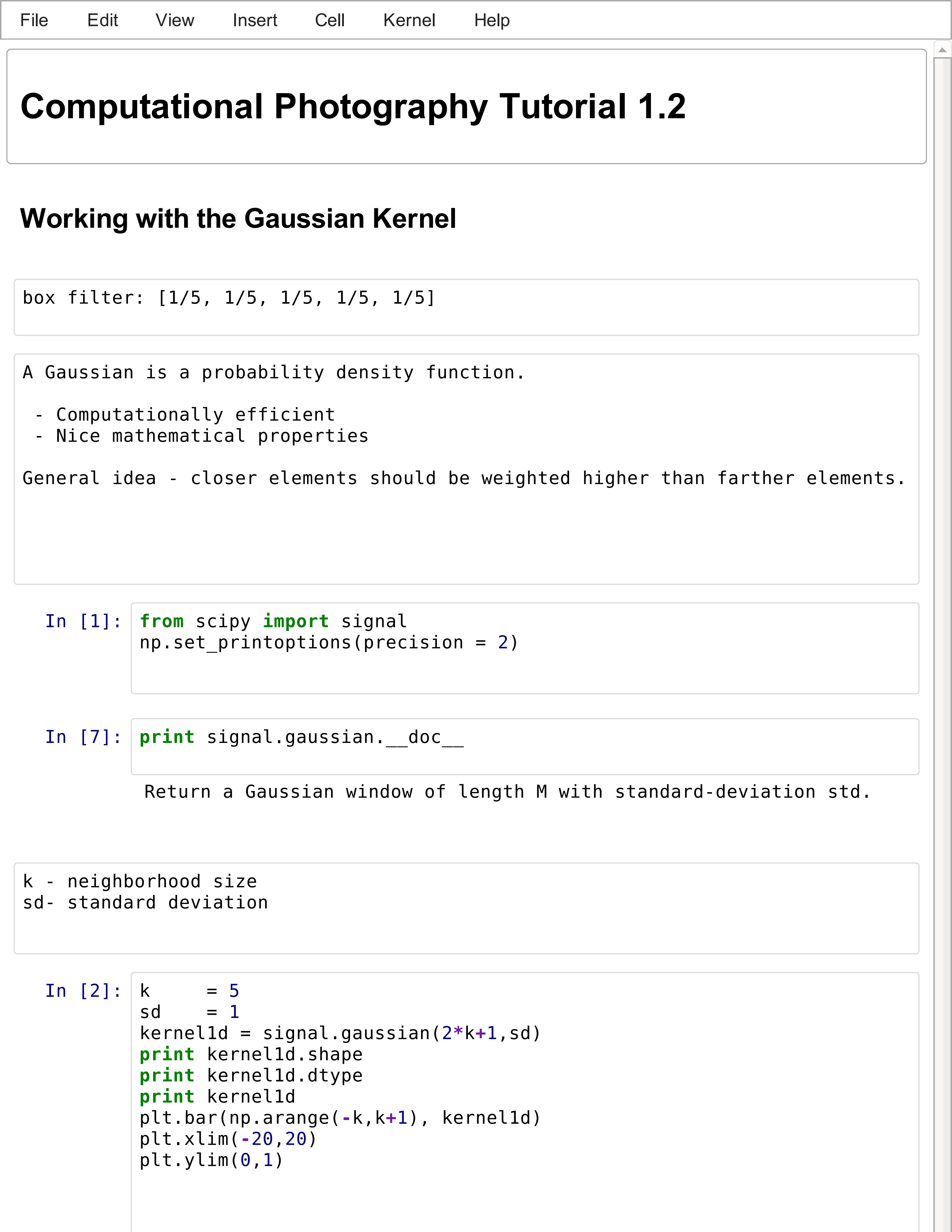


Computational Photography Tutorial 1.2

Working with the Gaussian Kernel

```
box filter: [1/5, 1/5, 1/5, 1/5, 1/5]
```

```
A Gaussian is a probability density function.

 - Computationally efficient
 - Nice mathematical properties

General idea - closer elements should be weighted higher than farther elements.
```

