In [4]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('births.csv')
```

In [5]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15547 entries, 0 to 15546
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   year    15547 non-null  int64  
 1   month   15547 non-null  int64  
 2   day     15067 non-null  float64
 3   gender  15547 non-null  object 
 4   births  15547 non-null  int64  
dtypes: float64(1), int64(3), object(1)
memory usage: 607.4+ KB
```

In [6]:

```python
df.isnull().sum()
```

Out[6]:

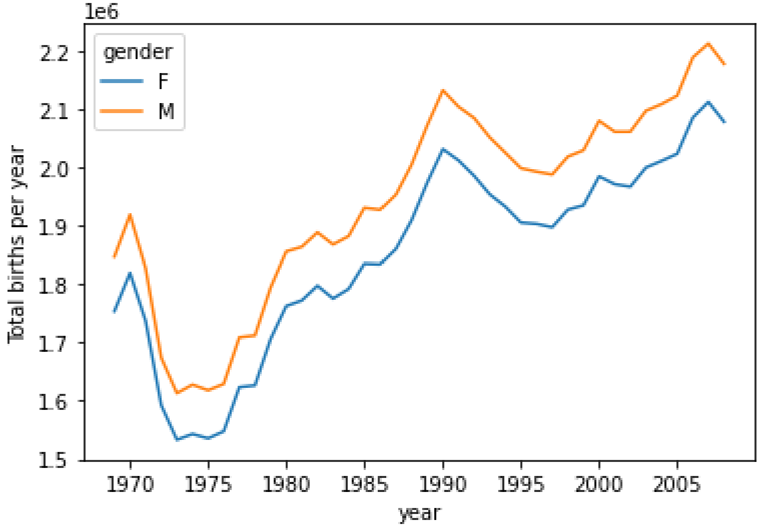
```
year        0
month       0
day       480
gender      0
births      0
dtype: int64
```

In [11]:

```python
# i) Total number of US births by year and gender
total_births_by_year_and_gender = df.groupby(['year', 'gender'])['births'].sum()
print(total_births_by_year_and_gender)

total = df.pivot_table('births', index='year', columns='gender', aggfunc='sum').plot()
plt.ylabel('Total births per year')
```

```
year  gender
1969  F         1753634
      M         1846572
1970  F         1819164
      M         1918636
1971  F         1736774
                 ...
2006  M         2188268
2007  F         2111890
      M         2212118
2008  F         2077929
      M         2177227
Name: births, Length: 80, dtype: int64
```

```
Out[11]: Text(0, 0.5, 'Total births per year')
```



```
In [5]: df.day.unique()
```

```
Out[5]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12., 13.,
               14., 15., 16., 17., 18., 19., 20., 21., 22., 23., 24., 25., 26.,
               27., 28., 29., 30., 31., 99., nan])
```

```
In [6]: df = df[(df.day>=1) & (df.day<=31)]
```

```
In [7]: df['day'].unique()
```

Out[7]:

```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12., 13.,
       14., 15., 16., 17., 18., 19., 20., 21., 22., 23., 24., 25., 26.,
       27., 28., 29., 30., 31.])
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
year      0
month     0
day       0
gender    0
births    0
dtype: int64
```

In [24]:

```
# Convert to datetime
df['date'] = pd.to_datetime(df[['year', 'month', 'day']],format='%Y%m%d',errors='coerce')

# Get the day of the week as a numerical value (Monday=0, Sunday=6)
df['day_of_week_num'] = df['date'].dt.dayofweek

# Create decade column
df['decade'] = (df['year'] // 10) * 10
```

In [25]:

```
df.head()
```

Out[25]:

	year	month	day	gender	births	date	day_of_week_num	decade	day_of_week
0	1969	1	1.0	F	4046	1969-01-01	2.0	1960	2.0
1	1969	1	1.0	M	4440	1969-01-01	2.0	1960	2.0
2	1969	1	2.0	F	4454	1969-01-02	3.0	1960	3.0
3	1969	1	2.0	M	4548	1969-01-02	3.0	1960	3.0
4	1969	1	3.0	F	4548	1969-01-03	4.0	1960	4.0

In [26]:

```
df.isnull().sum()
```

Out[26]:

```
year                0
month               0
day               480
gender              0
births              0
date              937
day_of_week_num   937
decade              0
day_of_week       937
dtype: int64
```

In [27]:

```
df.dropna(inplace=True)
```

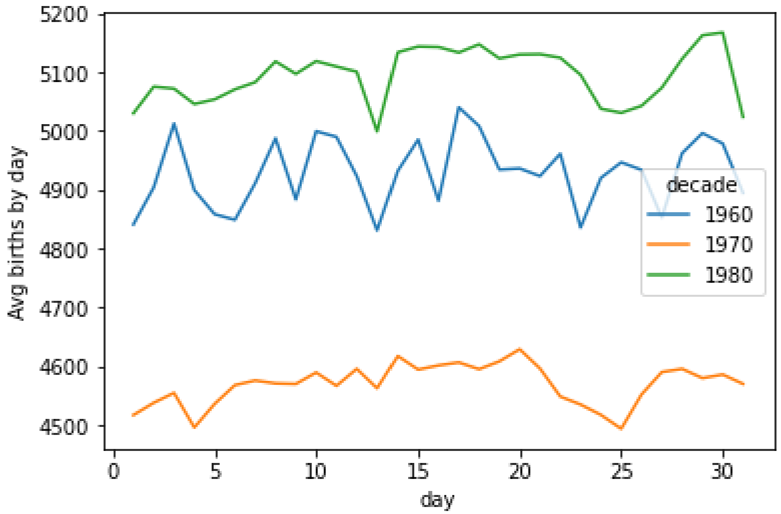
In [28]:

```
# ii) Average daily births by day of week and decade

avg_daily_births = df.groupby(['decade', 'day_of_week_num'])['births'].mean()
print(avg_daily_births)


df.pivot_table('births', index='day', columns='decade', aggfunc='mean').plot()
plt.ylabel('Avg births by day')
```

```
decade  day_of_week_num
1960    0.0                5063.826923
        1.0                5286.096154
        2.0                5074.622642
        3.0                4978.288462
        4.0                5107.884615
        5.0                4651.057692
        6.0                4342.346154
1970    0.0                4689.097701
        1.0                4885.252399
        2.0                4750.376200
        3.0                4696.923372
        4.0                4782.095785
        5.0                4207.784483
        6.0                3979.278736
1980    0.0                5276.907249
        1.0                5503.842553
        2.0                5367.642553
        3.0                5333.485106
        4.0                5393.087234
        5.0                4483.901064
        6.0                4308.120469
Name: births, dtype: float64
```

Out[28]:

```
Text(0, 0.5, 'Avg births by day')
```



In [ ]: