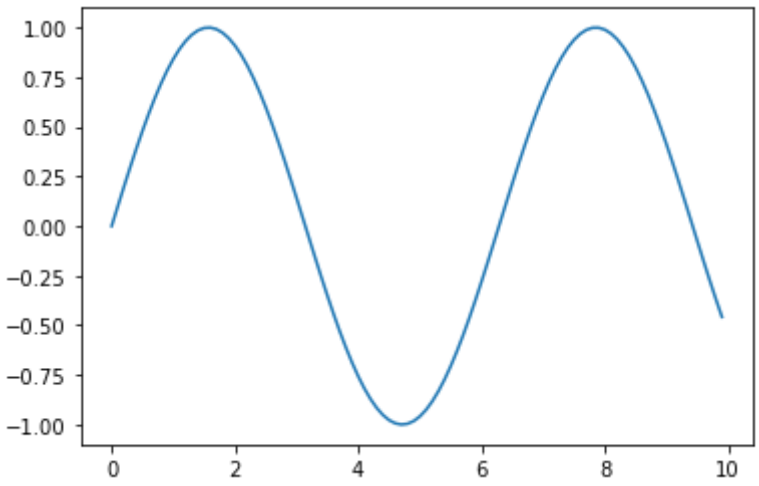
Matplotlib画图

Mattplotlib画图入门

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y = np.sin(x)

# 画出
plt.plot(x,y)
# 显示
plt.show()
```



练习，画出x在[-10, 10]的区间时，采样间隔为0.1, 0.5, 1时的下列函数图像：

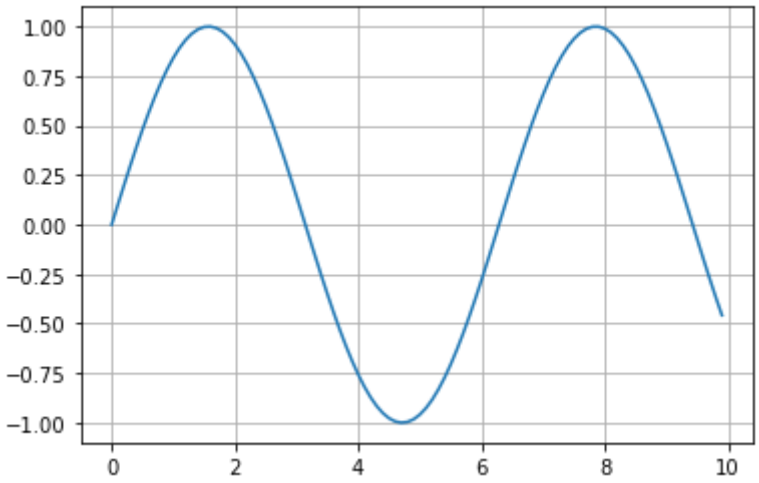
- y = 2x+10
- y = cos(x)+sin(x)
- y = e*(cos(x)+10)

In [ ]:
```
x = np.arange(-10, 10, 0.1)
y = np.exp(np.cos(x)+10)
plt.plot(x,y)
plt.show()
```

增加网格线

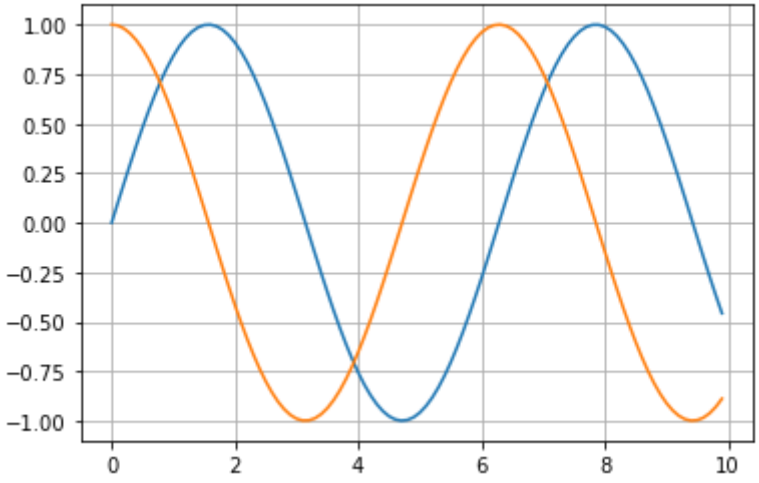
In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y = np.sin(x)

# 画出
plt.plot(x,y)
```

```
# 增加网格线
plt.grid()

# 显示
plt.show()
```



