Orthogonalizing vectors

In [1]:

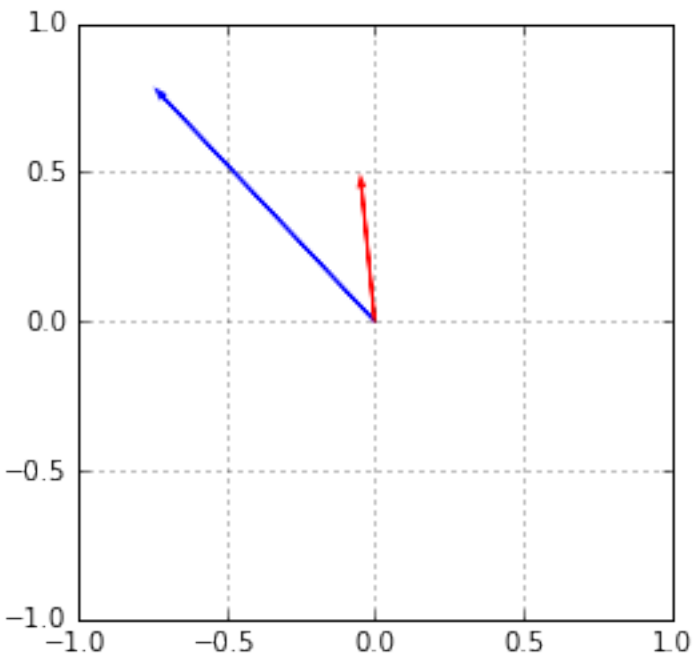
```
#keep
import numpy as np
import matplotlib.pyplot as pt
%matplotlib inline
```

In [2]:

```
#keep
np.random.seed(13)
x = np.random.randn(2)
y = np.random.randn(2)
```

In [3]:

```
#keep
pt.arrow(0, 0, x[0], x[1], color="blue")
pt.arrow(0, 0, y[0], y[1], color="red")
pt.xlim([-1, 1])
pt.ylim([-1, 1])
pt.gca().set_aspect("equal")
pt.grid()
```



Are those orgonal? How would we find out?

In [4]:

```
x.dot(y)
```

Out[4]:

```
0.37226452785072001
```

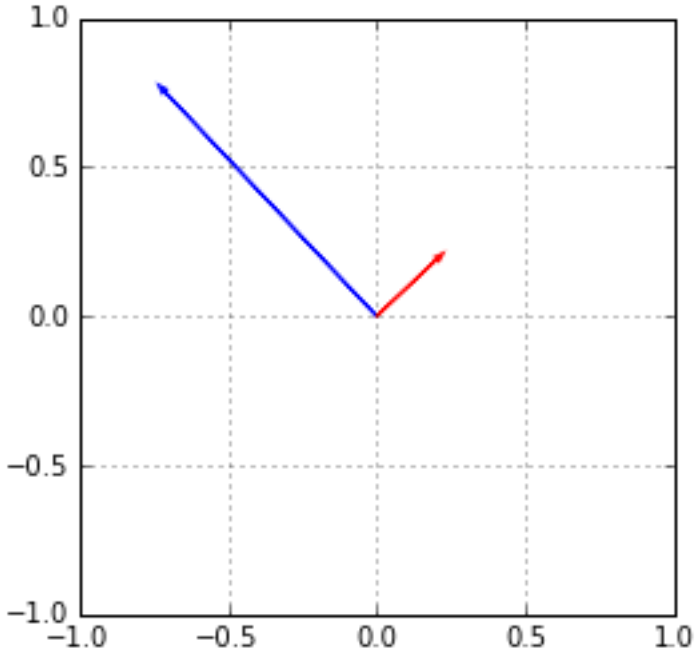
Now use the formula to make `ynew` which *is* orthogonal to `x`:

In [5]:

```
ynew = y - y.dot(x)/x.dot(x)*x
```

In [6]:

```
#keep
pt.arrow(0, 0, x[0], x[1], color="blue")
pt.arrow(0, 0, ynew[0], ynew[1], color="red")
pt.xlim([-1, 1])
pt.ylim([-1, 1])
pt.gca().set_aspect("equal")
pt.grid()
```



In [ ]:

```

```