

Out[16]: 10634
```

Class Challenge 2

How many city employees are in IPRA? (Department == ‘IPRA’)

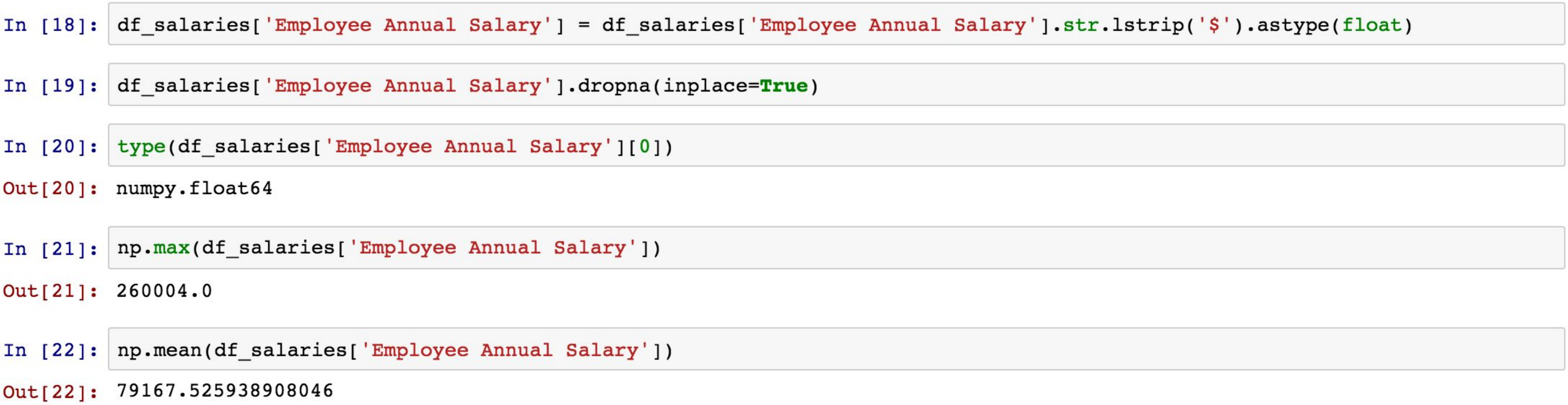
Class Challenge 2

How many city employees are in IPRA? (Department == ‘IPRA’)



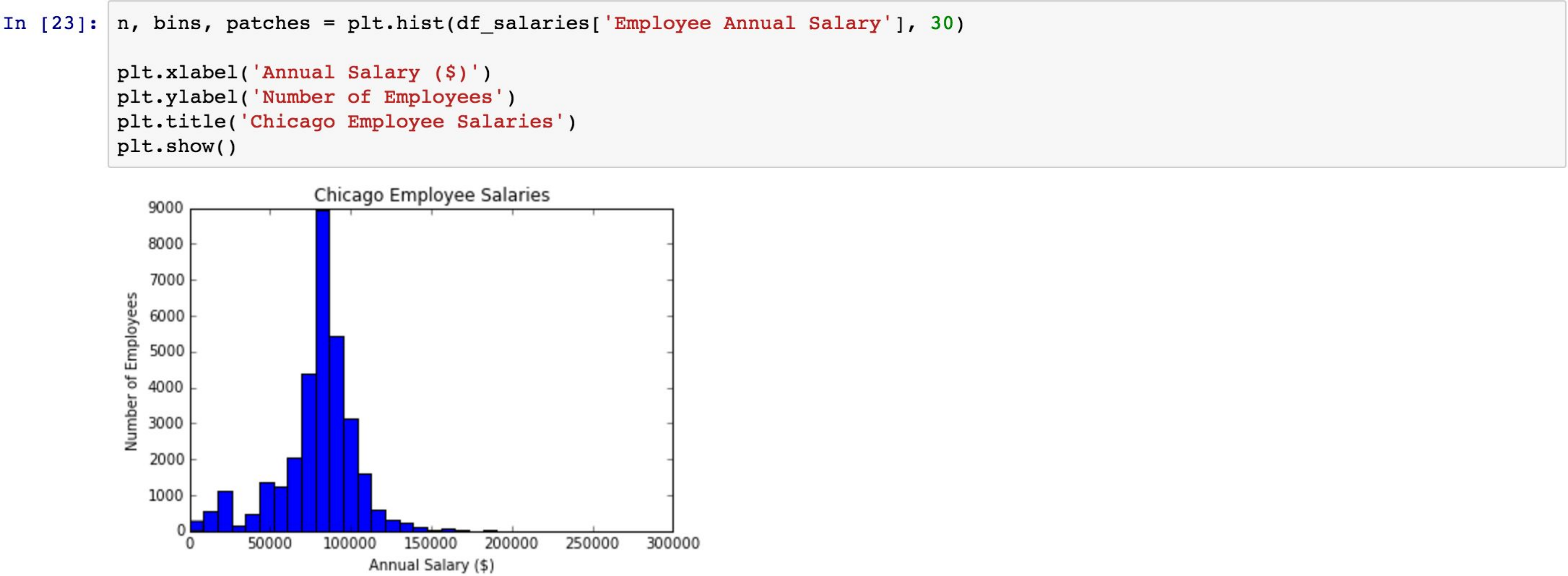
```
In [37]: len(df_salaries[df_salaries['Department'] == 'IPRA'])

