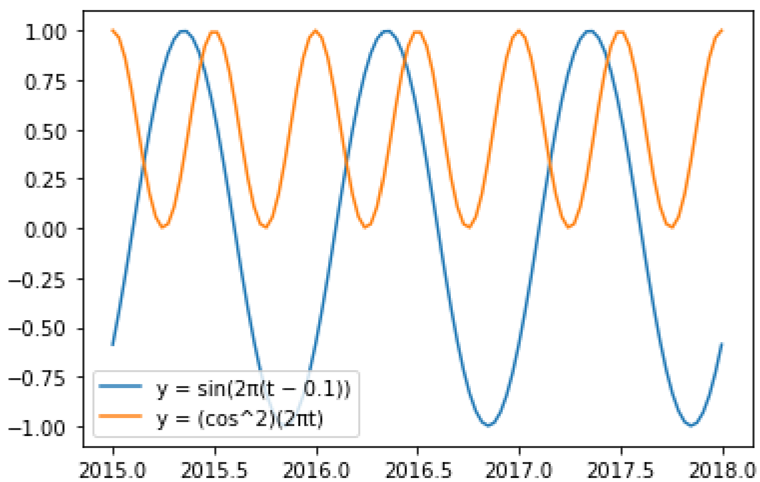
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.linear_model import LinearRegression
import os
os.chdir(r"C:\Users\camiu\M336\MATH 336 FOLDER(shen)\MATH336 (SHEN)")
```

2.1

In [2]:
```
# Define the data sequence
t = np.linspace(2015, 2018, 100)

#functions
y1 = np.sin(2 * np.pi * (t - 0.1))
y2 = np.cos(2 * np.pi * t) ** 2

#plot the two functions
plt.plot(t, y1, label='y = sin(2π(t - 0.1))')
plt.plot(t, y2, label='y = (cos^2)(2πt)')

plt.legend()
plt.show()
```



2.4

In [3]:
```
A = np.array([[-3, 2, 1], [-2, -1, 1], [2, 1, -4]])
b = np.array([1, 2, 0])
x = np.linalg.solve(A, b)
print(x)
```

```
[-1.         -0.66666667 -0.66666667]
```

2.5

In [4]:
```
df = pd.read_csv("CA042239T.csv")
(df)
```

Out[4]:

	State_id	YEAR	Month	TMAX (F)	TMEAN (F)	TMIN (F)
0	'042239'	1887	1	.	.	.
1	'042239'	1887	2	.	.	.
2	'042239'	1887	3	.	.	.
3	'042239'	1887	4	.	.	.
4	'042239'	1887	5	.	.	.
...	...	...	...	...	...	...
1531	'042239'	2014	8	83.2	68.3	53.4
1532	'042239'	2014	9	82.5	66.7	50.8
1533	'042239'	2014	10	75.4	58.6	41.8
1534	'042239'	2014	11	62.3	49.1	35.8
1535	'042239'	2014	12	52.4	42.2	32

1536 rows × 6 columns

In [5]:
```
1536/12
```
Out[5]:
```
128.0
```

In [6]:
```
tmax = df['TMAX (F)'].values
tmax
```
Out[6]:
```
array(['.', '.', '.', ..., '75.4', '62.3', '52.4'], dtype=object)
```

In [7]:
```
df.iloc[1524,3]
```
Out[7]:
```
'60.2'
```

In [8]:
```
#df = pd.read_csv("CA042239T.csv")

# Extract the Tmax, Tmin, and Tmean columns
tmax = df['TMAX (F)'].values

columns = 12
rows = 128


# Reshape the Tmax data into a matrix with each row as a year and each column as a mor
tmax_matrix = tmax.reshape(rows, columns)

print(tmax_matrix)
```
```
[['.' '.' '.' ... '.' '.' '.']
 ['.' '.' '.' ... '.' '.' '.']
 ['.' '.' '.' ... '.' '.' '.']
 ...
 ['57.4' '52.9' '57' ... '71.1' '63.9' '51.9']
 ['48.6' '50.3' '61.2' ... '68.5' '60.9' '54.1']
 ['60.2' '59' '61.6' ... '75.4' '62.3' '52.4']]
```

In [9]:
```
df = pd.read_csv("CA042239T.csv")
(df)
```

Out[9]:

	State_id	YEAR	Month	TMAX (F)	TMEAN (F)	TMIN (F)
0	'042239'	1887	1	.	.	.
1	'042239'	1887	2	.	.	.
2	'042239'	1887	3	.	.	.
3	'042239'	1887	4	.	.	.
4	'042239'	1887	5	.	.	.
...	...	...	...	...	...	...
1531	'042239'	2014	8	83.2	68.3	53.4
1532	'042239'	2014	9	82.5	66.7	50.8
1533	'042239'	2014	10	75.4	58.6	41.8
1534	'042239'	2014	11	62.3	49.1	35.8
1535	'042239'	2014	12	52.4	42.2	32

1536 rows × 6 columns

In [10]:
```
tmin = df['TMIN (F) '].values
columns = 12
rows = 128


