



(<http://www.pieriandata.com>)

---

Copyright Pierian Data

For more information, visit us at [www.pieriandata.com](http://www.pieriandata.com) (<http://www.pieriandata.com>).

## Matplotlib Basics

Here we cover the minimum basics of matplotlib functionality, just enough to understand how Pandas plotting and Seaborn are built on top of Matplotlib. We will mainly use Pandas or Seaborn plotting throughout the course, here we show just the basic interactions possible with matplotlib. Do not consider this a comprehensive guide! For more information on matplotlib, visit: <https://matplotlib.org/tutorials/index.html> (<https://matplotlib.org/tutorials/index.html>).

---

```
In [4]: import numpy as np
import pandas as pd
```

## Visualizing Plots

Review video to understand this section!

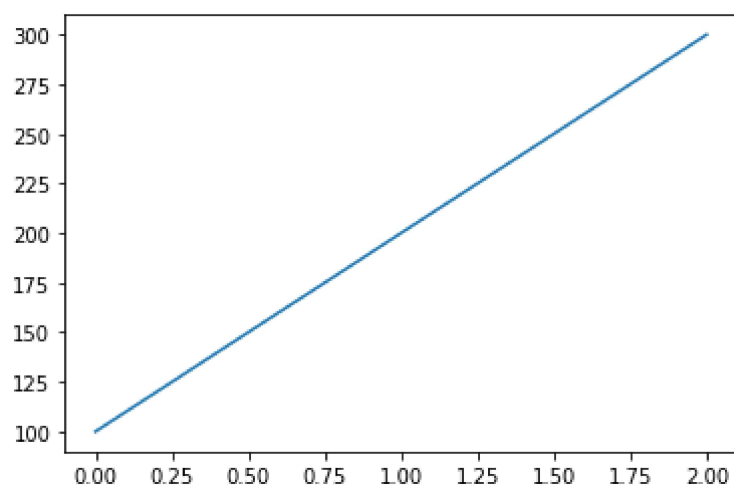
```
In [5]: import matplotlib.pyplot as plt

# JUPYTER NOTEBOOK ONLY
# %matplotlib inline
```

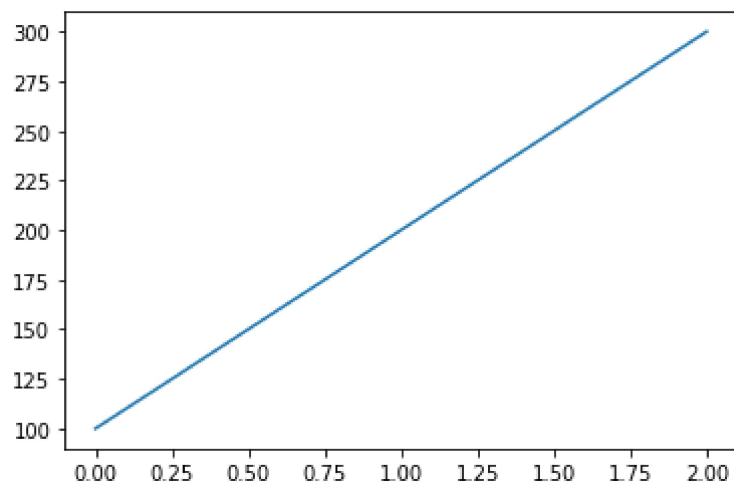
```
In [6]: x = [0,1,2]
y = [100,200,300]
```

```
In [8]: plt.plot(x,y)
```

```
Out[8]: [<matplotlib.lines.Line2D at 0x169e481f4c8>]
```

