

```
In [1]: import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]: HEADERS = {'User-Agent': 'Mozilla/5.0 (iPad; CPU OS 12_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0 Mobile/21E5195.0 Safari/604.1'}
r = requests.get("https://weather.com/weather/tenday/1/ae8230efd4bc57fdf721a02c")
print(r.text)
```

```
<!doctype html><html dir="ltr" lang="en-US"><head>
  <meta data-react-helmet="true" charset="utf-8"/><meta data-react-helmet="true" name="viewport" content="width=device-width, initial-scale=1, viewport-fit=cover"/><meta data-react-helmet="true" name="robots" content="max-image-preview:large"/><meta data-react-helmet="true" name="robots" content="index, follow"/><meta data-react-helmet="true" name="referrer" content="origin"/><meta data-react-helmet="true" name="description" content="Be prepared with the most accurate 10-day forecast for London, England, United Kingdom with highs, lows, chance of precipitation from The Weather Channel and Weather.com"/><meta data-react-helmet="true" name="msapplication-TileColor" content="#ffffff"/><meta data-react-helmet="true" name="msapplication-TileImage" content="/daily/assets/ms-icon-144x144.d353af.png"/><meta data-react-helmet="true" name="theme-color" content="#ffffff"/><meta data-react-helmet="true" name="apple-itunes-app" content="app-id=295646461"/><meta data-react-helmet="true" property="og:title" content="London, England, United Kingdom 10-Day Weather Forecast - The Weather Channel | Weather.com"/><meta data-react-helmet="true" property="og:image" content="https://s.w-x.co/240x180_twc_default.png"/><meta data-react-helmet="true" property="og:image:url" content="https://s.w-x.co/240x180_twc_default.png"/><meta data-react-helmet="true" property="og:image:alt" content="Weather Channel logo" /></head>
```

```
In [3]: #create a beautifulsoup object
soup= BeautifulSoup(r.text, "html.parser")
print(soup.prettify())
```

```
<!DOCTYPE html>
