Submitted by: Angelo Luis C. Cu

```
import requests # function used before on 7.2
def make_request(endpoint, payload=None):
  """
  Make a request to a specific endpoint on the weather API
  passing headers and optional payload.
  Parameters:
  - endpoint: The endpoint of the API you want to
  make a GET request to.
  - payload: A dictionary of data to pass along
  with the request.
  Returns:
  Response object.
  """
  return requests.get(
    f'https://www.ncdc.noaa.gov/cdo-web/api/v2/{endpoint}',
    headers={
    'token': 'KFwvUbvvdsWskOidlBysYFNxyWtdtVcV'
    },
    params=payload
  )
```

```
import datetime
from IPython import display # for updating the cell dynamically
current = datetime.date(2018, 1, 1) # from jan 1, 2018
end = datetime.date(2019, 1, 1) # to jan 1, 2019
results = []
while current < end: # while the date is not yet jan 1, 2019,
  display.clear_output(wait=True)
  display.display(f'Gathering data for {str(current)}') # gather data

  response = make_request(
    'data',
    {
      'datasetid' : 'GHCND', # daily summaries
      'locationid' : 'CITY:US360019', # NYC
      'startdate' : current,
      'enddate' : current,
      'units' : 'metric',
      'limit' : 1000 # max allowed
    }
  )
  if response.ok:
    results.extend(response.json()['results']) # adds the result to results list
  current += datetime.timedelta(days=1) # adds 1 day for the earlier while loop to work
```

```
'Gathering data for 2018-12-31'
```

```
import pandas as pd
df = pd.DataFrame(results) # transform the list to a dataframe

df.head()
```

	date	datatype	station	attributes	value
0	2018-01-01T00:00:00	PRCP	GHCND:US1CTFR0039	,,N,0800	0.0
1	2018-01-01T00:00:00	PRCP	GHCND:US1NJBG0015	,,N,1050	0.0
2	2018-01-01T00:00:00	SNOW	GHCND:US1NJBG0015	,,N,1050	0.0
3	2018-01-01T00:00:00	PRCP	GHCND:US1NJBG0017	,,N,0920	0.0
4	2018-01-01T00:00:00	SNOW	GHCND:US1NJBG0017	,,N,0920	0.0

Next steps: View recommended plots

```
df.to_csv('data/nyc_weather_2018.csv', index=False) # added a folder named data to colab
```

```
import sqlite3 # manipulating data with sqlite
```

```
with sqlite3.connect('data/weather.db') as connection:
  df.to_sql(
    'weather', connection, index=False, if_exists='replace'
  )


response = make_request( # another way to get data
  'stations',
  {
    'datasetid' : 'GHCND', # daily summaries
    'locationid' : 'CITY:US360019', # NYC
    'limit' : 1000 # max allowed
  }
)
# convert directly to a dataframe
stations = pd.DataFrame(response.json()['results'])[['id', 'name', 'latitude', 'longitude', 'elevation']]
# save to csv
stations.to_csv('data/weather_stations.csv', index=False)
```