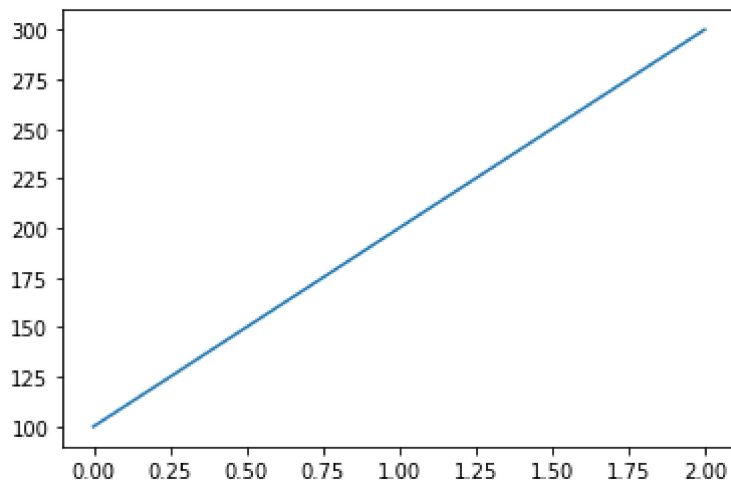


```
In [9]: # add semicolon to hide matplotlib text output  
plt.plot(x,y);
```



```
In [10]: # When running a .py file , you need to add plt.show() at the end of your command
```

```
In [11]: # For running .py files!  
plt.plot(x,y)  
plt.show()
```



## Basic Tools

We will only use pure matplotlib for really quick, basic plots.

```
In [12]: housing = pd.DataFrame({'rooms': [1, 1, 2, 2, 2, 3, 3, 3],  
                                'price': [100, 120, 190, 200, 230, 310, 330, 305]})
```

```
In [13]: housing
```

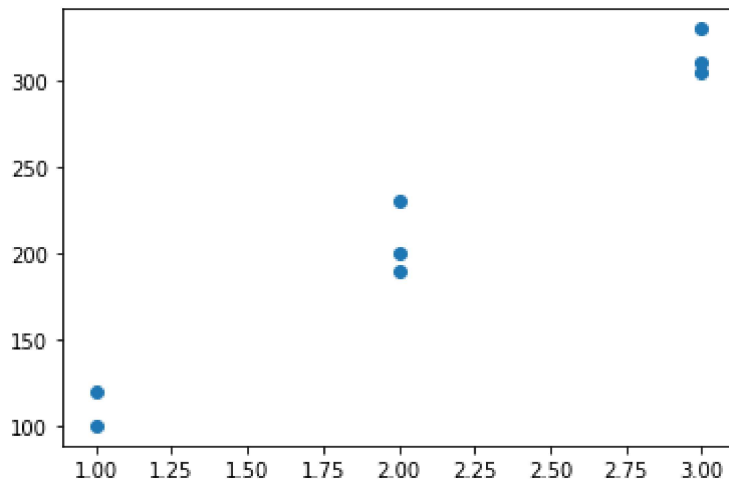
```
Out[13]:
```

	rooms	price
0	1	100
1	1	120
2	2	190
3	2	200
4	2	230
5	3	310
6	3	330
7	3	305

```
In [16]: # Probably not a great plot, since this implies a continuous relationship!  
# plt.plot(housing['rooms'], housing['price'])
```

```
In [17]: plt.scatter(housing['rooms'],housing['price'])
```

```
Out[17]: <matplotlib.collections.PathCollection at 0x169e4a5eec8>
```

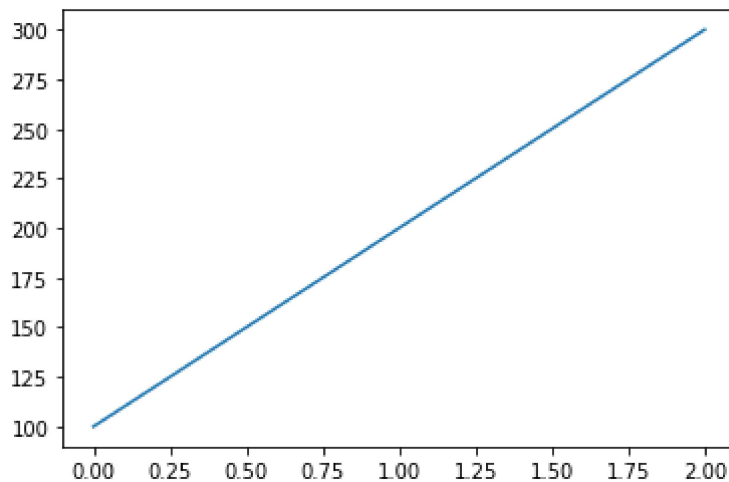


## Style Calls

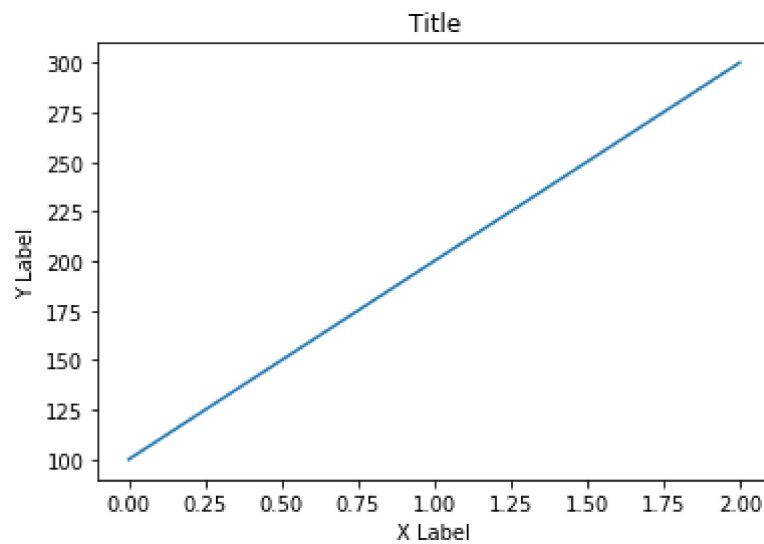
One of the main reasons to learn the absolute basics is to see how the style interactions effect the API.

```
In [18]: plt.plot(x,y)
```

```
Out[18]: [<matplotlib.lines.Line2D at 0x169e4ac9888>]
```



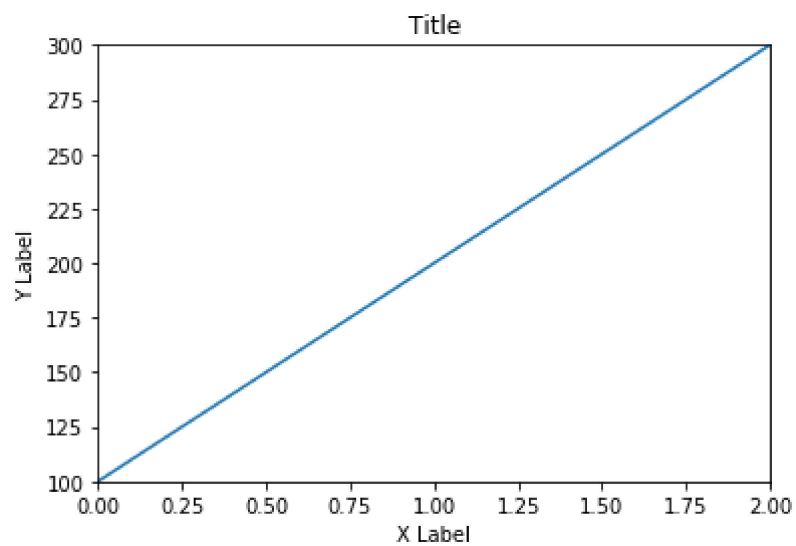
```
In [21]: plt.plot(x,y)
plt.title('Title')
plt.xlabel('X Label')
plt.ylabel('Y Label');
```



```
In [22]: plt.plot(x,y)

# Axis and ticks
plt.xlim(0,2)
plt.ylim(100,300)

# Labeling
plt.title('Title')
plt.xlabel('X Label')
plt.ylabel('Y Label');
```



```
In [24]: plt.plot(x,y,color='red')
```

```
# Axis and ticks
```

```
plt.xlim(0,2)
```