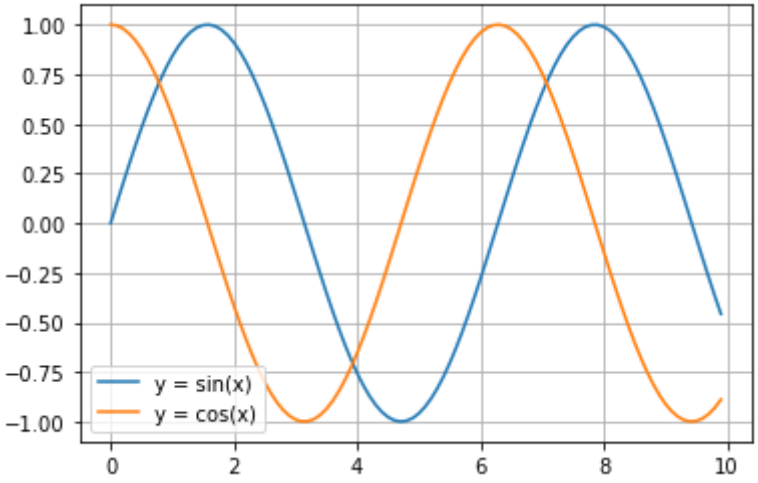
同一张Figure下多个图形

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

# 画出
plt.plot(x,y1)
plt.plot(x,y2)

# 增加网格线
plt.grid()

# 显示
plt.show()
```



图例

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

# 画出
plt.plot(x,y1, label = "y = sin(x)")
plt.plot(x,y2, label = "y = cos(x)")

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

# 显示
plt.show()
```



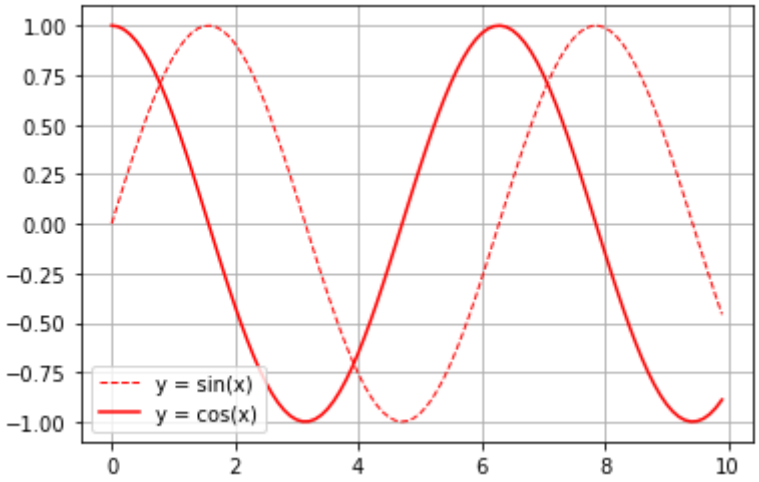
legend()的参数

- loc 位置 'best','upper/down right/left'

设置曲线样式

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

# 画出
plt.plot(x,y1, label = "y = sin(x)", color = '#FF0000', linewidth =1.0 , linestyle =
plt.plot(x,y2, label = "y = cos(x)", color = '#FF0000', linewidth = 1.5, linestyle =