In [1]:
```
from scipy import signal
np.set_printoptions(precision = 2)
```

In [7]:
```
print signal.gaussian.__doc__
```

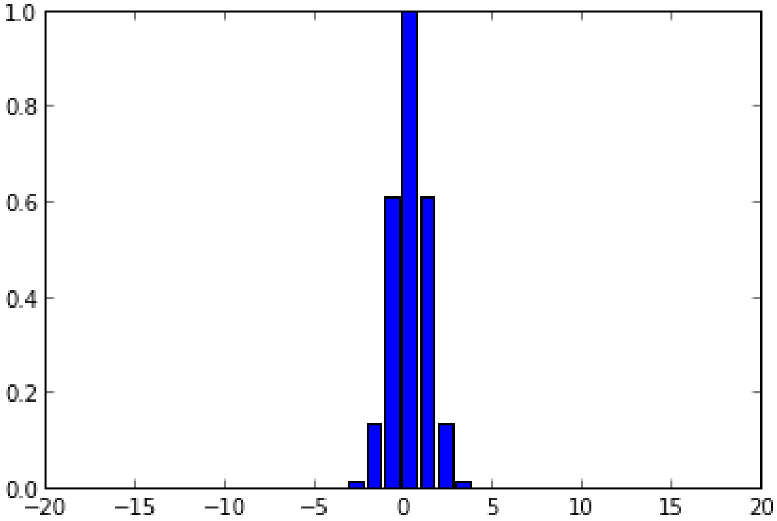
```
Return a Gaussian window of length M with standard-deviation std.
```

```
k - neighborhood size
sd- standard deviation
```

In [2]:
```
k     = 5
sd    = 1
kernel1d = signal.gaussian(2*k+1,sd)
print kernel1d.shape
print kernel1d.dtype
print kernel1d
plt.bar(np.arange(-k,k+1), kernel1d)
plt.xlim(-20,20)
plt.ylim(0,1)
```

```
(11,)
float64
[  3.73e-06   3.35e-04   1.11e-02   1.35e-01   6.07e-01   1.00e+00
   6.07e-01   1.35e-01   1.11e-02   3.35e-04   3.73e-06]
```

```
Out[2]: (0, 1)
```

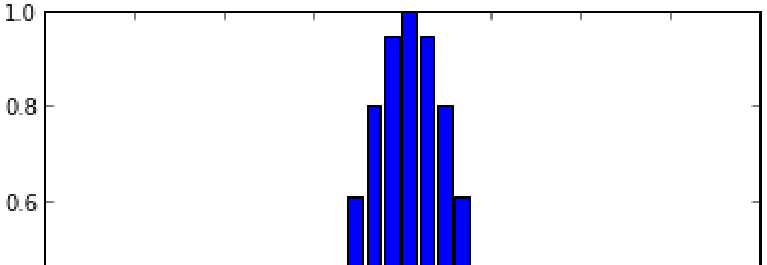


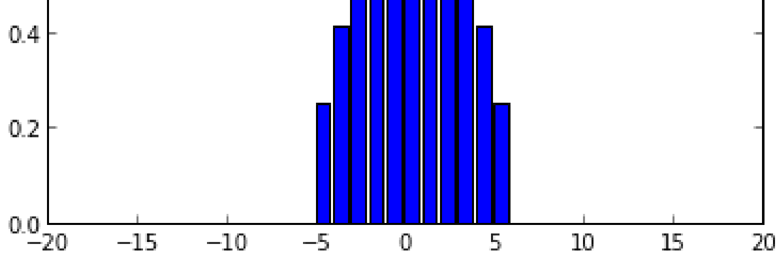
Varying sd makes neighbors more or less influential.

```
In [3]: k      = 5
        sd     = 3
        kernel1d = signal.gaussian(2*k+1,sd)
        print kernel1d.shape
        print kernel1d.dtype
        print kernel1d
        plt.bar(np.arange(-k,k+1), kernel1d)
        plt.xlim(-20,20)
        plt.ylim(0,1)
```

```
(11,)
float64
[ 0.25  0.41  0.61  0.8   0.95  1.    0.95  0.8   0.61  0.41  0.25]
```

```
Out[3]: (0, 1)
```





Varying k selects fewer or more neighbors.

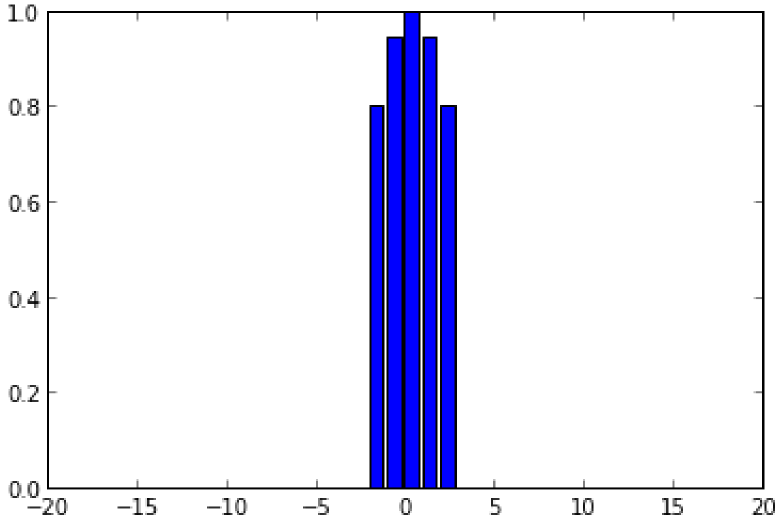
In [5]:

```
k      = 2
sd     = 3
kernel1d = signal.gaussian(2*k+1,sd)
print kernel1d.shape
print kernel1d.dtype
print kernel1d
plt.bar(np.arange(-k,k+1), kernel1d)
plt.xlim(-20,20)
plt.ylim(0,1)
```

```
(5,)
float64
[ 0.8   0.95  1.    0.95  0.8 ]
```

Out[5]:

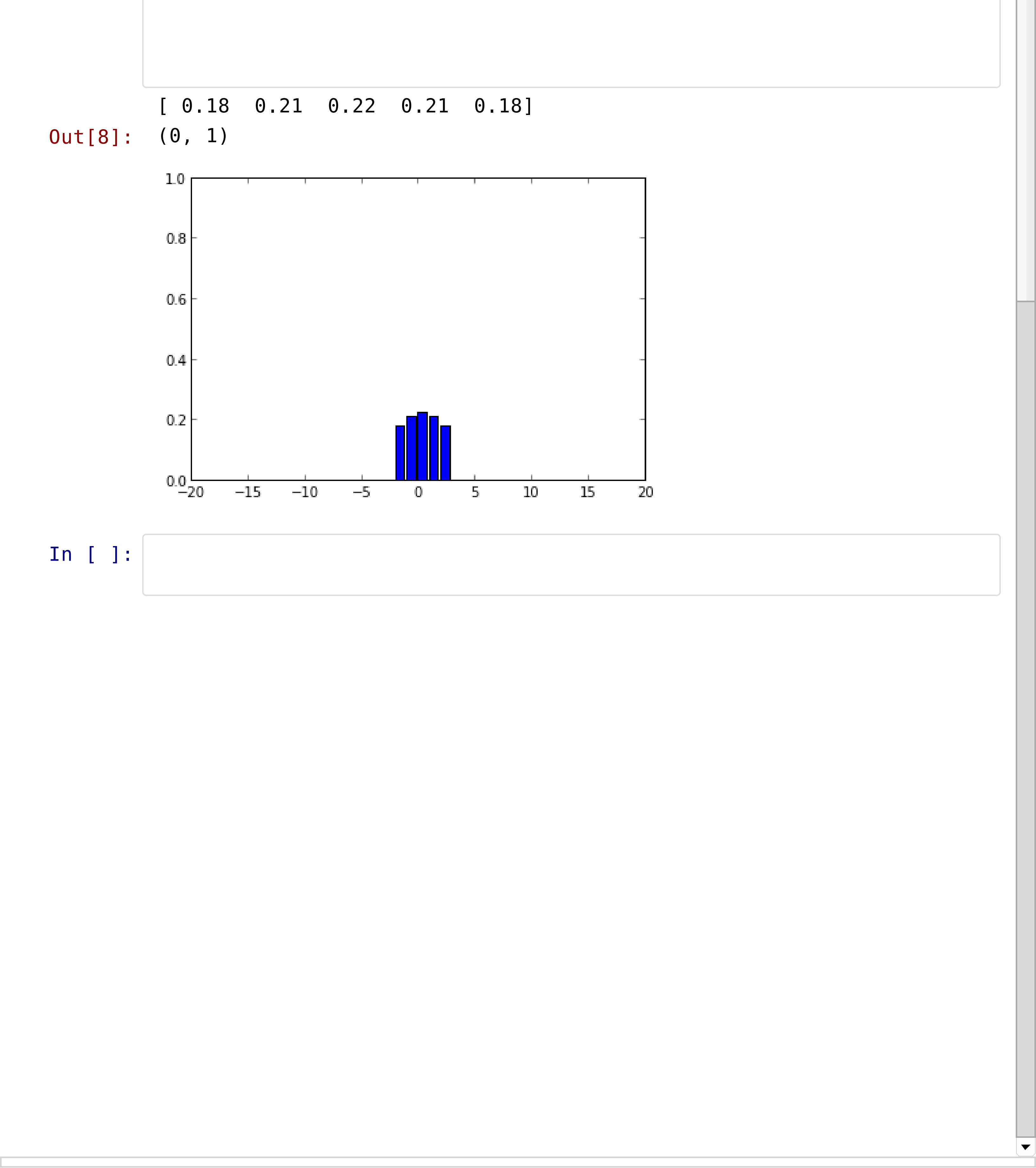
```
(0, 1)
```



Normalizing ensures that our output is on the same scale as the input (stay in the 0,255 range).

In [8]:

```
kernel1d = kernel1d/kernel1d.sum()
print kernel1d
plt.bar(np.arange(-k,k+1), kernel1d)
plt.xlim(-20,20)
plt.ylim(0,1)
```



```
[ 0.18  0.21  0.22  0.21  0.18]
```

```
Out[8]: (0, 1)
```

```
In [ ]:
```