<html dir="ltr" lang="en-US">
  <head>
    <meta charset="utf-8" data-react-helmet="true"/>
    <meta content="width=device-width, initial-scale=1, viewport-fit=cover" data-react-helmet="true" name="viewport"/>
    <meta content="max-image-preview:large" data-react-helmet="true" name="robots"/>
    <meta content="index, follow" data-react-helmet="true" name="robots"/>
    <meta content="origin" data-react-helmet="true" name="referrer"/>
    <meta content="Be prepared with the most accurate 10-day forecast for London, England, United Kingdom with highs, lows, chance of precipitation from The Weather Channel and Weather.com" data-react-helmet="true" name="description"/>
    <meta content="#ffffff" data-react-helmet="true" name="msapplication-TileColor"/>
    <meta content="/daily/assets/ms-icon-144x144.d353af.png" data-react-helmet="true" name="msapplication-TileImage"/>
    <meta content="#ffffff" data-react-helmet="true" name="theme-color"/>
  </head>
```

```
In [95]: #extract the forecast data  
Temperature=soup.select(".DailyForecast--DisclosureList--nosQS .DailyContent--  
Date=soup.select(".DailyForecast--DisclosureList--nosQS .DailyContent--daypartl  
Others= soup.select(".DailyForecast--DisclosureList--nosQS .DetailsTable--value
```

In [96]: Temperature

```

Out[96]: [<span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">82°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">61°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">84°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">57°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">75°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">57°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">74°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">62°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">75°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">59°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