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')
```

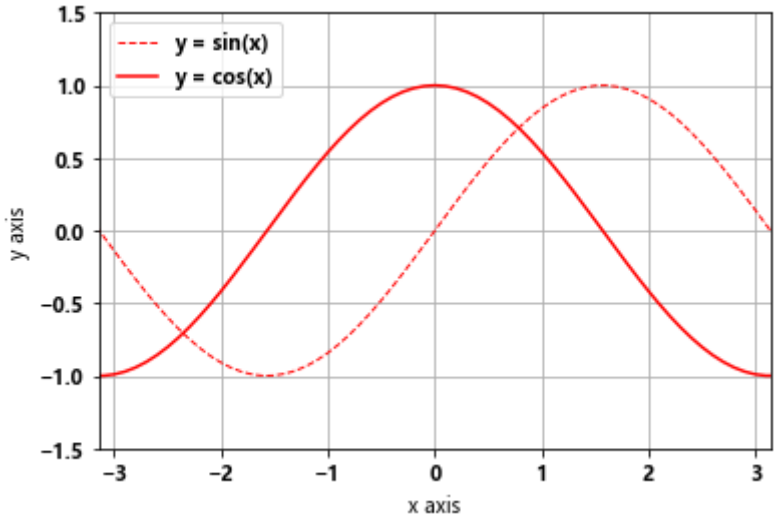
```
# 显示
plt.show()
```



设置坐标轴

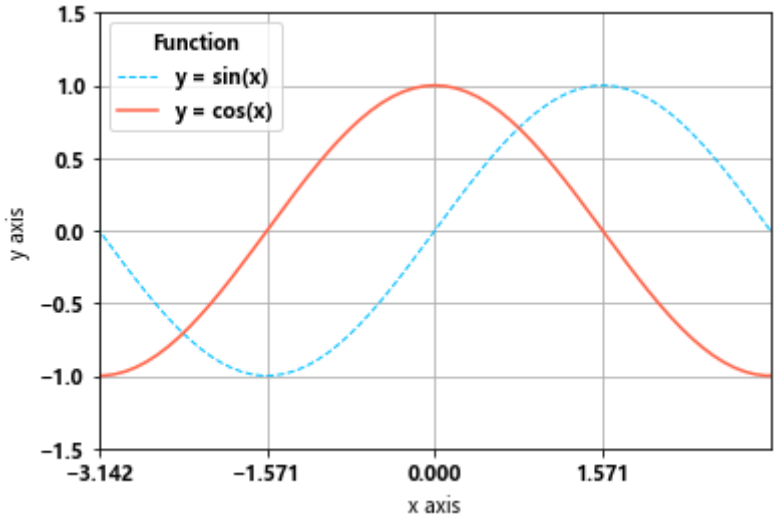
In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(-10,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

# 画出
plt.plot(x,y1, label = "y = sin(x)", color = '#FF0000', linewidth =1.0 , linestyle =
plt.plot(x,y2, label = "y = cos(x)", color = '#FF0000', linewidth = 1.5, linestyle =

# 显示坐标轴的Label
plt.xlabel("x axis")
plt.ylabel("y axis")

# 设置坐标轴的显示范围
plt.xlim((-np.pi, np.pi))
plt.ylim((-1.5, 1.5))

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

# 显示
plt.show()
```

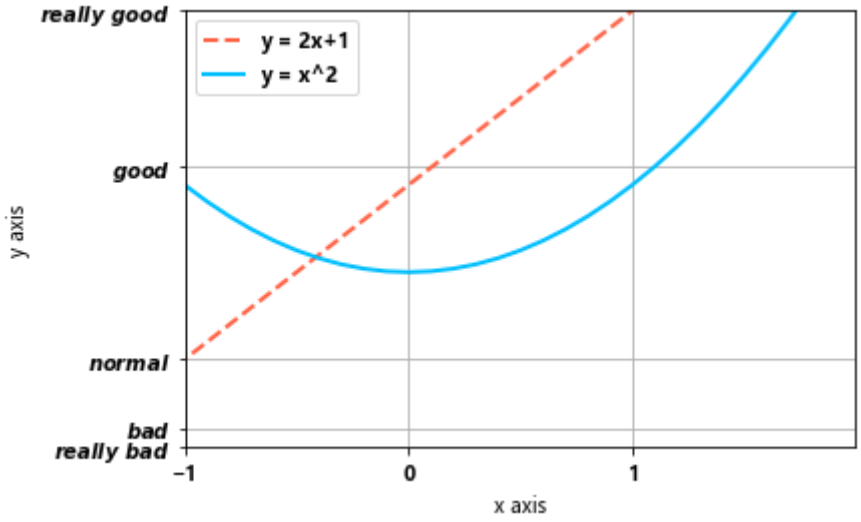


设置坐标轴的tick区间

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(-10,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

# 画出
plt.plot(x,y1, label = "y = sin(x)", color = 'deepskyblue', linewidth =1.0 , linesty
plt.plot(x,y2, label = "y = cos(x)", color = 'tomato', linewidth = 1.5, linestyle =

# 显示坐标轴的Label
plt.xlabel("x axis")
plt.ylabel("y axis")

# 设置坐标轴的显示范围
plt.xlim((-np.pi, np.pi))
plt.ylim((-1.5, 1.5))

# 设置坐标轴的tick区间
x_ticks = np.arange(-np.pi, np.pi, np.pi/2)
plt.xticks(x_ticks)
# plt.xticks(new_ticks,['-pi','-pi/2','0','pi/2'])

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best', title = "Function")

