Birth Rate Analysis using Python

*Background

Birth rate is a demographic indicator and an essential component of population dynamics and is commonly used in demographic analysis, social planning, and policymaking. Birth rate analysis is critical for understanding population dynamics, predicting future trends, and formulating policies and interventions related to healthcare, education, infrastructure, and social welfare. It provides valuable insights into the demographic makeup of a population and helps societies make informed decisions for sustainable development and well-being.

*Objective

To analyze and visualize birth rates.

*Data Source

Birth rate data was downloaded from the Centers for Disease Control (CDC) via https://raw.githubusercontent.com/amankharwal/Birthrate-Analysis/master/births.csv

In [75]:
```
# Import libraries

import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [76]:
```
# Load dataset

Births = pd.read_csv("births.csv")
print(Births.head())
```

```
   Year  Month  Day Gender  Births
0  1969      1  1.0      F    4046
1  1969      1  1.0      M    4440
2  1969      1  2.0      F    4454
3  1969      1  2.0      M    4548
4  1969      1  3.0      F    4548
```

*Data Preparation

In [77]:
```
# Replace missing or null values

Births['Day'].fillna(0, inplace=True)
```

In [78]:
```
# Convert series data to integer

Births['Day'] = Births['Day'].astype(int)
```

*Exploratory Data Analysis (EDA)

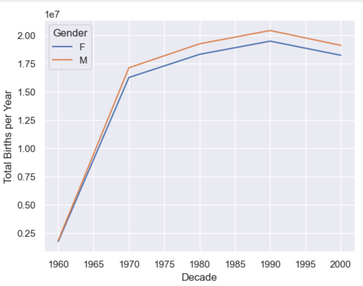
In [79]:
```
# Create additional column and name it as decade
# Use the appropriate formula to compute for the births per decade

Births['Decade'] = 10 * (Births['Year'] // 10)
Births.pivot_table('Births', index='Decade', columns='Gender', aggfunc='sum')
print(Births.head())
```

```
   Year  Month  Day Gender  Births  Decade
0  1969      1    1      F    4046    1960
1  1969      1    1      M    4440    1960
2  1969      1    2      F    4454    1960
3  1969      1    2      M    4548    1960
4  1969      1    3      F    4548    1960
```

From this initial exploration, it appears that male births outnumber female births every decade. Let's use the built-in plotting tools in Pandas to visualize the total number of births by year to see clearly.

In [80]:
```
# Import libraries

import matplotlib.pyplot as plt
import seaborn as sns
```

In [81]:
```
# Plot the total number of births per year

sns.set()
Birth_Decade = Births.pivot_table('Births', index='Decade', columns='Gender', aggfunc='sum')
Birth_Decade.plot()
plt.ylabel("Total Births per Year")
plt.show()
```



This plot confirms that male births outnumber female births every decade.

*Data Cleaning

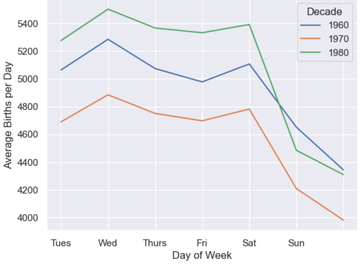
In [82]:
```
# Remove outliers (mistyped dates or missing values)

import numpy as np

quartiles = np.percentile(Births['Births'], [25, 50, 75])
mu = quartiles[1]
sig = 0.74 * (quartiles[2] - quartiles[0]) # Robust estimate of the sample mean, where the 0.74 comes from the
                                           # interquartile range of a Gaussian distribution
```

In [83]:
```
# Use query() method to filter rows with births outside these values

Births = Births.query('(Births > @mu - 5 * @sig) & (Births < @mu + 5 * @sig)')
Births.index = pd.to_datetime(10000 * Births.Year + 100 * Births.Month + Births.Day,
                              format='%Y%m%d')
Births['Day of Week'] = Births.index.dayofweek
```

*Data Visualization

Using this, let's plot births by weekday for different decades

In [84]:
```
# Plot the mean births per day for different decades

Births.pivot_table('Births', index='Day of Week', columns='Decade', aggfunc='mean').plot()
plt.gca().set_xticklabels(['Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun'])
plt.ylabel('Average Births per Day');
plt.show()
```



Apparently births are lower on weekends than weekdays

Note that the 1990s and 2000s are missing because the CDC data contains only the month of birth starting in 1989

Another interesting view is to plot the mean number of births daily each year

In [85]:
```
# First, group the data separately by month and day

Births_Month = Births.pivot_table('Births', [Births.index.month, Births.index.day])
print(Births_Month.head())
```

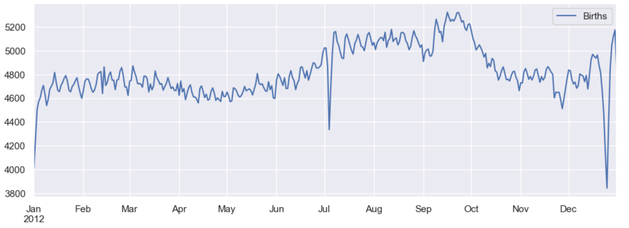
```
       Births
1 1  4009.225
  2  4247.400
  3  4500.900
  4  4571.350
  5  4603.625
```

In [86]:
```
# Place the year, month, and day before each births

Births_Month.index = [pd.datetime(2012, month, day) for (month, day) in Births_Month.index]
print(Births_Month.head())
```

```
              Births
2012-01-01  4009.225
2012-01-02  4247.400
2012-01-03  4500.900
2012-01-04  4571.350
2012-01-05  4603.625
```

Focusing on the month and day only, we now have a timeseries data reflecting the average number of births by date per year

In [87]:
```
# Plot the average number of births by date per year

fig, ax = plt.subplots(figsize=(12, 4))
Births_Month.plot(ax=ax)
plt.show()
```



End