USING THE NCEI API

```
import requests
def make_request(endpoint, payload=None):
  return requests.get(
  f'https://www.ncdc.noaa.gov/cdo-web/api/v2/{endpoint}',
  headers={
  'token': 'ADPwCcoPIaSsUZaapBWqIqMcJIbfpgVB'
  },
  params=payload
  )
```

COLLECTING ALL DATA POINTS fOR 2018 IN NYC

```
import datetime
from IPython import display # for updating the cell dynamically
current = datetime.date(2018, 1, 1)
end = datetime.date(2019, 1, 1)
results = []
while current < end:
  # update the cell with status information
  display.clear_output(wait=True)
  display.display(f'Gathering data for {str(current)}')
  response = make_request(
    'data',
    {
      'datasetid' : 'GHCND', # Global Historical Climatology Network - Daily (GHCND) dataset
      'locationid' : 'CITY:US360019', # NYC
      'startdate' : current,
      'enddate' : current,
      'units' : 'metric',
      'limit' : 1000 # max allowed
    }
  )
  if response.ok:
    # we extend the list instead of appending to avoid getting a nested list
    results.extend(response.json()['results'])
  # update the current date to avoid an infinite loop
  current += datetime.timedelta(days=1)
```

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

```
import pandas as pd
df = pd.DataFrame(results)
df.head()
```

	date	datatype	station	attributes	value

	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('/content/nyc_weather_2018.csv', index=False)
```

```
import sqlite3

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

```
response = make_request('stations', {'datasetid':'GHCND', 'locationid':'CITY:US360019', 'limit
stations = pd.DataFrame(response.json()['results'])[['id', 'name', 'latitude', 'longitude', 'e
stations.to_csv('/content/weather_stations.csv', index=False)
with sqlite3.connect('/content/weather.db') as connection:
  stations.to_sql('stations', connection, index=False, if_exists='replace')
```

```
1
```