# 显示
plt.show()
```

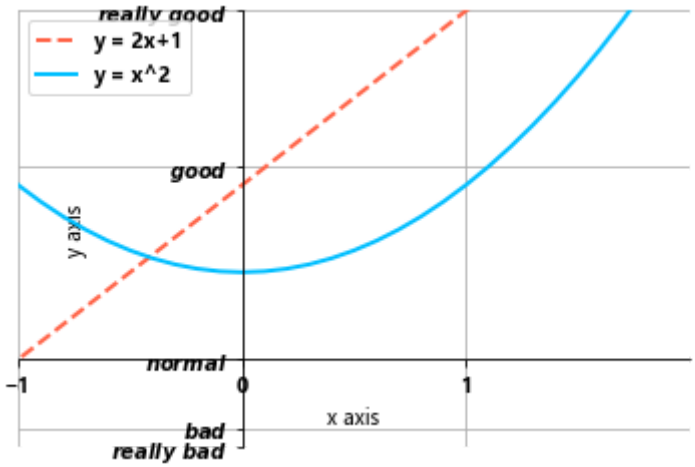


In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(-10,10,0.1)
y1 = 2*x+1
y2 = x**2

# 画出
plt.plot(x,y1, label = "y = 2x+1", color = 'tomato', linewidth =2 , linestyle = '--'
plt.plot(x,y2, label = "y = x^2", color = 'deepskyblue', linewidth = 2, linestyle =

# 显示坐标轴的Label
plt.xlabel("x axis")
plt.ylabel("y axis")

# 设置坐标轴的显示范围
plt.xlim((-1,2))
plt.ylim((-2,3))

# 设置坐标轴的tick区间
x_ticks = np.arange(-1,2,1)
plt.xticks(x_ticks)
plt.yticks([-2, -1.8, -1, 1.22, 3], ['$really\ bad$', '$bad$', '$normal$', '$good$',

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

# 显示
plt.show()
```



移动坐标轴的位置

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(-10,10,0.1)
y1 = 2*x+1
y2 = x**2

# 画出
plt.plot(x,y1, label = "y = 2x+1", color = 'tomato', linewidth =2 , linestyle = '--'
plt.plot(x,y2, label = "y = x^2", color = 'deepskyblue', linewidth = 2, linestyle =

# 显示坐标轴的Label
plt.xlabel("x axis")
plt.ylabel("y axis")

# 设置坐标轴的显示范围
plt.xlim((-1,2))
plt.ylim((-2,3))

# 设置坐标轴的tick区间
x_ticks = np.arange(-1,2,1)
plt.xticks(x_ticks)
plt.yticks([-2, -1.8, -1, 1.22, 3], ['$really\ bad$', '$bad$', '$normal$', '$good$',

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

# 获取坐标轴对象，并移动坐标轴
ax = plt.gca()
ax.spines['right'].set_color('None')
ax.spines['top'].set_color('None')
ax.spines['bottom'].set_position(('data',-1))
ax.spines['left'].set_position(('data',0))

# 显示
plt.show()
```



Annotation 与 text 标注

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(-1, 5, 0.1)
y = 2*x + 1

plt.plot(x, y, label = 'y = 2x+1')
plt.grid()
plt.legend()
plt.xlabel('x axis')
plt.ylabel('y axis')

# 标注
x_1 = 2
y_1 = 2*x_1 + 1

plt.scatter(x_1, y_1)

plt.plot([x_1, x_1],[0, y_1], linestyle = '--')

plt.annotate(f'2x+1={y_1}', xy=(x_1, y_1), xycoords='data', xytext=(+30, -30),
             textcoords='offset points', fontsize=16,
             arrowprops=dict(arrowstyle='->', connectionstyle="arc3,rad=.2"))

ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data', 0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data', 0))

plt.text(-3.7, 3, "Hello text",
         fontdict={'size': 16, 'color': 'r'})

plt.show()
```

