

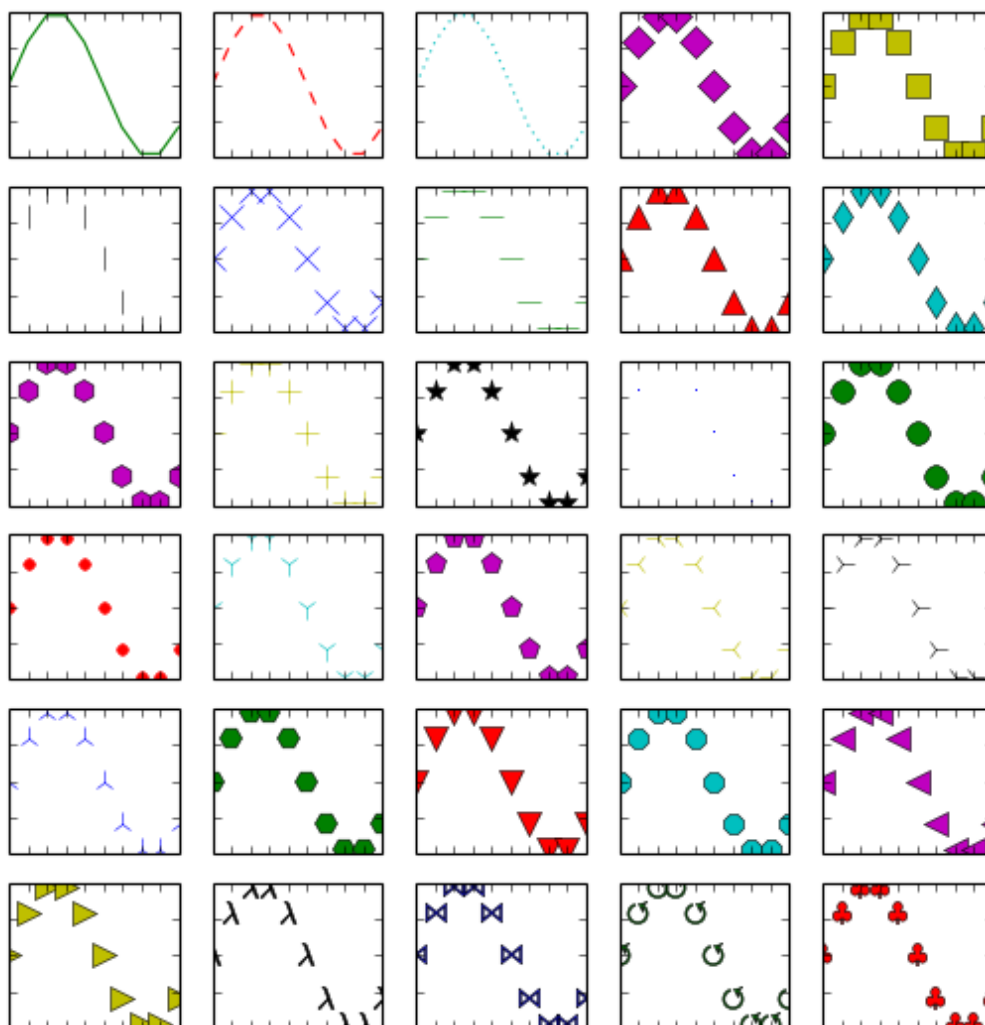
## pylab\_examples example code: line\_styles.py

([Source code](#), [png](#), [hires.png](#), [pdf](#))

[Show Source](#)



module,



```
#!/usr/bin/env python
# This should probably be replaced with a demo that shows all
# line and marker types in a single panel, with labels.
```

```
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
import numpy as np
```

```
t = np.arange(0.0, 1.0, 0.1)
s = np.sin(2*np.pi*t)
```

```

linestyles = ['_', '-', '--', ':']
markers = []
for m in Line2D.markers:
    try:
        if len(m) == 1 and m != ' ':
            markers.append(m)
    except TypeError:
        pass

styles = markers + [
    r'$\lambda$',
    r'$\bowtie$',
    r'$\circlearrowleft$',
    r'$\clubsuit$',
    r'$\checkmark$']

colors = ('b', 'g', 'r', 'c', 'm', 'y', 'k')

plt.figure(figsize=(8,8))

axisNum = 0
for row in range(6):
    for col in range(5):
        axisNum += 1
        ax = plt.subplot(6, 5, axisNum)
        color = colors[axisNum % len(colors)]
        if axisNum < len(linestyles):
            plt.plot(t, s, linestyles[axisNum], color=color, markersize=10)
        else:
            style = styles[(axisNum - len(linestyles)) % len(styles)]
            plt.plot(t, s, linestyle='None', marker=style, color=color, markersize=10)
        ax.set_yticklabels([])
        ax.set_xticklabels([])

plt.show()

```

Keywords: python, matplotlib, pylab, example, codex (see [Search examples](#))

[home](#) | [examples](#) | [gallery](#) | [pyplot](#) | [docs](#) »

[modules](#) | [index](#)