

In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
```

`plt.plot` 에 list 두 개가 아닌 행이 두 개인 `np.array` object를 넣게되면 list 두 개를 넣었을때와는 다른 모양의 그래프들이 나옵니다. `b = np.matrix([[x_11, x_12, x_13], [x_21, x_22, x_23]])`의  $2 \times 3$  matrix 형태의 `np.array` object를 넣게되면 세 개의 그래프들이 나오는데 첫번째 그래프는 두 점

$$(0, x_{11}), (1, x_{21})$$

을 이은 것이고 두번째 그래프는 두 점

$$(0, x_{12}), (1, x_{22})$$

을 이은 것이며, 세번째 그래프는 두 점

$$(0, x_{13}), (1, x_{23})$$

을 이은 것입니다.

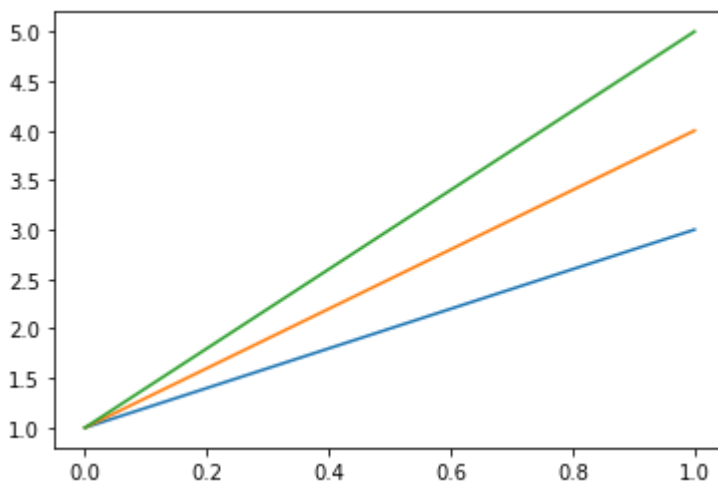
따라서, 굳이 행이 두 개인 `np.array` object가 아니더라도 그래프는 그려집니다. 정리하면  $m \times n$  matrix 형태의 `np.array` object의 경우  $n$ 개의 그래프가 나타나는데 이 그래프들 각각은  $m$ 개의 점들을 차례로 이은 것입니다.

In [2]:

```
b = np.array([[1, 1, 1], [3, 4, 5]])
plt.plot(b)
```

Out[2]:

```
[<matplotlib.lines.Line2D at 0x1179f4fa4f0>,
<matplotlib.lines.Line2D at 0x1179f4fa520>,
<matplotlib.lines.Line2D at 0x1179f4fa550>]
```



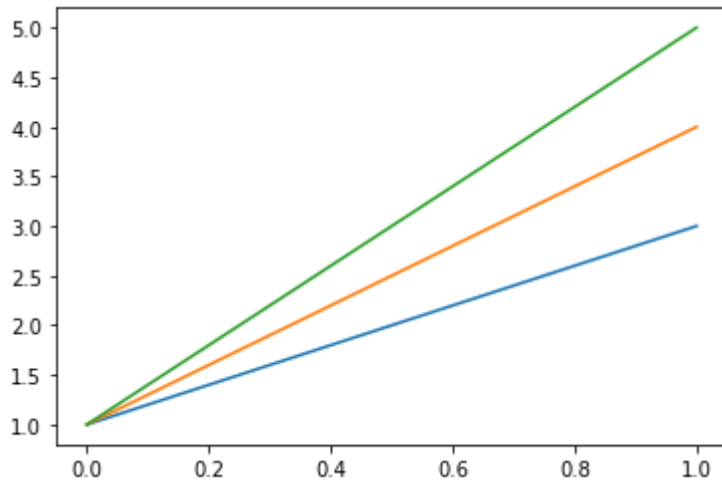
또한,  $m \times n$  matrix 형태의 python list object라고 하더라도 `np.array`때와 정확히 같은 형태의 결과가 나타납니다.

In [3]:

```
b = [[1, 1, 1], [3, 4, 5]]  
plt.plot(b)
```

Out[3]:

```
[<matplotlib.lines.Line2D at 0x1179f607a90>,  
<matplotlib.lines.Line2D at 0x1179f607ac0>,  
<matplotlib.lines.Line2D at 0x1179f607be0>]
```



심지어, np.matrix일 때도 잘 작동합니다.

따라서, 제 자료에서 " X 와 Y 가 np.matrix로 주어지거나 pandas의 dataframe으로 주어질 수 없다."라는 문장은 정확하지 않은 문장입니다. 공식 홈페이지의 "All of plotting functions expect numpy.array or numpy.ma.masked\_array as input. Classes that are 'array-like' such as pandas data objects and numpy.matrix may or may not work as intended. It is best to convert these to numpy.array objects prior to plotting." 라는 문장을 제가 오역한 것 같습니다.