Out[37]: 83
```

Plotting



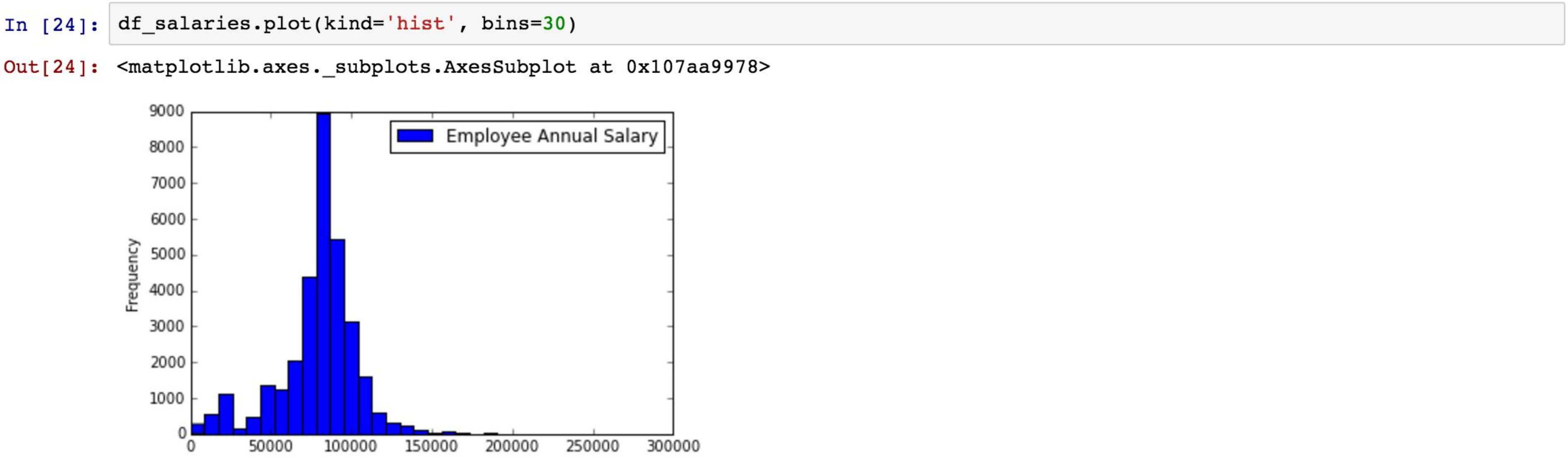
```
In [18]: df_salaries['Employee Annual Salary'] = df_salaries['Employee Annual Salary'].str.lstrip('$').astype(float)

In [19]: df_salaries['Employee Annual Salary'].dropna(inplace=True)

In [20]: type(df_salaries['Employee Annual Salary'][0])