">73°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">56°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">73°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">57°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">71°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">58°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">72°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">56°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">74°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">57°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">73°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">57°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">75°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">57°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">76°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">58°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">78°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">59°</span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">79°</span>]

```

```
span>,
  <span class="DailyContent--temp--1s3a7" data-testid="TemperatureValue">59°</
span>]
```

In [97]: Date

```
Out[97]: [<span class="DailyContent--daypartDate--3VG1z">Sat 24</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sat 24</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sun 25</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sun 25</span>,
  <span class="DailyContent--daypartDate--3VG1z">Mon 26</span>,
  <span class="DailyContent--daypartDate--3VG1z">Mon 26</span>,
  <span class="DailyContent--daypartDate--3VG1z">Tue 27</span>,
  <span class="DailyContent--daypartDate--3VG1z">Tue 27</span>,
  <span class="DailyContent--daypartDate--3VG1z">Wed 28</span>,
  <span class="DailyContent--daypartDate--3VG1z">Wed 28</span>,
  <span class="DailyContent--daypartDate--3VG1z">Thu 29</span>,
  <span class="DailyContent--daypartDate--3VG1z">Thu 29</span>,
  <span class="DailyContent--daypartDate--3VG1z">Fri 30</span>,
  <span class="DailyContent--daypartDate--3VG1z">Fri 30</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sat 01</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sat 01</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sun 02</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sun 02</span>,
  <span class="DailyContent--daypartDate--3VG1z">Mon 03</span>,
  <span class="DailyContent--daypartDate--3VG1z">Mon 03</span>,
  <span class="DailyContent--daypartDate--3VG1z">Tue 04</span>,
  <span class="DailyContent--daypartDate--3VG1z">Tue 04</span>,
  <span class="DailyContent--daypartDate--3VG1z">Wed 05</span>,
  <span class="DailyContent--daypartDate--3VG1z">Wed 05</span>,
  <span class="DailyContent--daypartDate--3VG1z">Thu 06</span>,
  <span class="DailyContent--daypartDate--3VG1z">Thu 06</span>,
  <span class="DailyContent--daypartDate--3VG1z">Fri 07</span>,
  <span class="DailyContent--daypartDate--3VG1z">Fri 07</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sat 08</span>,
  <span class="DailyContent--daypartDate--3VG1z">Sat 08</span>]
```

In [98]: Others

```
Out[98]: [<span class="DetailsTable--value--2YD0-" data-testid="PercentageValue">56%
</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="UVIndexValue">7 of 1
0</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="SunriseTime">4:44 am
</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="SunsetTime">9:21 pm
</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="PercentageValue">61%
</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="UVIndexValue">0 of 1
0</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="MoonriseTime">10:54
am</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="MoonsetTime">12:39 a
m</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="PercentageValue">48%
</span>,
