3.1 绘制折线图

plt.plot([x],y,[fmt],data=None,kwargs)

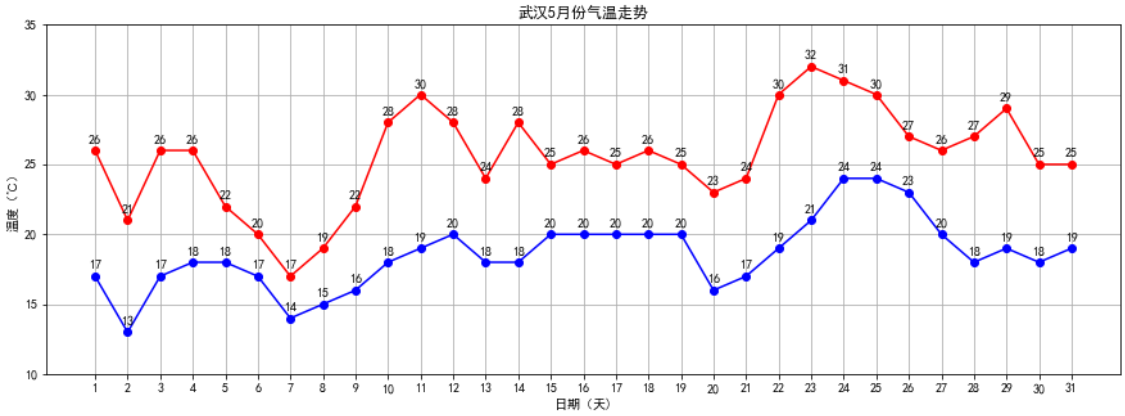
- 其中plot是一个画图的函数，他的参数为plot([x],y,[fmt],data=None,**kwargs)。
- fmt可以传一个字符串，用来给这个图做一些样式修改的。
- 默认的绘制样式是b-，也就是蓝色实体线条。

案例：

1. 以下是武汉某一个月的天气数据，按照时间的顺序绘制成折线图，其中数据highest是最高温度，lowest是最低温度。最高温度线条用红色，最低温度线条用蓝色。
2. 具体的坐标点，用圆点marker表示。
3. 在图中标注出所有点的温度值。
4. 把x轴的时间刻度按照1号-31号标记出来，并且标记x轴的标题为："日期（天）"、y轴的标题为："温度（℃）"。
5. 图的标题是“武汉5月份气温走势”。
6. 显示图例在右上角。

highest=[26,21,26,26,22,20,17,19,22,28,30,28,24,28,25,26,25,26,25,23,24,30,32,31,30,27,26,27,29,25,25]
lowest=[17,13,17,18,18,17,14,15,16,18,19,20,18,18,20,20,20,20,20,16,17,19,21,24,24,23,20,18,19,18,19]

效果图参考：



In [ ]:

In [ ]:

In [ ]:

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']
import numpy as np
import pandas as pd
```

In [ ]:

```
#定义两个列表，分别存放highest和lowest
highest = [26,21,26,26,22,20,17,19,22,28,30,28,24,28,25,26,25,26,25,23,24,30,32,31,30,27,26,27,29,25
lowest =  [17,13,17,18,18,17,14,15,16,18,19,20,18,18,20,20,20,20,20,16,17,19,21,24,24,23,20,18,19,18

#绘图前先设置好画布的大小
plt.figure(figsize=(15,5))

#绘制最高温度折线，并设置线条颜色和标记点
plt.plot(highest,marker='o',color='r')

#绘制最低温度折线，并设置线条颜色和标记点
plt.plot(_____,_______,_______)

# 设置x轴的坐标和标题
plt.xticks(range(31),range(1,32))
plt.xlabel("日期（天)")

# 设置y轴的坐标和文本
plt.yticks(___________,___________)
plt.ylabel(__________)

# 设置文本标记
for x in range(31):
    temp1=highest[x]
    temp2=lowest[x]
    plt.annotate(temp1,xy=(x,temp1),xytext=(x-0.3,temp1+0.5))
    plt.annotate(temp2,xy=(x,temp2),xytext=(x-0.3,temp2+0.5))


# 设置标题
plt._____(_______)

#设置网格
plt._____()

#显示图片
plt.show()
```

笔记：

1. plt.annotate(text,xy,xytext,arrowprops)是用来做文本注释的

- text代表的是需要注释的文本
- xy代表的是需要注释的坐标点
- xytext代表的是文本的坐标点。

2.xticks、yticks与xlabel、ylabel的区别

- xticks、yticks是指的坐标轴的刻度
  - 其中plt.xticks()中的第一个参数是用来划分刻度，第二个参数是用来指定展示的刻度值。
- xlabel、ylabel代表的是坐标的标题。

In [ ]:

In [ ]:

```
```

3.2 绘制多个子图

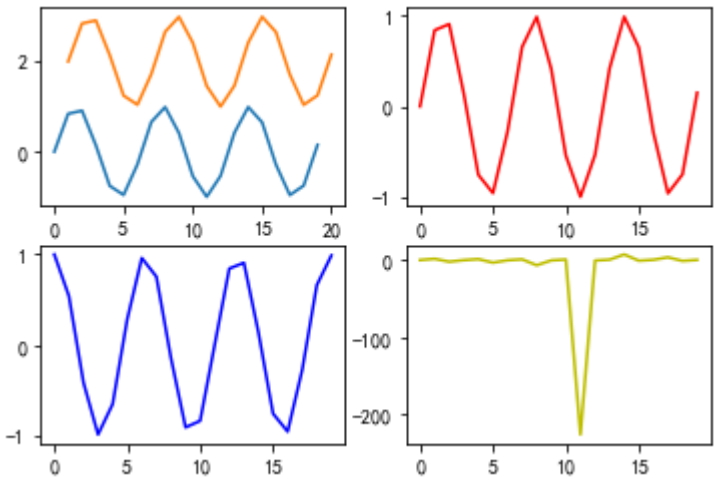
绘制子图的时候，我们可以使用plt.subplot或plt.subplots来实现。

In [6]:

```
#方法1：plt.subplot(行数，列数，第几个图)
plt.rcParams["axes.unicode_minus"] = False    # 使坐标轴刻度表签正常显示正负号
x=np.arange(20)
y=np.sin(x)
plt.figure()
plt.subplot(221)
plt.plot(x,y,x+1,y+2)
plt.subplot(222)
plt.plot(np.sin(x),'r')
plt.subplot(223)
plt.plot(np.cos(x),'b')
plt.subplot(224)
plt.plot(np.tan(x),'y')
```

Out[6]:

```
[<matplotlib.lines.Line2D at 0x1d544c96220>]
```

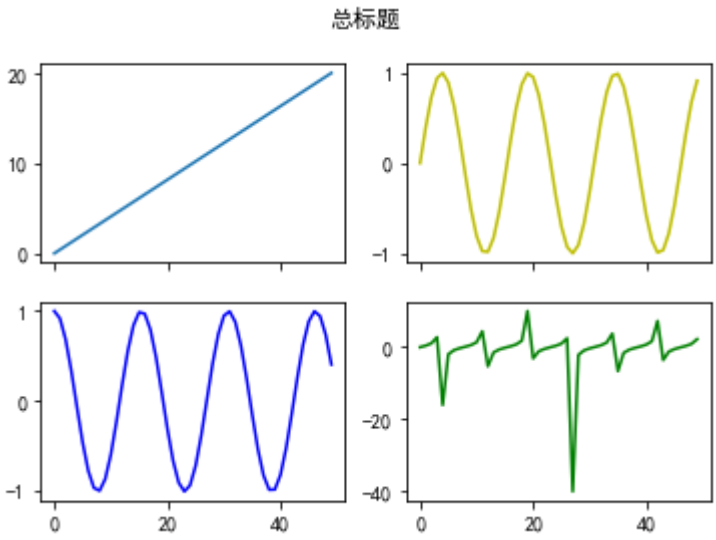


In [68]:

```
#方法2：plt.subplots(2,2)
values = np.linspace(0,20)
fig,axes = plt.subplots(2,2,sharex=True)
fig.suptitle("总标题")
ax1 = axes[0,0]
ax1.plot(values)
ax2 = axes[0,1]
ax2.plot(np.sin(values),c='y')
ax3 = axes[1,0]
ax3.plot(np.cos(values),c='b')
ax4 = axes[1,1]
ax4.plot(np.tan(values),c='g')
```

Out[68]:

```
[<matplotlib.lines.Line2D at 0x1d544f99d60>]
```



子图小结：

1. 如果想要在一个图上绘制多根折线，那么直接在plot中传递多个x和y就可以了，或者调用多次plot方法即可。
2. plt.ylim((start,end))：用来修改y轴的取值范围，同理x轴用plt.xlim()
3. plt.legend(loc='lower left'):loc参数表示图例放置位置（绘图函数需加入label参数标注）：
4. 如果想要在一个画板上绘制多个子图，那么可以使用plt.subplot或plt.subplots方法。
   - plt.subplot：plt.subplot(221)，第一个数字代表行，第二个数字代表列，第三个数字代表当前第几个子图。使用了subploth后绘制的所操作都是在当前子图上，直到出现了新的subplot。
   - plt.subplots：fig,axes = plt.subplots(2,2,sharex=True,sharey=True)，第一个代表行，第二个代表列，这个方法是先一次性绘制所有空子图，通过axes索引到各自子图对象，再对子图对象进行操作。

In [ ]: