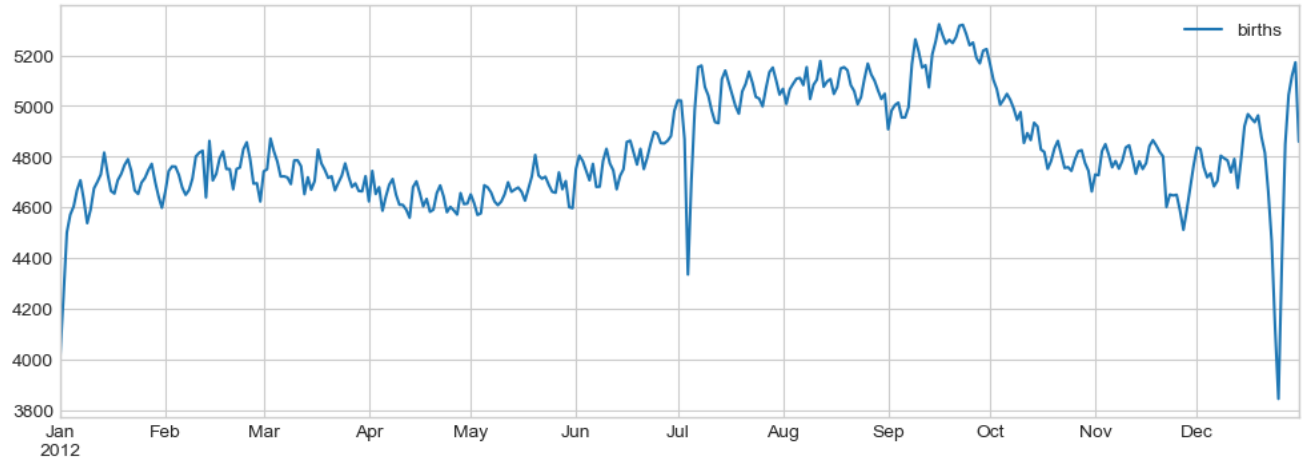
In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib as mpl
plt.style.use('seaborn-whitegrid')
import numpy as np
import pandas as pd
```

In [26]:
```
births=pd.read_csv("births.csv")
quartiles=np.percentile(births["births"],[25,50,75])#quartitles means dividing data into 4 segments
mu,sig=quartiles[1],0.74 *(quartiles[2]-quartiles[0])#mu is for mean and 0.74 correction factor
births=births.query('(births >@mu-5*@sig) & (births<@mu + 5*@sig)')#brith range calculation using mean and sigma
births["day"]=births["day"].astype(int)#making birth column data as integer
births.index=pd.to_datetime(10000 * births.year +100 * births.month +births.day,format='%Y%m%d')
births_by_date=births.pivot_table("births",[births.index.month, births.index.day])#grouping using pivot tables by month and day
births_by_date.index=[pd.datetime(2012, month, day) for (month, day) in births_by_date.index]
fig,ax=plt.subplots(figsize=(12, 4))
births_by_date.plot(ax=ax)
```

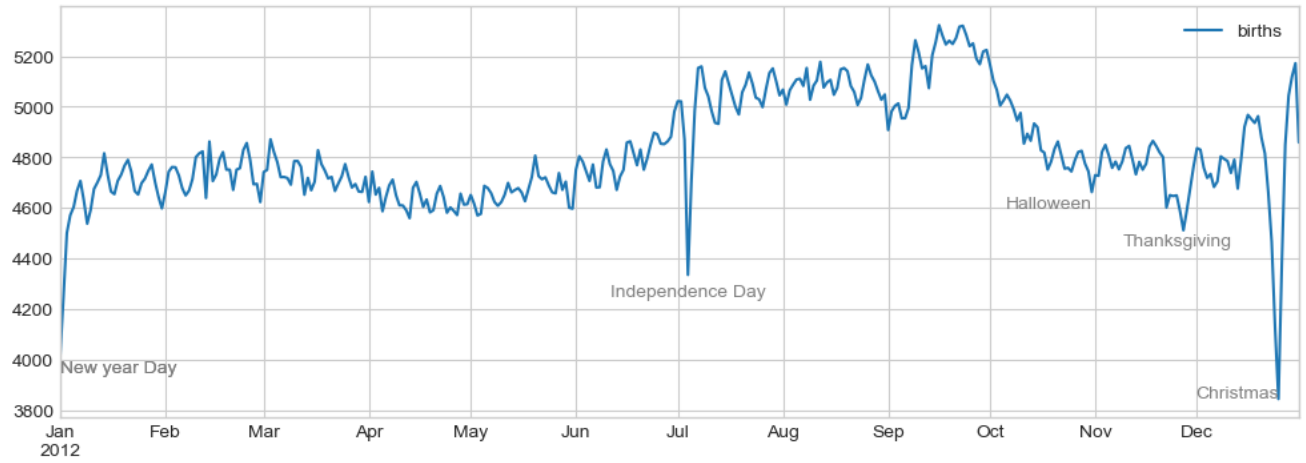
```
C:\Users\kdmag\AppData\Local\Temp\ipykernel_16920\911539569.py:8: FutureWarning: The pandas.datetime class is deprecated and wi
ll be removed from pandas in a future version. Import from datetime module instead.
  births_by_date.index=[pd.datetime(2012, month, day) for (month, day) in births_by_date.index]
```

Out[26]: <AxesSubplot:>



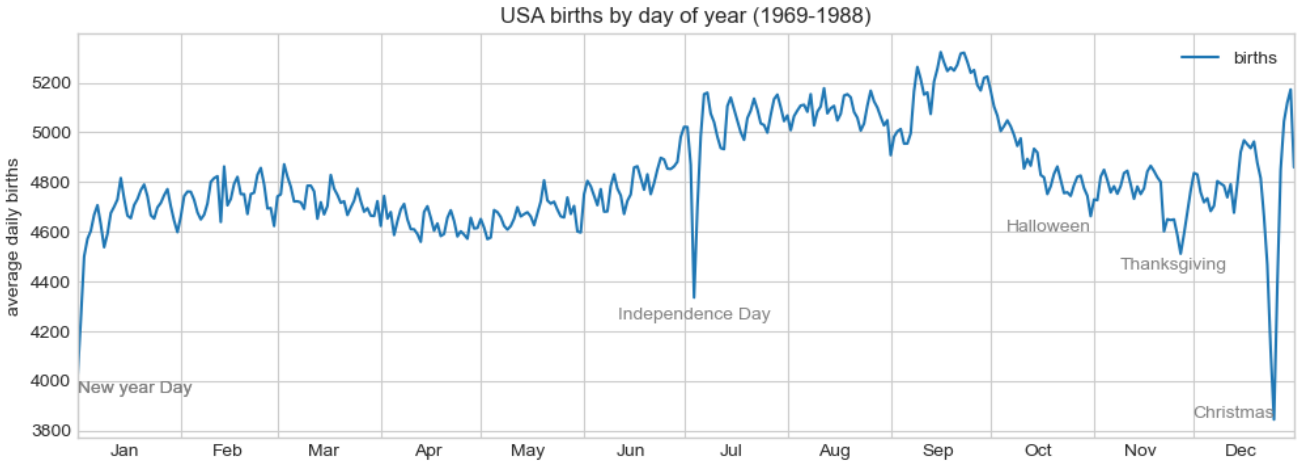
In [35]:
```
#for adding text we can use ax.text or plt.text
fig,ax=plt.subplots(figsize=(12, 4))
births_by_date.plot(ax=ax)
style=dict(size=10,color="gray")
ax.text("2012-1-1",3950,"New year Day",**style)
ax.text("2012-7-4",4250,"Independence Day",ha="center",**style)
ax.text("2012-1-1",3950,"New year Day",**style)
ax.text("2012-10-31",4600,"Halloween",ha="right",**style)
ax.text("2012-11-25",4450,"Thanksgiving",ha="center",**style)
ax.text("2012-12-25",3850,"Christmas",ha="right",**style)
```

Out[35]: Text(2012-12-25, 3850, 'Christmas')



In [38]:
```
ax.set(title="USA births by day of year (1969-1988)",ylabel="average daily births")
# Format the x axis with centered month labels
ax.xaxis.set_major_locator(mpl.dates.MonthLocator())
ax.xaxis.set_minor_locator(mpl.dates.MonthLocator(bymonthday=15))
ax.xaxis.set_major_formatter(plt.NullFormatter())
ax.xaxis.set_minor_formatter(mpl.dates.DateFormatter('%h'))
fig
```

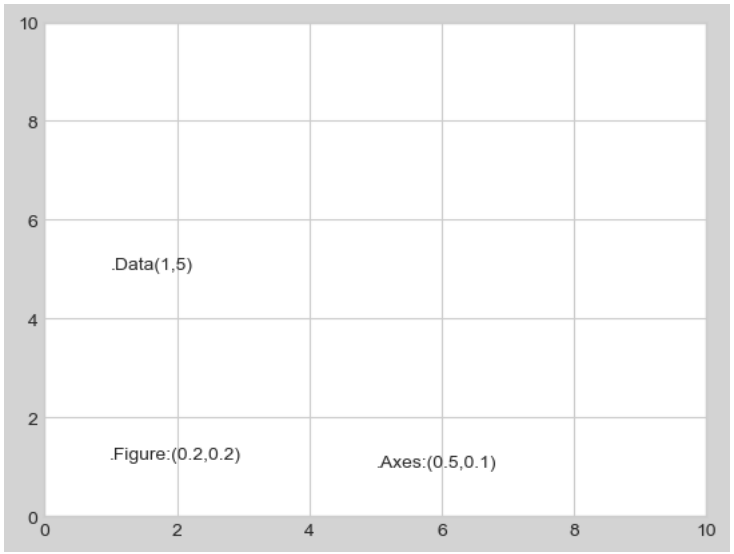
Out[38]:



In [40]:
```
#transforms
#ax.transData(transforms based on data coordinates)
#ax.transAxes(transforms based on axes)
#fig.transFigure(transforms based on figure)
```

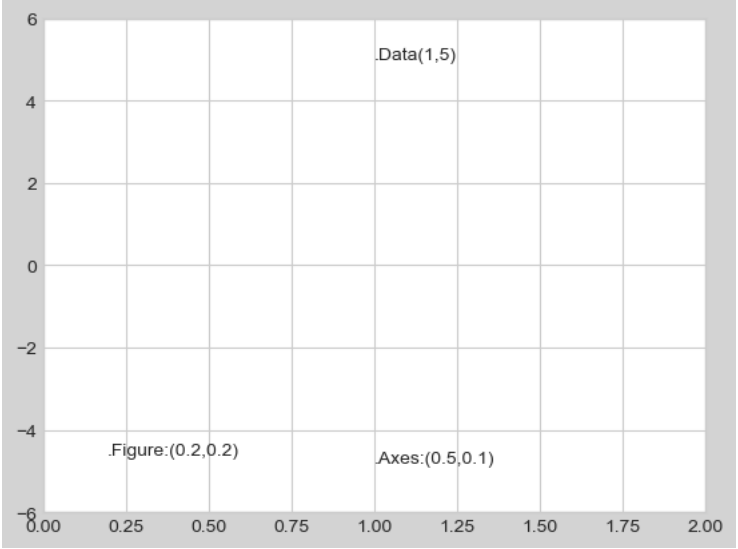
In [43]:
```
fig,ax=plt.subplots(facecolor='lightgray')
ax.axis([0,10,0,10])
