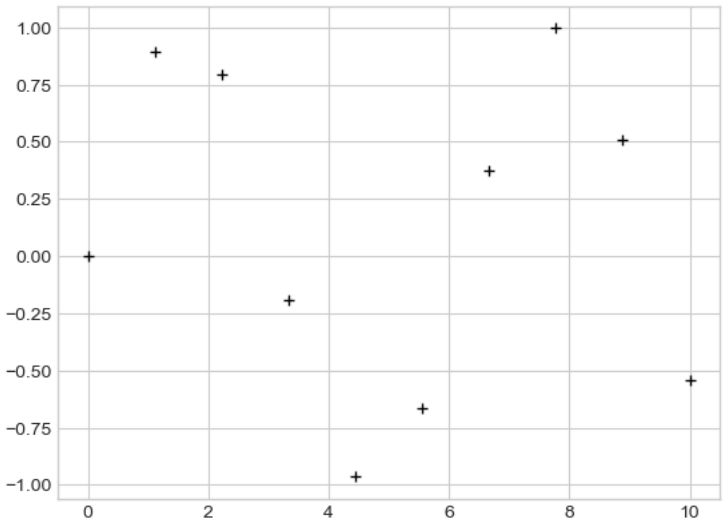
In [1]:
```
#simple scatter plots in matplotlib
```

In [2]:
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
import numpy as np
```

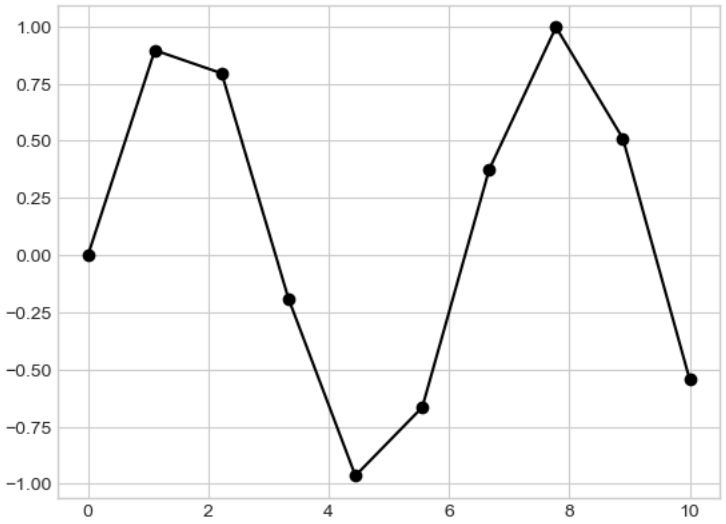
In [8]:
```
x=np.linspace(0,10,10)
y=np.sin(x)
plt.plot(x,y,"+",color="black")#"(symbol)" allows as to create scatter plots from plt.plot()
```

Out[8]: [<matplotlib.lines.Line2D at 0x1a5bfefea30>]



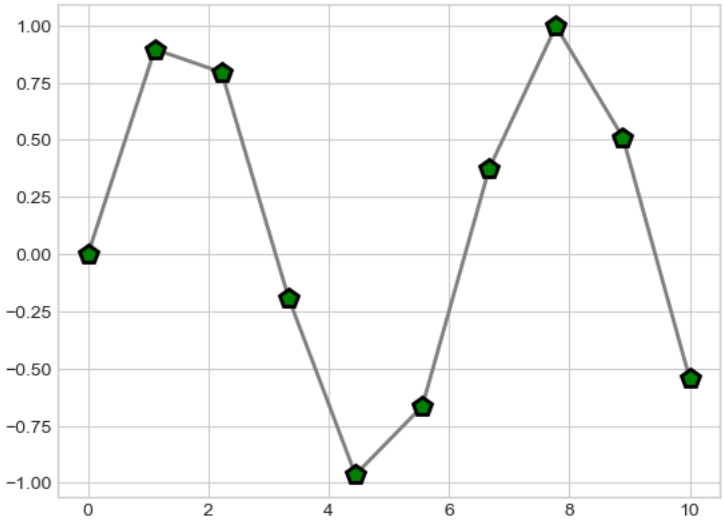
In [12]:
```
#combining scatter plots and line plots
x=np.linspace(0,10,10)
y=np.sin(x)
plt.plot(x,y,"-ok")
```

Out[12]: [<matplotlib.lines.Line2D at 0x1a5bff7d0d0>]



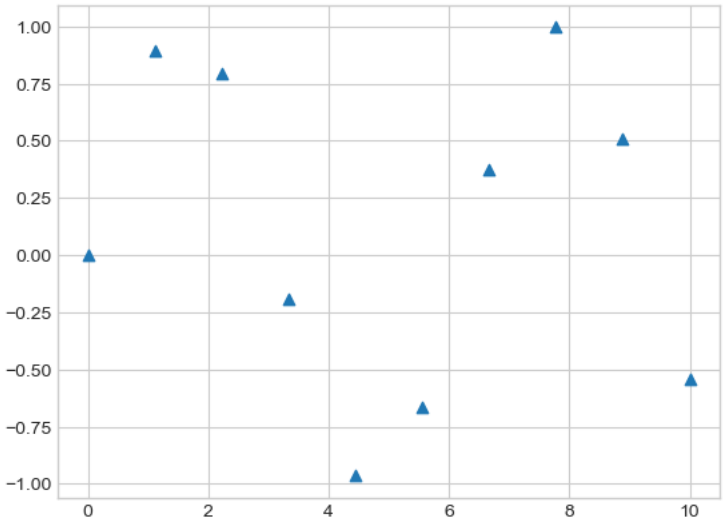
In [23]:
```
#additional plot arguments in matplotlib
plt.plot(x,y,"-p",color="grey",markersize=10,linewidth=2,markerfacecolor="green",markeredgecolor="black",markeredgewidth=2)
```

Out[23]:
```
[<matplotlib.lines.Line2D at 0x1a5c146d1c0>]
```



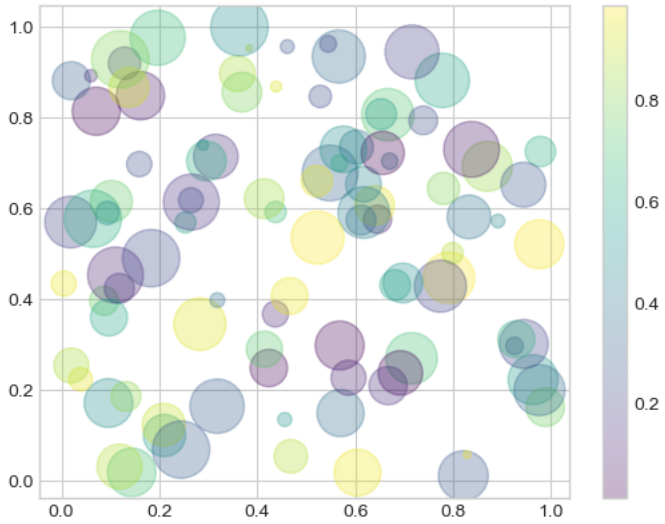
In [24]:
```
#scatter plots with plt.scatter()
plt.scatter(x,y,marker="^")
```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x1a5c14c8040>
```



In [51]:
```
rng=np.random.RandomState(0)
x=rng.rand(100)
y=rng.rand(100)
colors=rng.rand(100)
sizes=1000* rng.rand(100)
plt.scatter(x,y,c=colors,s=sizes,alpha=0.3,cmap="viridis")
#c argument take array of greyscale values
#s argument takes array of sizes
#alpha sets transperency of scatter plots
#cmap maps greyscale values to colors
plt.colorbar()#shows color scale
```

Out[51]:
```
<matplotlib.colorbar.Colorbar at 0x1a5c275f2b0>
```



In [52]:
```
#for larger datasets plt.plot() is efficient because it treats each point as a same object
#whereas in plt.scatter() each point object has to be constructed seperately
```