  <span class="DetailsTable--value--2YD0-" data-testid="UVIndexValue">7 of 1
0</span>]
```

```
In [103]: #seperate the temperatures out
Temperatures= [TemperatureValue.get_text() for TemperatureValue in Temperature
Temperatures
```

```
Out[103]: ['82°',
'61°',
'84°',
'57°',
'75°',
'57°',
'74°',
'62°',
'75°',
'59°',
'73°',
'56°',
'73°',
'57°',
'71°',
'58°',
'72°',
'56°',
'74°',
'57°',
'73°',
'57°',
'75°',
'57°',
'76°',
'58°',
'78°',
'59°',
'79°',
'59°']
```

```
In [68]: #seperate the dates out  
Dates= [daypartDate.get_text() for daypartDate in Date]  
Dates
```

```
Out[68]: ['Sat 24',  
          'Sat 24',  
          'Sun 25',  
          'Sun 25',  
          'Mon 26',  
          'Mon 26',  
          'Tue 27',  
          'Tue 27',  
          'Wed 28',  
          'Wed 28',  
          'Thu 29',  
          'Thu 29',  
          'Fri 30',  
          'Fri 30',  
          'Sat 01',  
          'Sat 01',  
          'Sun 02',  
          'Sun 02',  
          'Mon 03',  
          'Mon 03',  
          'Tue 04',  
          'Tue 04',  
          'Wed 05',  
          'Wed 05',  
          'Thu 06',  
          'Thu 06',  
          'Fri 07',  
          'Fri 07',  
          'Sat 08',  
          'Sat 08']
```



```
In [102]: #get the actual forecast values only
Other_Values= [PercentageValue.get_text() for PercentageValue in Others]
Other_Values
```

```
Out[102]: ['56%',
'7 of 10',
'4:44 am',
'9:21 pm',
'61%',
'0 of 10',
'10:54 am',
'12:39 am',
'48%',
'7 of 10',
'4:44 am',
'9:21 pm',
'65%',
'0 of 10',
'12:05 pm',
'12:51 am',
'46%',
'6 of 10',
'4:44 am',
'10:54 am']
```

```
In [199]: import numpy as np
```

```
In [64]: #convert dimension of Other_Values
num = np.array(Other_Values)
reshaped= num.reshape(15,8)
print(reshaped)
```

```
[['56%' '7 of 10' '4:44 am' '9:21 pm' '61%' '0 of 10' '10:54 am'
'12:39 am']
['48%' '7 of 10' '4:44 am' '9:21 pm' '65%' '0 of 10' '12:05 pm'
'12:51 am']