#transform=ax.transData is defualt
ax.text(1,5,".Data(1,5)",transform=ax.transData)
ax.text(0.5,0.1,".Axes:(0.5,0.1)",transform=ax.transAxes)#transAxes give location as fraction of axes length
ax.text(0.2,0.2,".Figure:(0.2,0.2)",transform=fig.transFigure)#trnasFigure is similar to axes but from bottom left of figure
```

Out[43]:
```
Text(0.2, 0.2, '.Figure:(0.2,0.2)')
```



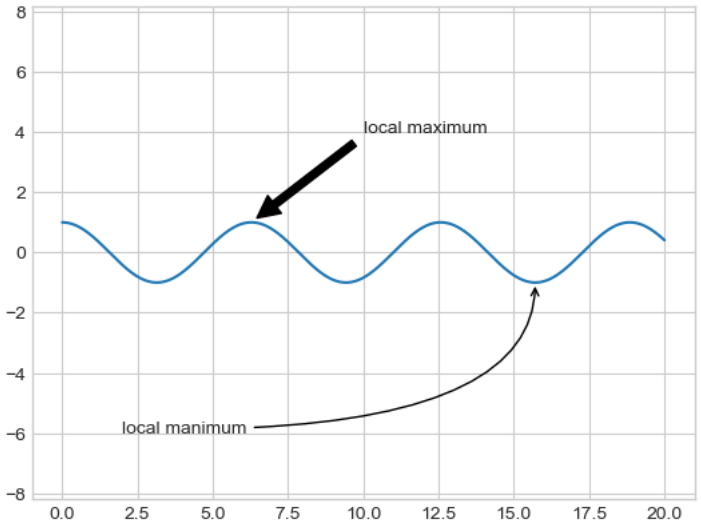
In [44]:
```
#changing axes limit
ax.set_xlim(0,2)
ax.set_ylim(-6,6)
fig#only tarnsData is changed
```

Out[44]:



In [54]:
```
#arrows and annotation
%matplotlib inline
fig,ax=plt.subplots()
x=np.linspace(0,20,1000)
ax.plot(x,np.cos(x))
ax.axis("equal")
ax.annotate("local maximum",xy=(6.28,1),xytext=(10,4),arrowprops=dict(facecolor="black",shrink=0.05))
ax.annotate("local manimum",xy=(5*np.pi,-1),xytext=(2,-6),arrowprops=dict(arrowstyle="->",
                                                                          connectionstyle="angle3,angleA=0,angleB=-90"))
```

Out[54]:
```
Text(2, -6, 'local manimum')
```



In [67]:
```
#all options in arrows
fig,ax=plt.subplots(figsize=(12, 4))
births_by_date.plot(ax=ax)
ax.annotate("New Year's Day",xy=('2012-1-1', 4100),  xycoords='data',xytext=(50, -30), textcoords='offset points',
arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=-0.2"))
ax.annotate("Independence Day",xy=("2012-7-4",4250),xytext=(10,-40),xycoords="data",textcoords="offset points",
            bbox=dict(boxstyle="round",fc="none",ec="gray"),ha="center",arrowprops=dict(arrowstyle="->"))
ax.annotate('', xy=('2012-9-1', 4850), xytext=('2012-9-7', 4850),
            xycoords='data', textcoords='data',
            arrowprops={'arrowstyle': '|-|,widthA=0.2,widthB=0.2', })
ax.annotate('Halloween', xy=('2012-10-31', 4600),  xycoords='data',
            xytext=(-80, -40), textcoords='offset points',
            arrowprops=dict(arrowstyle="fancy",
                            fc="0.6", ec="none",
                            connectionstyle="angle3,angleA=0,angleB=-90"))
ax.annotate('Thanksgiving', xy=('2012-11-25', 4500),  xycoords='data',
            xytext=(-120, -60), textcoords='offset points',
            bbox=dict(boxstyle="round4,pad=.5", fc="0.9"),
            arrowprops=dict(arrowstyle="->",
                            connectionstyle="angle,angleA=0,angleB=80,rad=20"))
ax.annotate('Christmas', xy=('2012-12-25', 3850),  xycoords='data',
             xytext=(-30, 0), textcoords='offset points',
             size=13, ha='right', va="center",
             bbox=dict(boxstyle="round", alpha=0.1),
             arrowprops=dict(arrowstyle="wedge,tail_width=0.5", alpha=0.1))
```

Out[67]:
```
Text(-30, 0, 'Christmas')
```