Out[20]: numpy.float64

In [21]: np.max(df_salaries['Employee Annual Salary'])

Out[21]: 260004.0

In [22]: np.mean(df_salaries['Employee Annual Salary'])

Out[22]: 79167.525938908046
```

matplotlib histogram



```
In [23]: n, bins, patches = plt.hist(df_salaries['Employee Annual Salary'], 30)

         plt.xlabel('Annual Salary ($)')
         plt.ylabel('Number of Employees')
         plt.title('Chicago Employee Salaries')
         plt.show()
```

pandas histogram



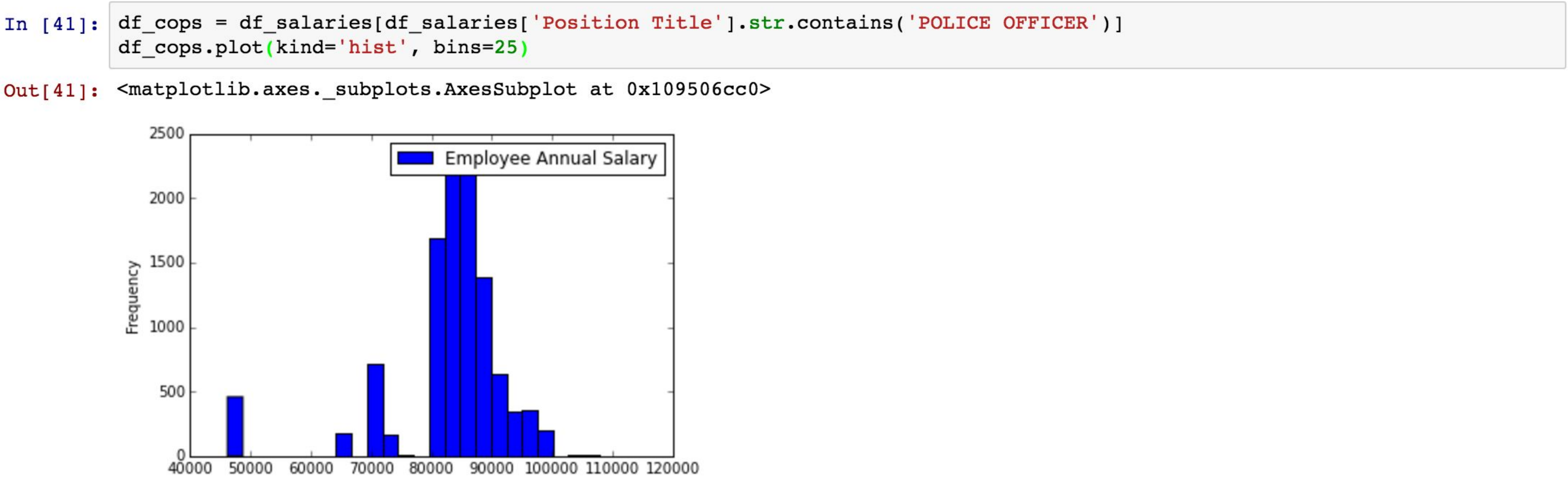
```
In [24]: df_salaries.plot(kind='hist', bins=30)