['46%' '6 of 10' '4:44 am' '9:21 pm' '58%' '0 of 10' '1:15 pm' '1:01 am']
['62%' '4 of 10' '4:45 am' '9:21 pm' '76%' '0 of 10' '2:28 pm' '1:12 am']
['66%' '5 of 10' '4:45 am' '9:21 pm' '74%' '0 of 10' '3:44 pm' '1:23 am']
['54%' '6 of 10' '4:46 am' '9:21 pm' '67%' '0 of 10' '5:05 pm' '1:37 am']
['56%' '5 of 10' '4:47 am' '9:21 pm' '73%' '0 of 10' '6:31 pm' '1:55 am']
['67%' '5 of 10' '4:47 am' '9:20 pm' '78%' '0 of 10' '7:56 pm' '2:21 am']
['61%' '6 of 10' '4:48 am' '9:20 pm' '70%' '0 of 10' '9:13 pm' '2:59 am']
['56%' '7 of 10' '4:49 am' '9:20 pm' '71%' '0 of 10' '10:13 pm'
'3:55 am']
['61%' '7 of 10' '4:49 am' '9:19 pm' '72%' '0 of 10' '10:55 pm'
'5:12 am']
['58%' '7 of 10' '4:50 am' '9:19 pm' '72%' '0 of 10' '11:23 pm'
'6:43 am']
['59%' '6 of 10' '4:51 am' '9:18 pm' '70%' '0 of 10' '11:44 pm'
'8:17 am']
['57%' '7 of 10' '4:52 am' '9:18 pm' '71%' '0 of 10' '--' '9:48 am']
['61%' '6 of 10' '4:53 am' '9:17 pm' '73%' '0 of 10' '12:00 am'
'11:15 am']]
```

In [104]: *#convert dimension of the Temperatures*

```
num = np.array(Temperatures)
reshaped_ = num.reshape(15,2)
print(reshaped_)
```

```
[['82°' '61°']
 ['84°' '57°']
 ['75°' '57°']
 ['74°' '62°']
 ['75°' '59°']
 ['73°' '56°']
 ['73°' '57°']
 ['71°' '58°']
 ['72°' '56°']
 ['74°' '57°']
 ['73°' '57°']
 ['75°' '57°']
 ['76°' '58°']
 ['78°' '59°']
 ['79°' '59°']]
```

In [168]: *#convert dimension of the dates*

```
num = np.array(Dates)
reshaped_1= num.reshape(15,2)
print(reshaped_1)
```

```
[['Sat 24' 'Sat 24']
 ['Sun 25' 'Sun 25']
 ['Mon 26' 'Mon 26']
 ['Tue 27' 'Tue 27']
 ['Wed 28' 'Wed 28']
 ['Thu 29' 'Thu 29']
 ['Fri 30' 'Fri 30']
 ['Sat 01' 'Sat 01']
 ['Sun 02' 'Sun 02']
 ['Mon 03' 'Mon 03']
 ['Tue 04' 'Tue 04']
 ['Wed 05' 'Wed 05']
 ['Thu 06' 'Thu 06']
 ['Fri 07' 'Fri 07']
 ['Sat 08' 'Sat 08']]
```

```
In [164]: #construct dataframes for Dates  
df3=pd.DataFrame(reshaped_1, columns=['Date_M', 'Date_N'])  
print(df3)
```