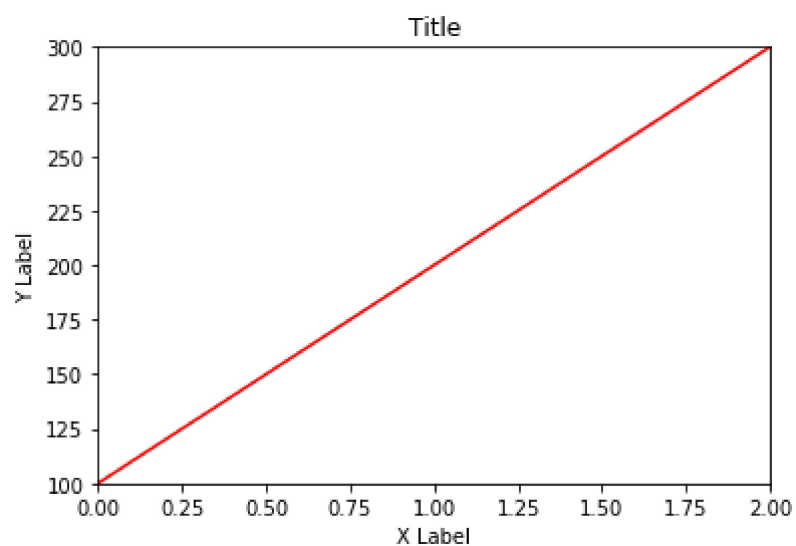
```
plt.ylim(100,300)
```

```
# Labeling
```

```
plt.title('Title')
```

```
plt.xlabel('X Label')
```

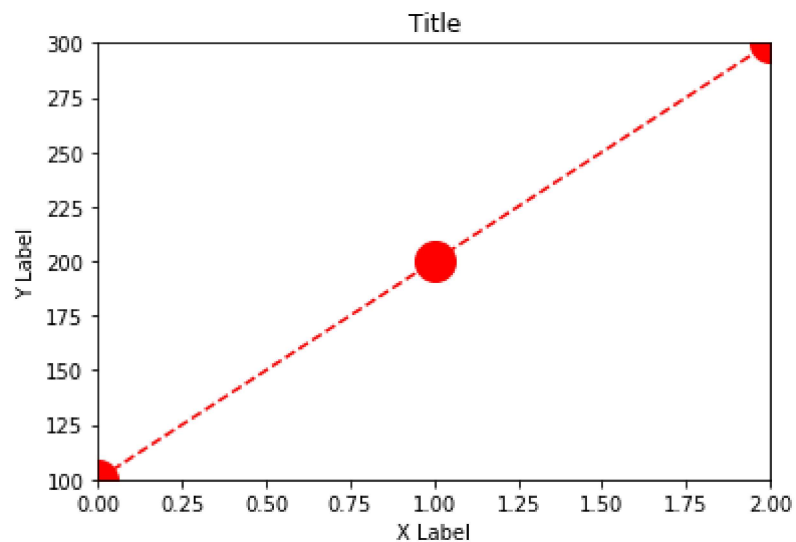
```
plt.ylabel('Y Label');
```



```
In [33]: plt.plot(x,y,color='red',marker='o',markersize=20,linestyle='--')

# Axis and ticks
plt.xlim(0,2)
plt.ylim(100,300)

# Labeling
plt.title('Title')
plt.xlabel('X Label')
plt.ylabel('Y Label');
```



That is all we will cover for matplotlib, the rest will be more intuitive with our use of pandas and seaborn!