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x107aa9978>
```

Class Challenge 3

Make a histogram of salaries for city employees, but only for police officers

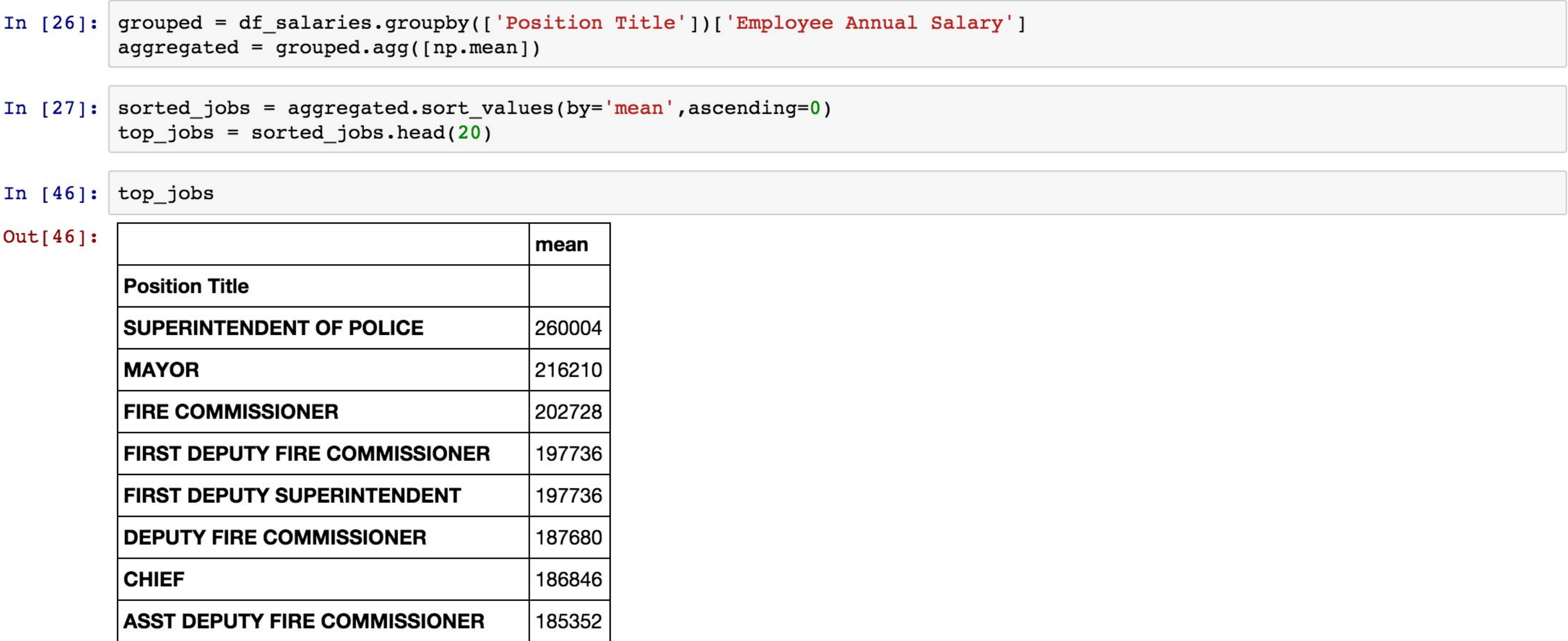
Class Challenge 3

Make a histogram of salaries for city employees, but only for police officers



```
In [41]: df_cops = df_salaries[df_salaries['Position Title'].str.contains('POLICE OFFICER')]
         df_cops.plot(kind='hist', bins=25)

Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x109506cc0>
```

Grouping and Aggregation