	Date_M	Date_N
0	Sat 24	Sat 24
1	Sun 25	Sun 25
2	Mon 26	Mon 26
3	Tue 27	Tue 27
4	Wed 28	Wed 28
5	Thu 29	Thu 29
6	Fri 30	Fri 30
7	Sat 01	Sat 01
8	Sun 02	Sun 02
9	Mon 03	Mon 03
10	Tue 04	Tue 04
11	Wed 05	Wed 05
12	Thu 06	Thu 06
13	Fri 07	Fri 07
14	Sat 08	Sat 08

```
In [169]: #construct dataframe for Temperatures  
df4=pd.DataFrame(reshaped_, columns=['Day', 'Night'])  
print(df4)
```

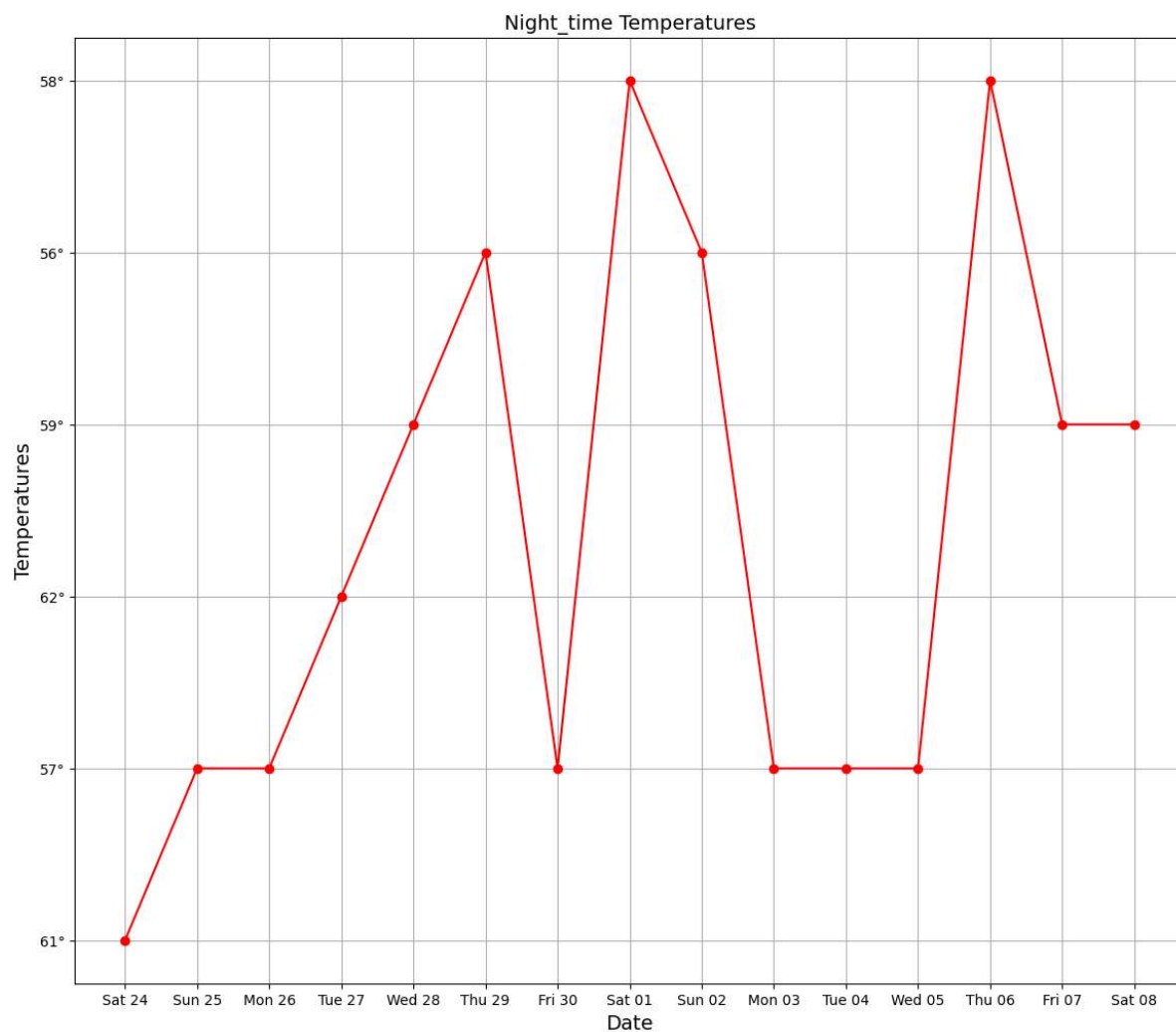
	Day	Night
0	82°	61°
1	84°	57°
2	75°	57°
3	74°	62°
4	75°	59°
5	73°	56°
6	73°	57°
7	71°	58°
8	72°	56°
9	74°	57°
10	73°	57°
11	75°	57°
12	76°	58°
13	78°	59°
14	79°	59°

```
In [134]: #construct dataframe for Other_Values
df1=pd.DataFrame(reshaped, columns =['Humidity_M', 'UV Index_M', 'Sunrise', 'Sunset', 'Humidity_N', 'UV Index_N', 'Moonrise', 'Moonset'])
print(df1)
```

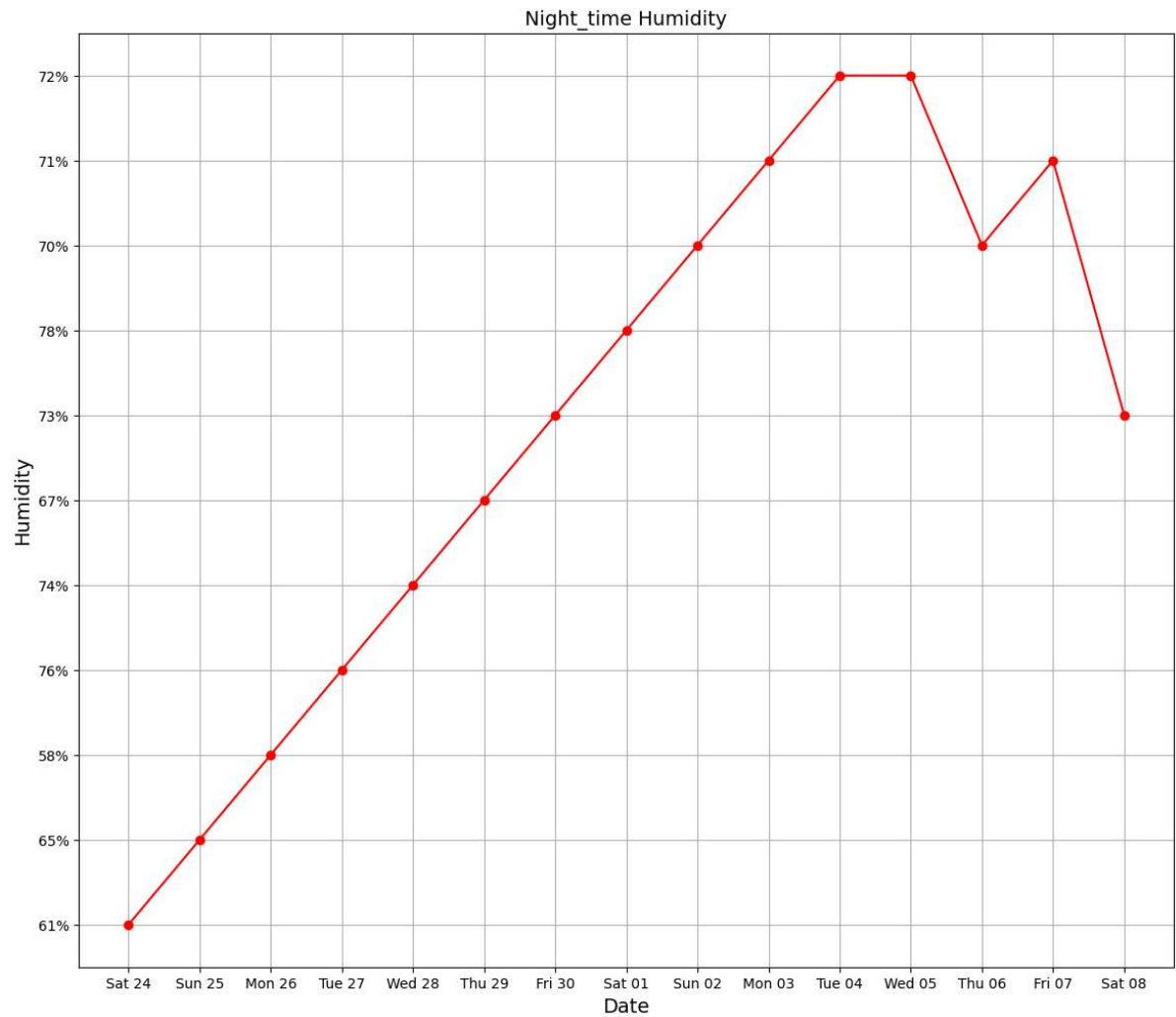
	Humidity_M	UV Index_M	Sunrise	Sunset	Humidity_N	UV Index_N	Moonrise	\
0	56%	7 of 10	4:44 am	9:21 pm	61%	0 of 10	10:54 am	
1	48%	7 of 10	4:44 am	9:21 pm	65%	0 of 10	12:05 pm	
2	46%	6 of 10	4:44 am	9:21 pm	58%	0 of 10	1:15 pm	
3	62%	4 of 10	4:45 am	9:21 pm	76%	0 of 10	2:28 pm	
4	66%	5 of 10	4:45 am	9:21 pm	74%	0 of 10	3:44 pm	
5	54%	6 of 10	4:46 am	9:21 pm	67%	0 of 10	5:05 pm	
6	56%	5 of 10	4:47 am	9:21 pm	73%	0 of 10	6:31 pm	
7	67%	5 of 10	4:47 am	9:20 pm	78%	0 of 10	7:56 pm	
8	61%	6 of 10	4:48 am	9:20 pm	70%	0 of 10	9:13 pm	
9	56%	7 of 10	4:49 am	9:20 pm	71%	0 of 10	10:13 pm	
10	61%	7 of 10	4:49 am	9:19 pm	72%	0 of 10	10:55 pm	
11	58%	7 of 10	4:50 am	9:19 pm	72%	0 of 10	11:23 pm	
12	59%	6 of 10	4:51 am	9:18 pm	70%	0 of 10	11:44 pm	
13	57%	7 of 10	4:52 am	9:18 pm	71%	0 of 10	--	
14	61%	6 of 10	4:53 am	9:17 pm	73%	0 of 10	12:00 am	

	Moonset
0	12:39 am
1	12:51 am
2	1:01 am
3	1:12 am
4	1:23 am
5	1:37 am
6	1:55 am
7	2:21 am
8	2:59 am
9	3:55 am
10	5:12 am
11	6:43 am
12	8:17 am
13	9:48 am
14	11:15 am

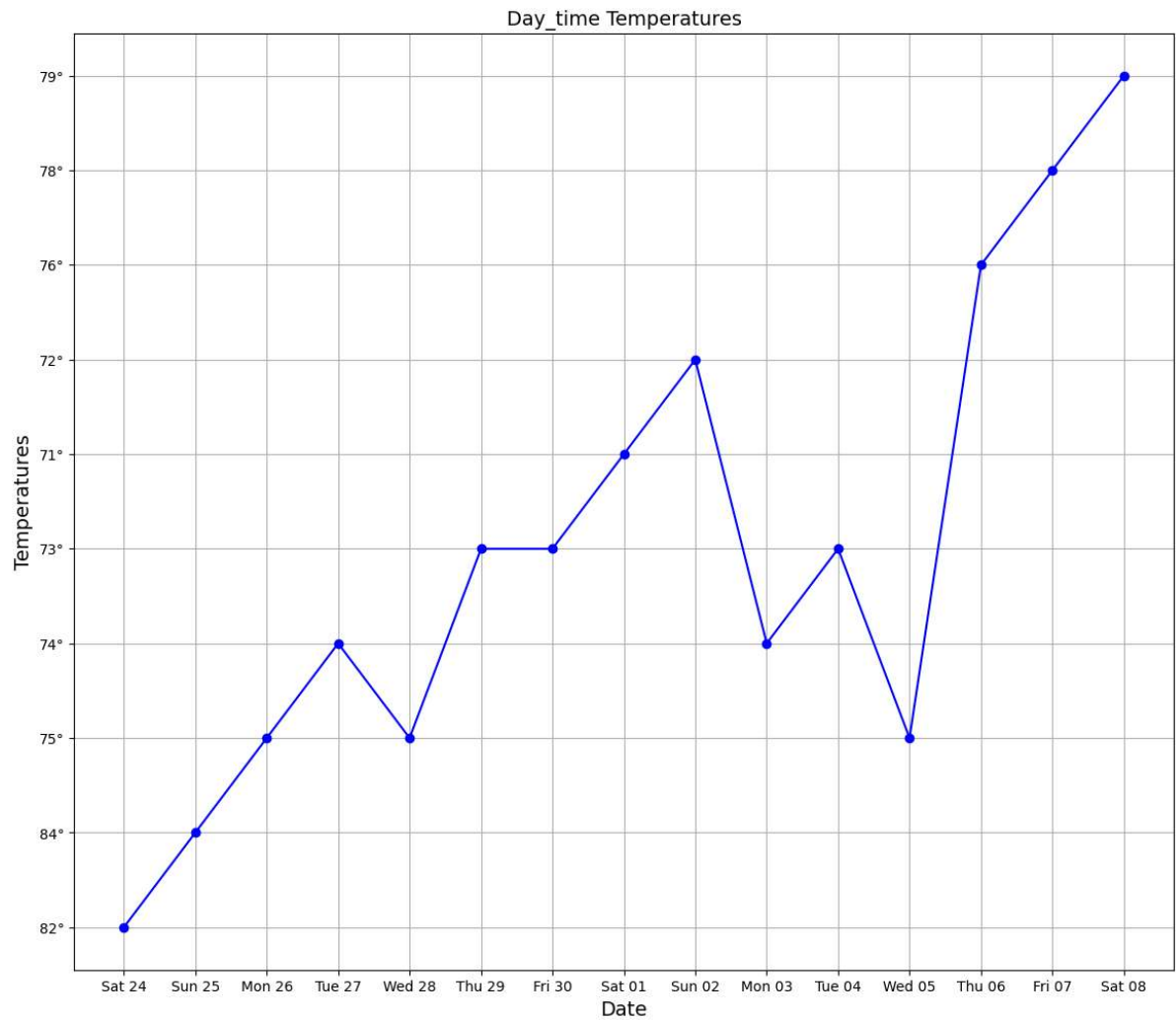
```
In [192]: #create visualizations