Hello text

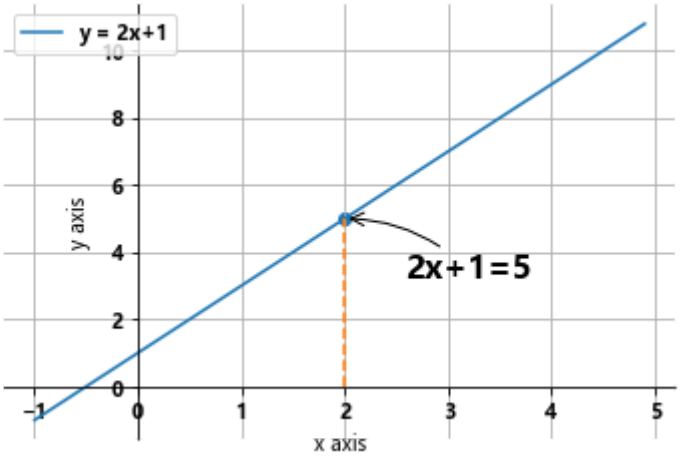
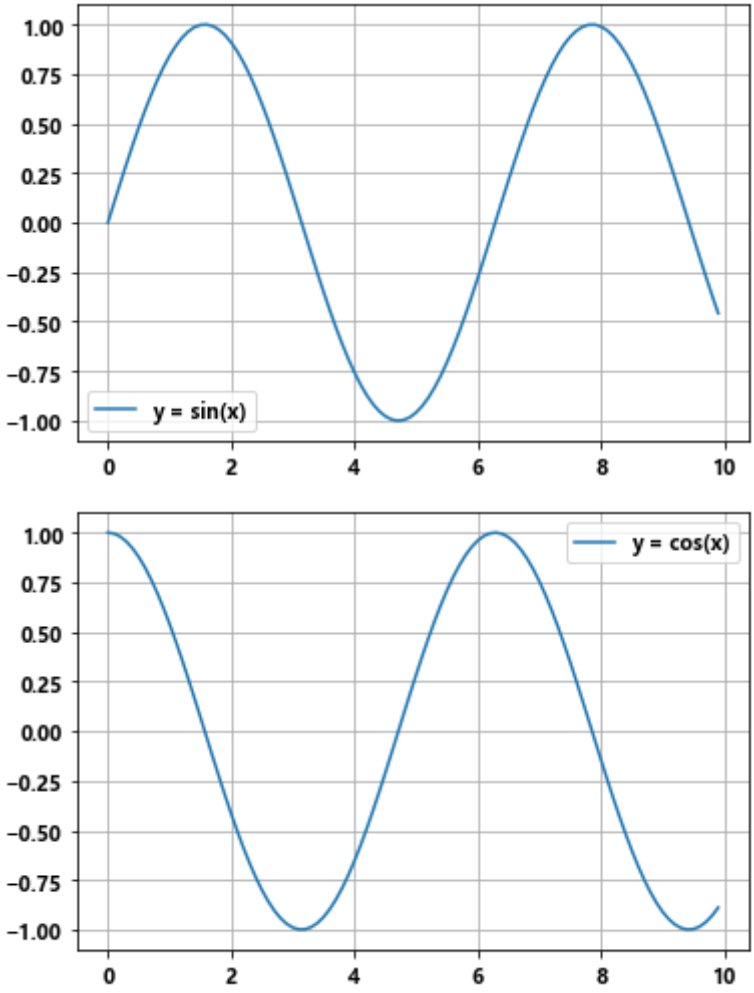


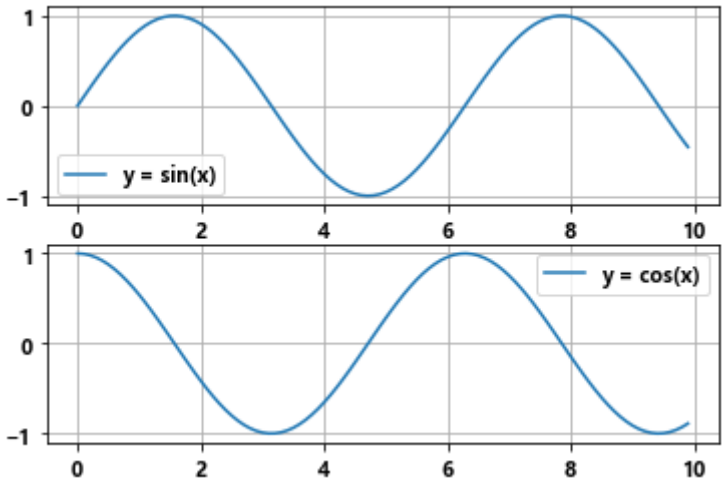
Figure 对象

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

plt.figure()
# 画出
plt.plot(x,y1, label = "y = sin(x)")

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

# 显示
plt.show()

plt.plot(x,y2, label = "y = cos(x)")

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')
plt.show()
```

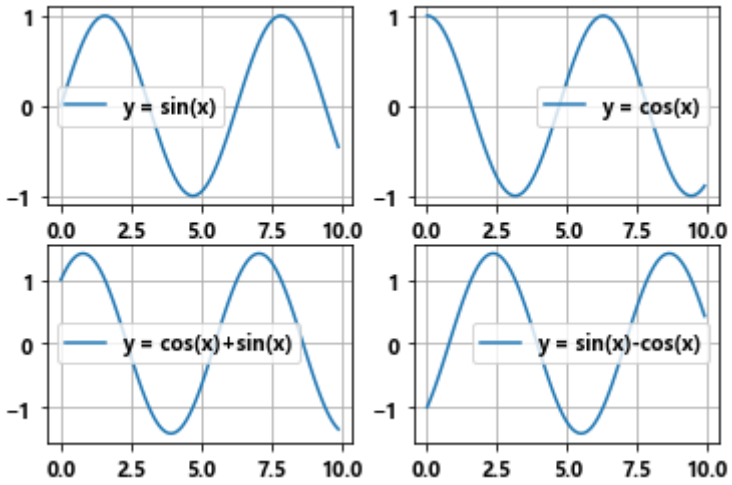


subplot 子图

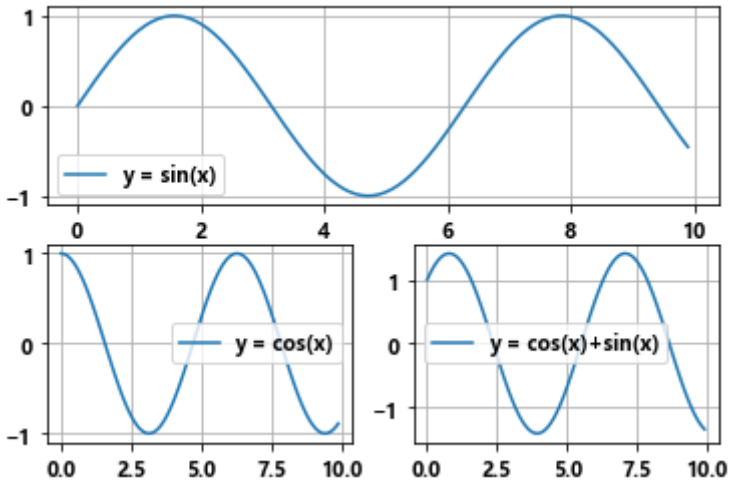
In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

plt.figure()

plt.subplot(2,1,1)
# 画出
plt.plot(x,y1, label = "y = sin(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

plt.subplot(2,1,2)
plt.plot(x,y2, label = "y = cos(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

plt.show()
```



In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = y1+y2
y4 = y1-y2

plt.figure()
plt.subplot(2,2,1)
# 画出
plt.plot(x,y1, label = "y = sin(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

plt.subplot(2,2,2)
plt.plot(x,y2, label = "y = cos(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

plt.subplot(2,2,3)
plt.plot(x,y3, label = "y = cos(x)+sin(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')


plt.subplot(2,2,4)
plt.plot(x,y4, label = "y = sin(x)-cos(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

plt.show()
```

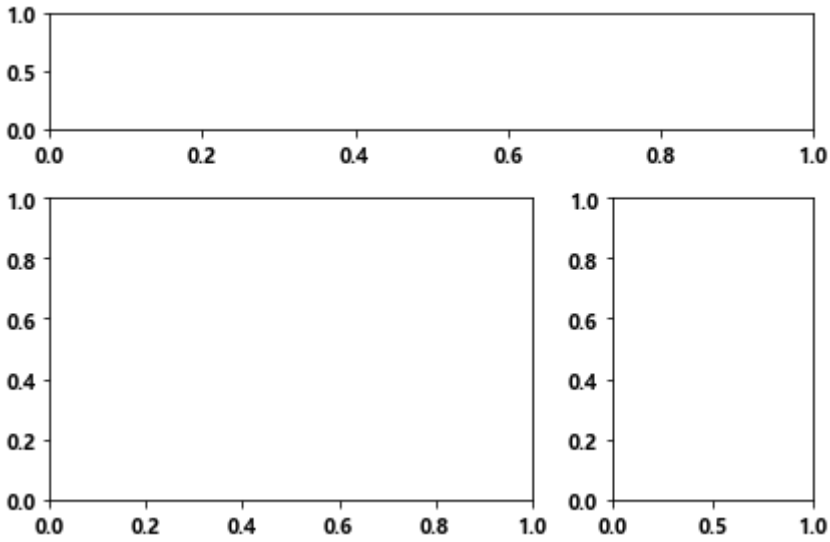


In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(0,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = y1+y2

plt.figure()
plt.subplot(2,1,1)
# 画出
plt.plot(x,y1, label = "y = sin(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

plt.subplot(2,2,3)
plt.plot(x,y2, label = "y = cos(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')

plt.subplot(2,2,4)
plt.plot(x,y3, label = "y = cos(x)+sin(x)")
# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best')


plt.show()
```