```
In [26]: grouped = df_salaries.groupby(['Position Title'])['Employee Annual Salary']
         aggregated = grouped.agg([np.mean])
```

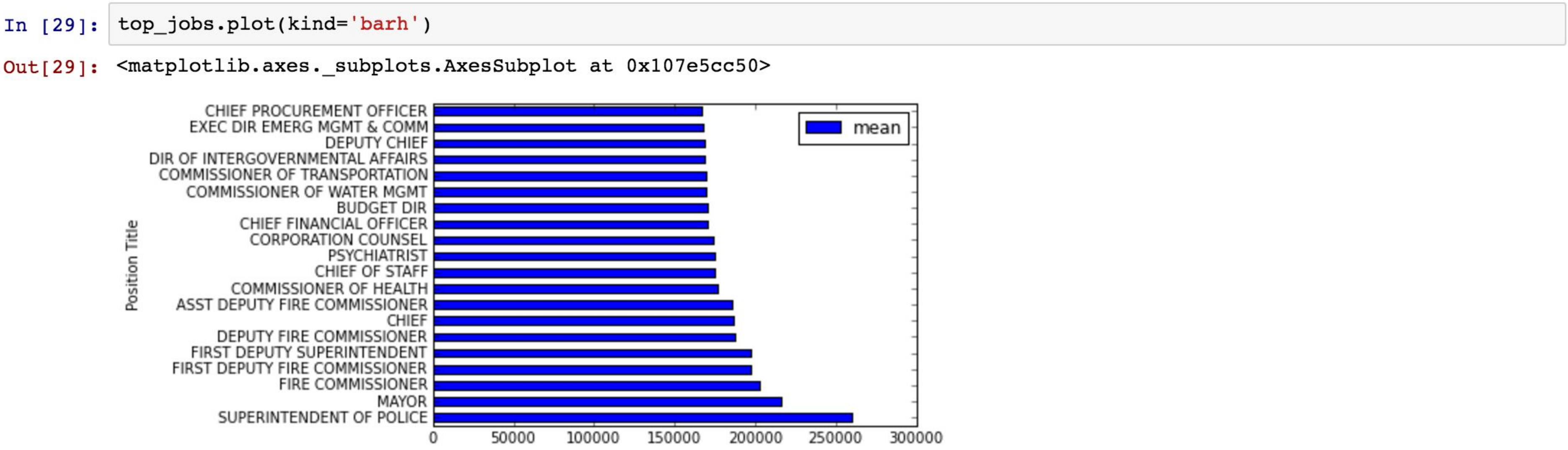
```
In [27]: sorted_jobs = aggregated.sort_values(by='mean',ascending=0)
         top_jobs = sorted_jobs.head(20)
```

```
In [46]: top_jobs
```

Out[46]:

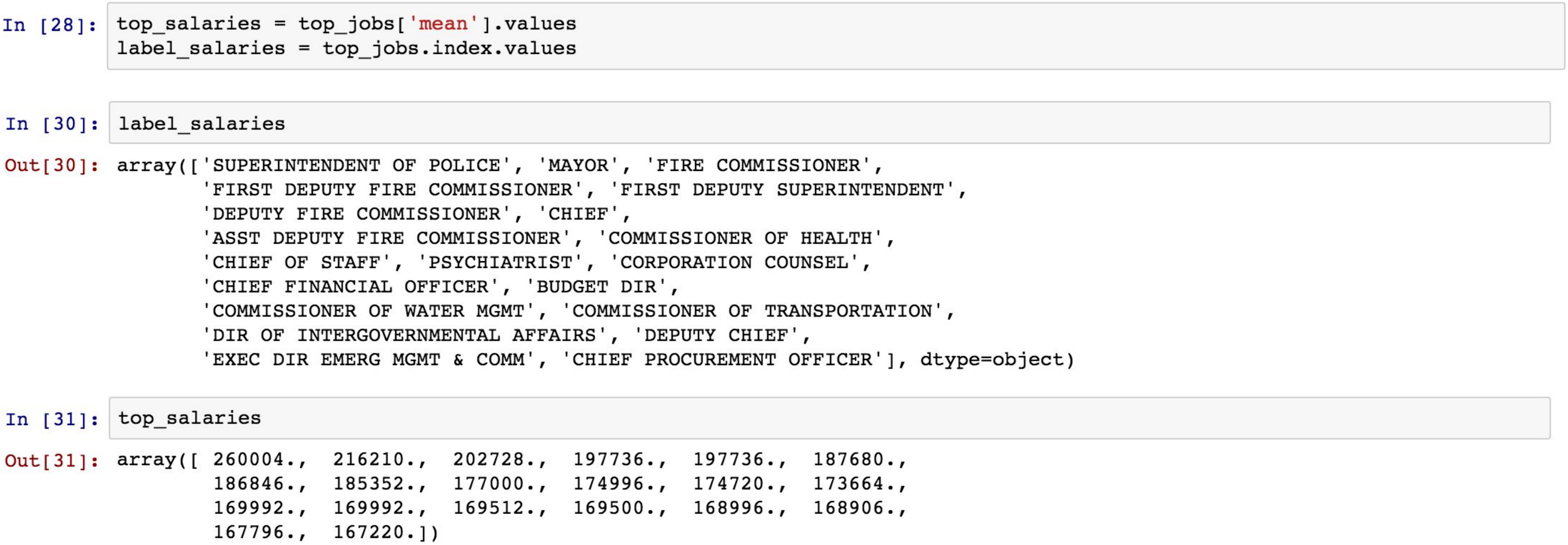
	mean
Position Title	
SUPERINTENDENT OF POLICE	260004
MAYOR	216210
FIRE COMMISSIONER	202728
FIRST DEPUTY FIRE COMMISSIONER	197736
FIRST DEPUTY SUPERINTENDENT	197736
DEPUTY FIRE COMMISSIONER	187680
CHIEF	186846
ASST DEPUTY FIRE COMMISSIONER	185352

Pandas bar chart



```
In [29]: top_jobs.plot(kind='barh')
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x107e5cc50>
```

Matplotlib bar chart

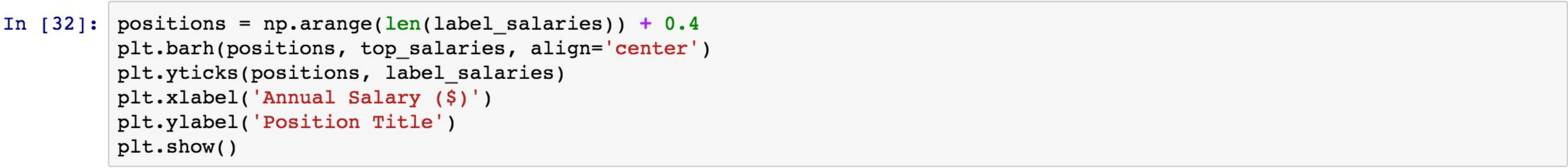


```
In [28]: top_salaries = top_jobs['mean'].values
         label_salaries = top_jobs.index.values
```

```
In [30]: label_salaries

Out[30]: array(['SUPERINTENDENT OF POLICE', 'MAYOR', 'FIRE COMMISSIONER',
                'FIRST DEPUTY FIRE COMMISSIONER', 'FIRST DEPUTY SUPERINTENDENT',
                'DEPUTY FIRE COMMISSIONER', 'CHIEF',
                'ASST DEPUTY FIRE COMMISSIONER', 'COMMISSIONER OF HEALTH',
                'CHIEF OF STAFF', 'PSYCHIATRIST', 'CORPORATION COUNSEL',
                'CHIEF FINANCIAL OFFICER', 'BUDGET DIR',
                'COMMISSIONER OF WATER MGMT', 'COMMISSIONER OF TRANSPORTATION',
                'DIR OF INTERGOVERNMENTAL AFFAIRS', 'DEPUTY CHIEF',
                'EXEC DIR EMERG MGMT & COMM', 'CHIEF PROCUREMENT OFFICER'], dtype=object)
```

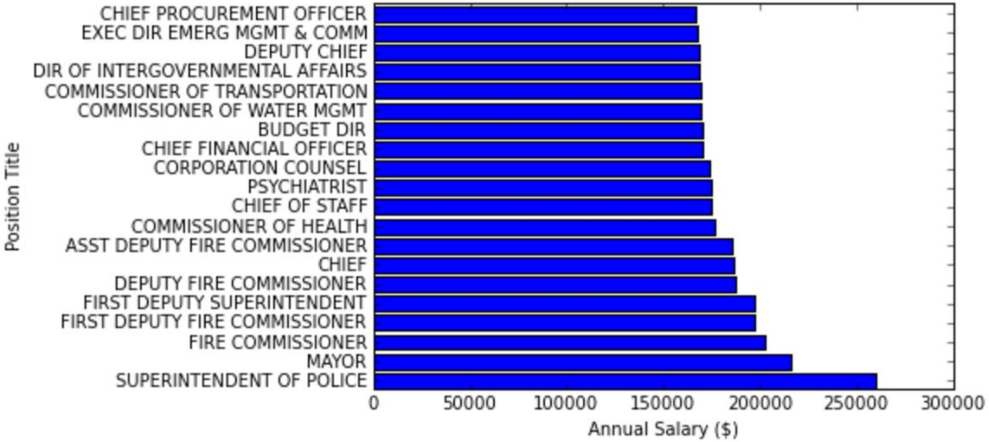
```
In [31]: top_salaries