plt.figure(figsize=(14,12))
plt.plot(df3['Date_N'], df4['Night'], color='red', marker='o')
plt.title('Night_time Temperatures', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Temperatures', fontsize=14)
plt.grid(True)
plt.show()
```



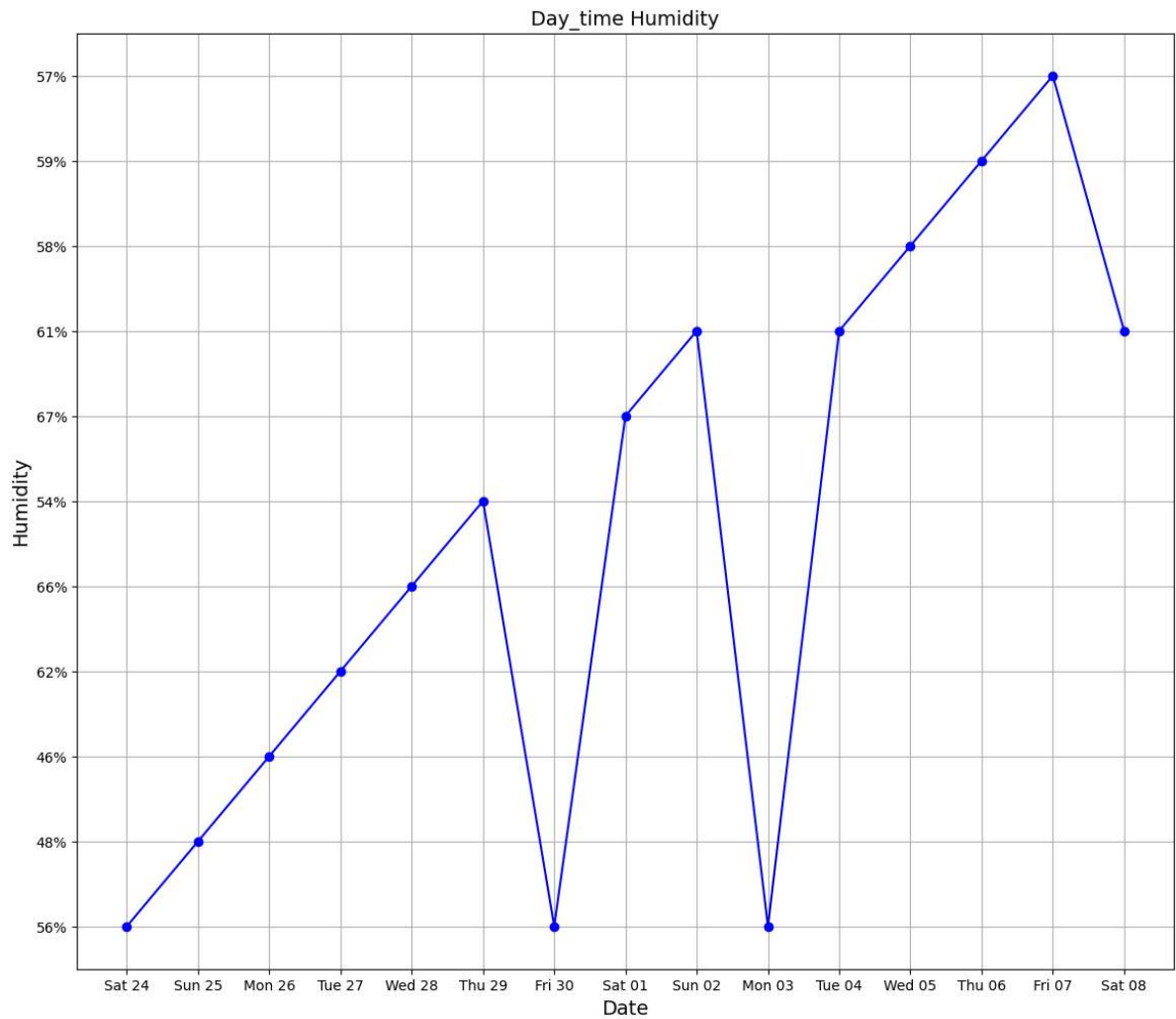
```
In [189]: #create visualizations
plt.figure(figsize=(14,12))
plt.plot(df3['Date_N'], df1['Humidity_N'], color='red', marker='o')
plt.title('Night_time Humidity', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Humidity', fontsize=14)
plt.grid(True)
plt.show()
```



```
In [190]: #create visualizations
plt.figure(figsize=(14,12))
plt.plot(df3['Date_M'], df4['Day'], color='blue', marker='o')
plt.title('Day_time Temperatures', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Temperatures', fontsize=14)
plt.grid(True)
plt.show()
```

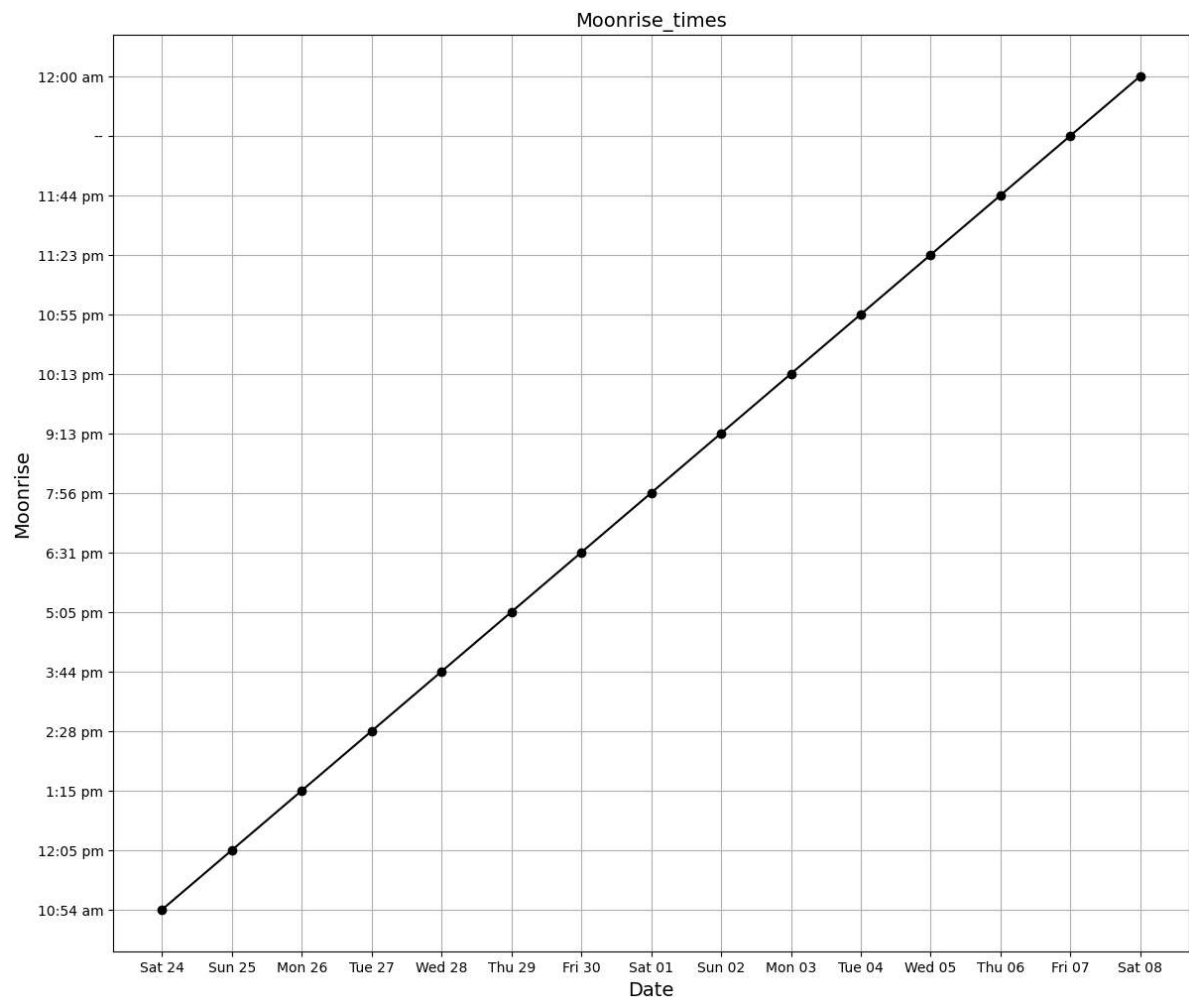


```
In [191]: #create visualizations
plt.figure(figsize=(14,12))
plt.plot(df3['Date_M'], df1['Humidity_M'], color='blue', marker='o')
plt.title('Day_time Humidity', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Humidity', fontsize=14)
plt.grid(True)
plt.show()
```

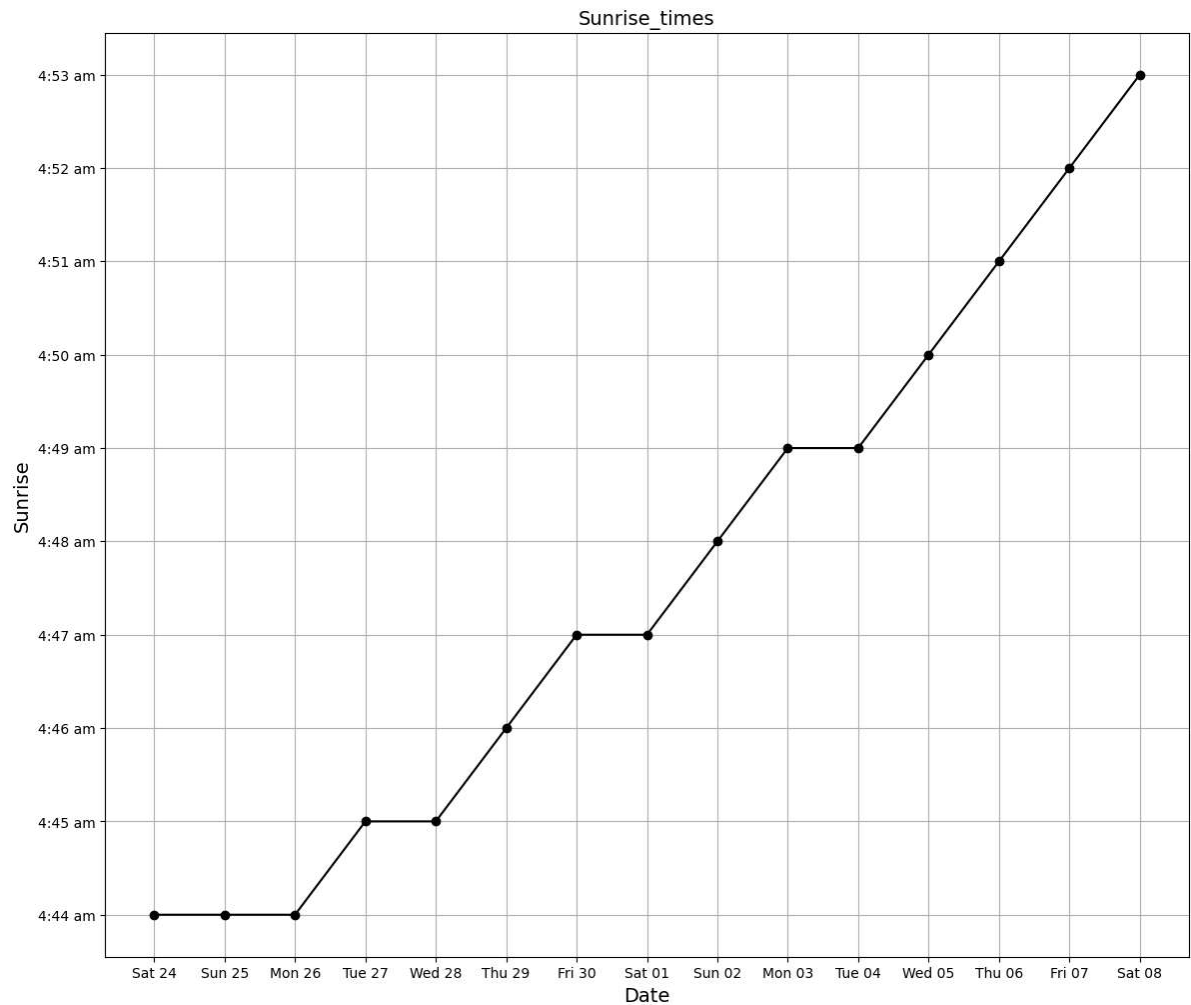




```
In [193]: #create visualizations
plt.figure(figsize=(14,12))
plt.plot(df3['Date_N'], df1['Moonrise'], color='black', marker='o')
plt.title('Moonrise_times', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Moonrise', fontsize=14)
plt.grid(True)
plt.show()
```



```
In [194]: #create visualizations
plt.figure(figsize=(14,12))
plt.plot(df3['Date_M'], df1['Sunrise'], color='black', marker='o')
plt.title('Sunrise_times', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Sunrise', fontsize=14)
plt.grid(True)
plt.show()
```



```
In [218]: x=df3['Date_M']  
y=df4['Day']  
plt.scatter(x,y)  
plt.xlabel=("Date")  
plt.ylabel=("Temperature (Fahrenheit)")  
plt.title=("Daytime Temperature Distribution")  
plt.grid(True)
```