subplot2grid 子图

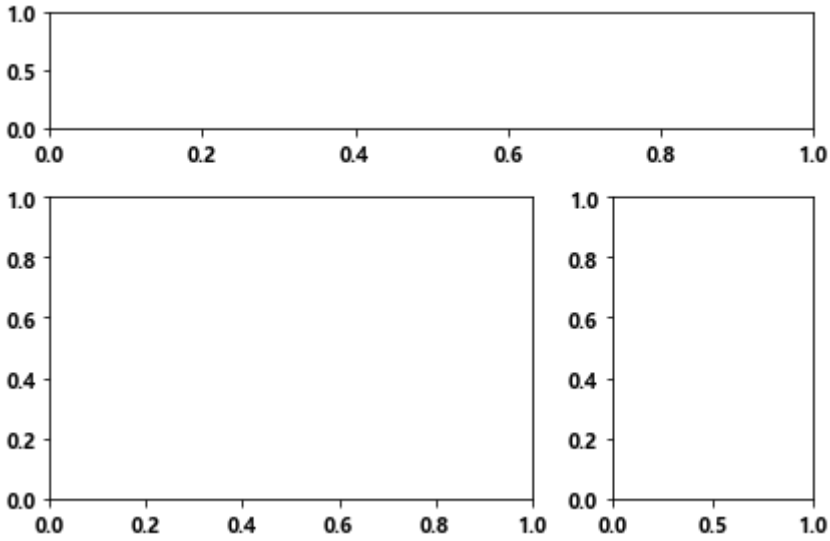
In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

plt.figure()
ax1 = plt.subplot2grid((3,3), (0,0), colspan=3, rowspan=1)
ax2 = plt.subplot2grid((3,3), (1,0), colspan=2, rowspan=2)
ax3 = plt.subplot2grid((3,3), (1,2), colspan=1, rowspan=2)

# 自适应调整子图的间距
plt.tight_layout()
plt.show()
```



gridspec 子图

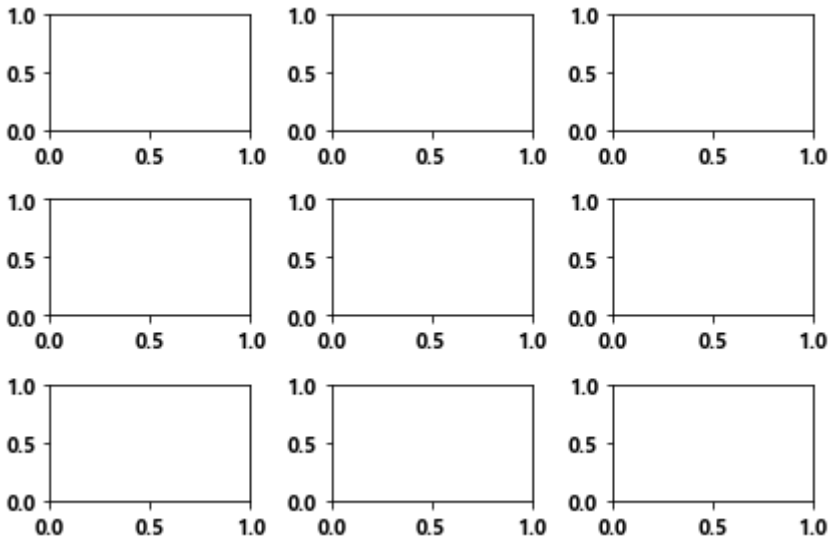
In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import numpy as np

plt.figure()
gs = gridspec.GridSpec(3,3)
ax1 = plt.subplot(gs[0,:])
ax2 = plt.subplot(gs[1:3,0:2])
ax3 = plt.subplot(gs[1:3,2])
# 自适应调整子图的间距
```

```
plt.tight_layout()
plt.show()
```