Out[31]: array([ 260004.,  216210.,  202728.,  197736.,  197736.,  187680.,
                 186846.,  185352.,  177000.,  174996.,  174720.,  173664.,
                 169992.,  169992.,  169512.,  169500.,  168996.,  168906.,
                 167796.,  167220.])
```

Matplotlib bar chart

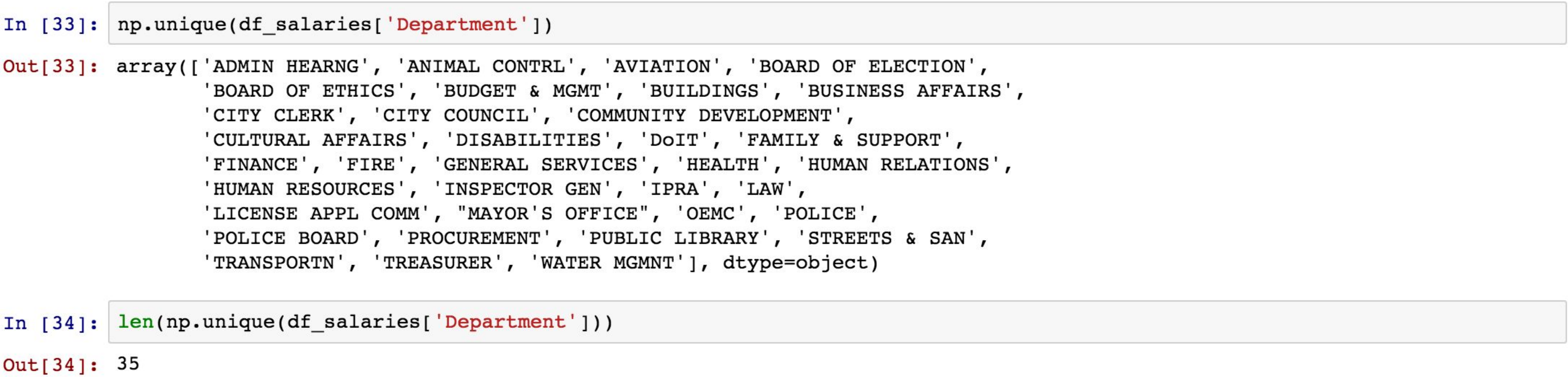


In [32]:

```
positions = np.arange(len(label_salaries)) + 0.4
plt.barh(positions, top_salaries, align='center')
plt.yticks(positions, label_salaries)
plt.xlabel('Annual Salary ($)')
plt.ylabel('Position Title')
plt.show()
```



Unique values



```
In [33]: np.unique(df_salaries['Department'])

Out[33]: array(['ADMIN HEARNG', 'ANIMAL CONTRL', 'AVIATION', 'BOARD OF ELECTION',
                'BOARD OF ETHICS', 'BUDGET & MGMT', 'BUILDINGS', 'BUSINESS AFFAIRS',
                'CITY CLERK', 'CITY COUNCIL', 'COMMUNITY DEVELOPMENT',
                'CULTURAL AFFAIRS', 'DISABILITIES', 'DoIT', 'FAMILY & SUPPORT',
                'FINANCE', 'FIRE', 'GENERAL SERVICES', 'HEALTH', 'HUMAN RELATIONS',
                'HUMAN RESOURCES', 'INSPECTOR GEN', 'IPRA', 'LAW',
                'LICENSE APPL COMM', "MAYOR'S OFFICE", 'OEMC', 'POLICE',
                'POLICE BOARD', 'PROCUREMENT', 'PUBLIC LIBRARY', 'STREETS & SAN',
                'TRANSPORTN', 'TREASURER', 'WATER MGMNT'], dtype=object)

In [34]: len(np.unique(df_salaries['Department']))

Out[34]: 35
```