# Reshape the Tmax data into a matrix with each row as a year and each column as a mor
tmin_matrix = tmin.reshape(rows, columns)
print(tmin_matrix)
```

```
[['. ' '. ' '. ' ... '. ' '. ' '. ']
 ['. ' '. ' '. ' ... '. ' '. ' '. ']
 ['. ' '. ' '. ' ... '. ' '. ' '. ']
 ...
 ['30.7' '30.5' '32.1' ... '39.5' '35.1' '29.7']
 ['28.9' '29' '36.7' ... '34.6' '34.6' '31.2']
 ['34.5' '34.5' '36.8' ... '41.8' '35.8' '32']]
```

In [11]:
```
df = pd.read_csv("CA042239T.csv")
(df)
```

Out[11]:

	State_id	YEAR	Month	TMAX (F)	TMEAN (F)	TMIN (F)
0	'042239'	1887	1	.	.	.
1	'042239'	1887	2	.	.	.
2	'042239'	1887	3	.	.	.
3	'042239'	1887	4	.	.	.
4	'042239'	1887	5	.	.	.
...	...	...	...	...	...	...
1531	'042239'	2014	8	83.2	68.3	53.4
1532	'042239'	2014	9	82.5	66.7	50.8
1533	'042239'	2014	10	75.4	58.6	41.8
1534	'042239'	2014	11	62.3	49.1	35.8
1535	'042239'	2014	12	52.4	42.2	32

1536 rows × 6 columns

In [12]:

```
tmean = df['TMEAN (F)'].values
columns = 12
rows = 128

tmean_matrix = tmean.reshape(rows, columns)
print(tmean_matrix)
```

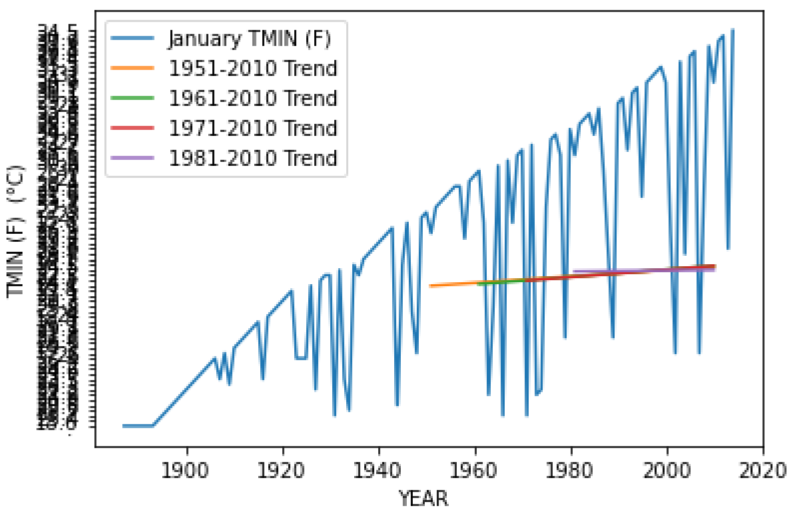
```
[['.' '.' '.' ... '.' '.' '.']
 ['.' '.' '.' ... '.' '.' '.']
 ['.' '.' '.' ... '.' '.' '.']
 ...
 ['44.1' '41.7' '44.6' ... '55.3' '49.5' '40.8']
 ['38.8' '39.7' '48.9' ... '51.5' '47.8' '42.7']
 ['47.4' '46.7' '49.2' ... '58.6' '49.1' '42.2']]
```

2.7

In [13]:

```
# Read the data file into a pandas DataFrame
#os.chdir(r"C:\Users\camiu\M336\MATH 336 FOLDER(shen)\MATH336 (SHEN)")
df = pd.read_csv("CA042239T.csv")

# Extract the Tmin column and the year column
tmin = df['TMIN (F) ']
years = df['YEAR  ']

# Create a new DataFrame with only the January Tmin values and the corresponding year
january_tmin = df[df['Month  '] == 1][['TMIN (F) ', 'YEAR  ']]

# Plot the January Tmin time series
plt.plot(january_tmin['YEAR  '], january_tmin['TMIN (F) '], '-', label='January TMIN (
plt.xlabel('YEAR  ')
plt.ylabel('TMIN (F)  (°C)')

# Define the time periods for the trend lines
periods = [
```

```
    [1951, 2010],
    [1961, 2010],
    [1971, 2010],
    [1981, 2010],
]