subplots子图

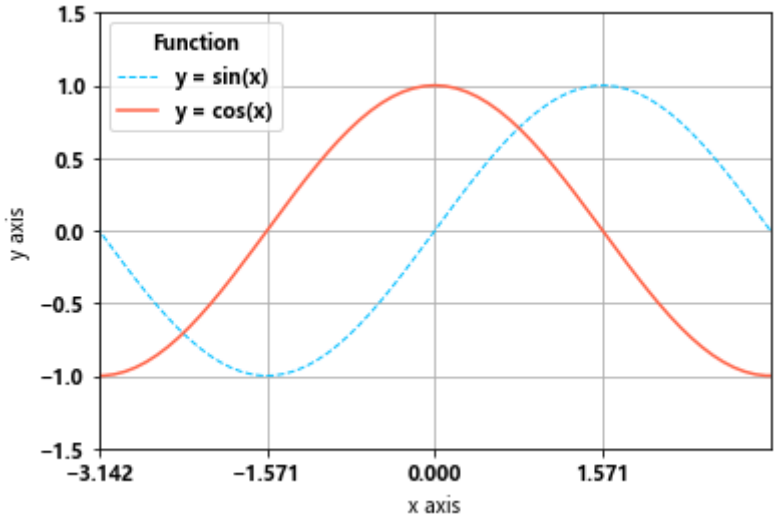
In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import numpy as np

fig, axes = plt.subplots(3,3)
# 自适应调整子图的间距
fig.tight_layout()
plt.show()
```



保存图片到本地

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 准备数据
x = np.arange(-10,10,0.1)
y1 = np.sin(x)
y2 = np.cos(x)

# 画出
```

```
plt.plot(x,y1, label = "y = sin(x)", color = 'deepskyblue', linewidth =1.0 , linesty
plt.plot(x,y2, label = "y = cos(x)", color = 'tomato', linewidth = 1.5, linestyle =

# 显示坐标轴的Label
plt.xlabel("x axis")
plt.ylabel("y axis")

# 设置坐标轴的显示范围
plt.xlim((-np.pi, np.pi))
plt.ylim((-1.5, 1.5))

# 设置坐标轴的tick区间
x_ticks = np.arange(-np.pi, np.pi, np.pi/2)
plt.xticks(x_ticks)
# plt.xticks(new_ticks,['-pi','-pi/2','0','pi/2'])

# 增加网格线
plt.grid()

# 增加图例
plt.legend(loc = 'best', title = "Function")

#保存到本地
plt.savefig('cosine.jpg')
plt.savefig('cosine.pdf')
# 显示
plt.show()
```

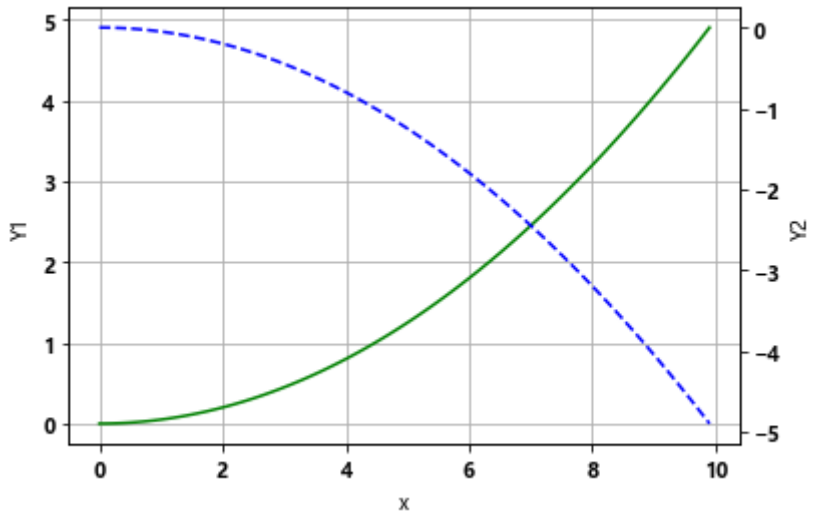


次坐标轴

In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 10, 0.1)
y1 = 0.05*x**2
y2 = -1*y1

fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.plot(x, y1, color = 'green', linestyle='-')
ax2.plot(x, y2, color = 'blue', linestyle='--')
ax1.set_xlabel('x')
ax1.set_ylabel('Y1')
ax2.set_ylabel('Y2')
```

```
ax1.grid()
plt.show()
```



In [ ]: