In [2]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
# use a for stock prices
a = np.array([8.70,8.91,8.71])
a
```

Out[3]:
```
array([8.7 , 8.91, 8.71])
```

In [4]:
```
a[1:]/a[:-1] -1
```

Out[4]:
```
array([ 0.02413793, -0.02244669])
```

In [5]:
```
stock_prices = pd.DataFrame({"Stock1": [8.70,8.91,8.71,8.43,8.73],
                             "Stock2": [10.66, 11.08, 10.71, 11.59,12.11]})
stock_prices
```

Out[5]:

	Stock1	Stock2
0	8.70	10.66
1	8.91	11.08
2	8.71	10.71
3	8.43	11.59
4	8.73	12.11

In [8]:
```
#Basic returns
stock_prices.iloc[1:].values/stock_prices.iloc[:-1] - 1
```

Out[8]:

	Stock1	Stock2
0	0.024138	0.039400
1	-0.022447	-0.033394
2	-0.032147	0.082166
3	0.035587	0.044866

In [9]:
```
#another way to do this. It will add an additional row.
stock_prices/stock_prices.shift(1) - 1
```

Out[9]:

	Stock1	Stock2
0	NaN	NaN
1	0.024138	0.039400
2	-0.022447	-0.033394
3	-0.032147	0.082166
4	0.035587	0.044866

In [10]:
```
# you can also use pct_change()
stock_prices.pct_change()
```

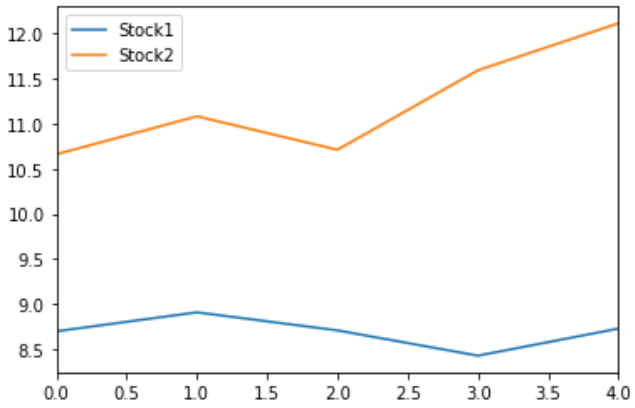
Out[10]:

	Stock1	Stock2
0	NaN	NaN
1	0.024138	0.039400
2	-0.022447	-0.033394
3	-0.032147	0.082166
4	0.035587	0.044866

In [11]:
```
returns = stock_prices.pct_change()
```

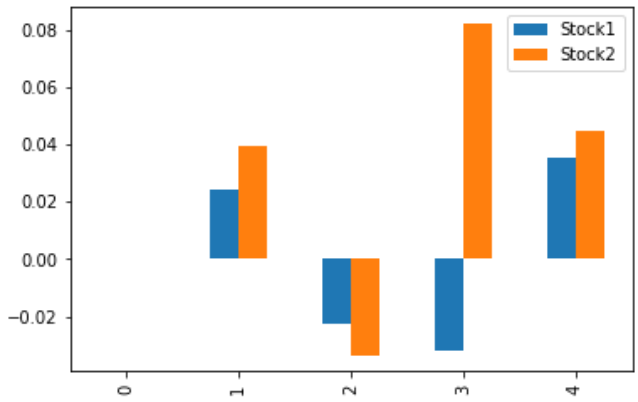
In [13]:
```
stock_prices.plot()
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x11598b908>



In [15]:
```
returns.plot.bar()
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x115c28e48>



In [16]:
```
returns.std()
```

Out[16]:
```
Stock1    0.033565
Stock2    0.048328
dtype: float64
```

In [17]:
```
returns.mean()
```

Out[17]:
```
Stock1    0.001283
Stock2    0.033260
dtype: float64
```

In [18]:
```
#compund return
np.prod(returns + 1) - 1
```

Out[18]:
```
Stock1    0.003448
Stock2    0.136023
dtype: float64
```

In [19]:
```
#another way
(returns + 1).prod() - 1
```

Out[19]:
```
Stock1    0.003448
Stock2    0.136023
dtype: float64
```

In [20]:
```
#another format
(((returns + 1).prod() - 1)*100).round(2)
```

Out[20]:
```
Stock1     0.34
Stock2    13.60
dtype: float64
```

Annualization

In [23]:
```
#per year
rm = 0.01
(1+rm)**12-1
```

Out[23]:
```
0.12682503013196977
```

In [25]:
```
#per quarter
rq = 0.04
(1 + rq)**4 -1
```

Out[25]:
```
0.1698585600000002
```

In [27]:
```
#per trading day
rd = 0.0001
(1 + rd)**252 - 1
```

Out[27]:
```
0.025518911987694626
```

In [ ]:
```

```