# Loop over the time periods
for period in periods:
    # Filter the data to only include the years in the current period
    filtered_data = january_tmin[(january_tmin['YEAR  '] >= period[0]) & (january_tmir

    # Create a Linear Regression model
    model = LinearRegression()

    # Fit the model to the data
    X = filtered_data['YEAR  '].values.reshape(-1, 1)
    y = filtered_data['TMIN (F) '].values
    model.fit(X, y)

    # Plot the trend line for the current period
    X_plot = np.array([[period[0]], [period[1]]])
    y_plot = model.predict(X_plot)
    plt.plot(X_plot, y_plot, label=f"{period[0]}-{period[1]} Trend")

# Add a legend to the plot
plt.legend()

plt.show()
```

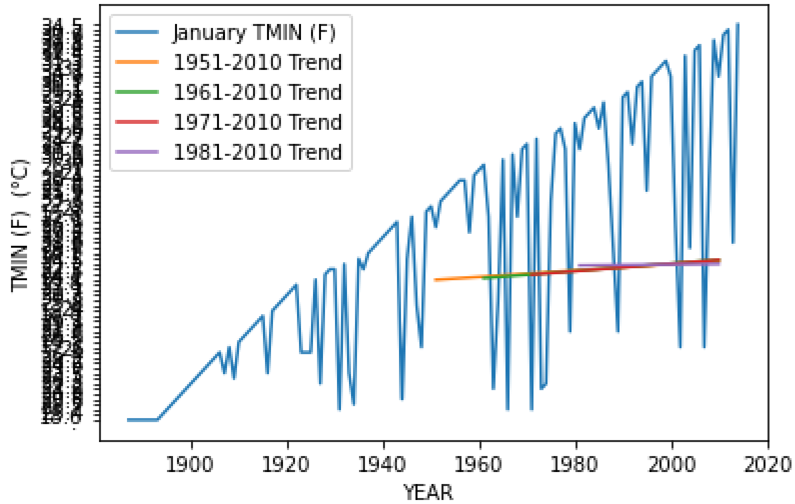


c. Based on the provided code, the temporal trend per decade for each of the four periods can be estimated by the slope of the linear regression trend line for each period. The slope represents the average change in temperature per year, which can be converted to the average change in temperature per decade by multiplying by 10. Below, I used the same code but modified it to show the temporal trends per decade.

In [14]:
```
# Load the data
df = pd.read_csv("CA042239T.csv")

# Extract the Tmin column and the year column
tmin = df['TMIN (F) ']
years = df['YEAR  ']
```

```
# Create a new DataFrame with only the January Tmin values and the corresponding year
january_tmin = df[df['Month  '] == 1][['TMIN (F) ', 'YEAR  ']]

# Plot the January Tmin time series
plt.plot(january_tmin['YEAR  '], january_tmin['TMIN (F) '], '-', label='January TMIN (
plt.xlabel('YEAR  ')
plt.ylabel('TMIN (F)  (°C)')

# Define the time periods for the trend lines
periods = [
    [1951, 2010],
    [1961, 2010],
    [1971, 2010],
    [1981, 2010],
]

# Loop over the time periods
for period in periods:
    # Filter the data to only include the years in the current period
    filtered_data = january_tmin[(january_tmin['YEAR  '] >= period[0]) & (january_tmin

    # Create a Linear Regression model
    model = LinearRegression()

    # Fit the model to the data
    X = filtered_data['YEAR  '].values.reshape(-1, 1)
    y = filtered_data['TMIN (F) '].values
    model.fit(X, y)

    # Plot the trend line for the current period
    X_plot = np.array([[period[0]], [period[1]]])
    y_plot = model.predict(X_plot)
    plt.plot(X_plot, y_plot, label=f"{period[0]}-{period[1]} Trend")

    # Calculate the trend per decade
    trend_per_year = model.coef_[0]
    trend_per_decade = trend_per_year * 10
    print(f"{period[0]}-{period[1]} Trend per Decade: {trend_per_decade:.2f} °C")

# Add a legend to the plot
plt.legend()

# Show the plot
plt.show()
```

```
1951-2010 Trend per Decade: 0.62 °C
1961-2010 Trend per Decade: 0.72 °C
1971-2010 Trend per Decade: 0.73 °C
1981-2010 Trend per Decade: 0.09 °C
```



2.9

In [15]:
```python
data = pd.read_csv("NOAAGlobalT.csv", header=0, index_col=0)
(data)
```

Out[15]:

	LAT	LON	1880-1	1880-2	1880-3	1880-4	1880-5	1880-6	1880-7	1880-8	...	2016-4	2016-5	201
1	-87.5	2.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
2	-87.5	7.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
3	-87.5	12.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
4	-87.5	17.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
5	-87.5	22.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2588	87.5	337.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
2589	87.5	342.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
2590	87.5	347.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
2591	87.5	352.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99
2592	87.5	357.5	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	...	-999.9	-999.9	-99

2592 rows × 1647 columns

In [16]:
```python
data.iloc[0,2:]
```

Out[16]:
```
1880-1    -999.9
1880-2    -999.9
1880-3    -999.9
1880-4    -999.9
1880-5    -999.9
           ...
2016-9    -999.9
2016-10   -999.9
2016-11   -999.9
2016-12   -999.9
2017-1    -999.9
Name: 1, Length: 1645, dtype: float64
```

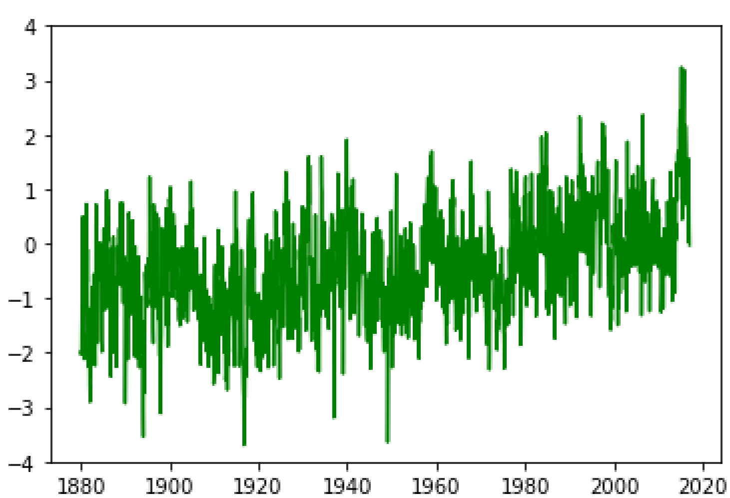
In [17]:
```
t = np.linspace(1880, 2017, 1645)
t
```

Out[17]:
```
array([1880.        , 1880.08333333, 1880.16666667, ..., 2016.83333333,
       2016.91666667, 2017.        ])
```

In [18]:
```
timesd = data.iloc[1776,2:]
timesd
```

Out[18]:
```
1880-1    -1.9840
1880-2    -2.0391
1880-3    -1.9442
1880-4    -1.2338
1880-5     0.1533
            ...
2016-9     0.0294
2016-10    0.6892
2016-11    1.5537
2016-12    0.3939
2017-1    -0.0339
Name: 1777, Length: 1645, dtype: float64
```

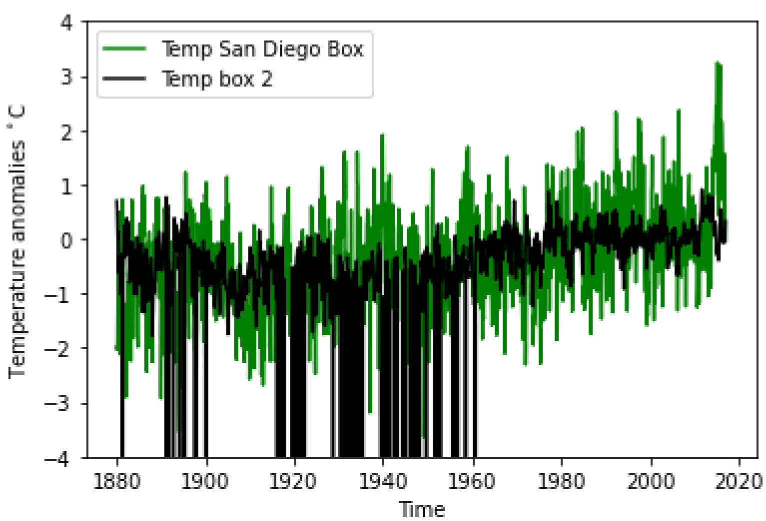
In [19]:
```
timesd.replace(-999.9, np.nan, inplace = True)
plt.plot(t, timesd, 'g')
plt.ylim([-4, 4])
plt.show()
```



In [20]:
```
t600 = data.iloc[599,2:]
t600
```

Out[20]:
```
1880-1     0.6943
1880-2     0.5580
1880-3     0.3095
1880-4     0.0025
1880-5     0.1199
             ...
2016-9     0.0271
2016-10    0.0883
2016-11   -0.0681
2016-12    0.0921
2017-1     0.3209
Name: 600, Length: 1645, dtype: float64
```

In [21]:
```python
timesd.replace(-999.9, np.nan, inplace = True)
plt.plot(t, timesd, 'g', label = 'Temp San Diego Box')
plt.plot(t, t600, '-k', label = 'Temp box 2')

plt.ylim([-4, 4])
plt.legend()
plt.xlabel('Time')
plt.ylabel('Temperature anomalies $^\circ$C')
plt.show()
```



In [ ]: