Found a python API on GitHub that allowed us to access music charts from billboard.com. Since we hit a road block with not being able to access any geographical or user data with Spotify API but saw audio feature data was available, decided to look at music preferences over the last 20 years and compare audio feature trends.

Requests to this API seemed very straightforward at first, didn't need any authorization or access token, didn't have to sign up for an account or get an API key.

In [1]:
```
pip install billboard.py
```

In [3]:
```
import pandas as pd
import json
import billboard
from time import sleep

#billboard.charts()
```

In [21]:
```
#Collect chart data from current week back to Jan 1, 2000

#Create empty list to store data collected
top100_data = []

chart_date = "2019-12-14"
top100_chart = billboard.ChartData('hot-100', chart_date)
counter = 100

for i in range (1, counter+1):
    top_song = top100_chart[0]  # Get no. 1 song on chart
    title = top_song.title
    artist = top_song.artist
    weeks = top_song.weeks

    top100_data_dict = {"Chart Date": chart_date, "Title": title, "Artist": artist,
                        "Number of Weeks In Top 100": weeks}

    top100_data.append(top100_data_dict)
    #Allow for break in requests to server between weeks
    sleep(7)
    #Reset chart date to previous week and get corresponding chart
    chart_date = top100_chart.previousDate
    top100_chart = billboard.ChartData('hot-100', chart_date)

print(top100_data)
```

```
9', 'Title': 'Nice For What', 'Artist': 'Drake', 'Number of Weeks In Top 100': 8}, {'Chart Date': '2018-06-02', 'Titl
e': 'Nice For What', 'Artist': 'Drake', 'Number of Weeks In Top 100': 7}, {'Chart Date': '2018-05-26', 'Title': 'This
Is America', 'Artist': 'Childish Gambino', 'Number of Weeks In Top 100': 2}, {'Chart Date': '2018-05-19', 'Title': 'T
his Is America', 'Artist': 'Childish Gambino', 'Number of Weeks In Top 100': 1}, {'Chart Date': '2018-05-12', 'Titl
e': 'Nice For What', 'Artist': 'Drake', 'Number of Weeks In Top 100': 4}, {'Chart Date': '2018-05-05', 'Title': 'Nice
```

However, once we started running the code to extract every number one hit song from each week over the last 20 years we hit a couple of problems.

The first problem: HTTP 429 Too Many Requests Response Status Code.

```
HTTPError: 429 Client Error: Too Many Requests for url: http://www.billboard.com/charts/hot-100/2011-05-07
```

It did not like us making so many requests at once. We incorporated a "sleep(5)" so that w each iteration it takes a 5 second break before a new request. We had 1040 requests to make, so we also broke this up into smaller blocks, iterating over 52 instead oof 100, then 26 instead of 52. We also gave it a bit of a break before running the next block of iterations. This worked decently consistently, sometimes we'd have to increase the sleep function to 7 or 8 seconds. Overall, it was working most of the time but it was VERY time consuming.

The next problem: A new error that arose after adding the "sleep( )" and working with smaller iterations, was a "Connection Error: Read Timed Out"

```
ConnectionError: HTTPSConnectionPool(host='www.billboard.com', port=443): Read timed
out.
```

which I interpreted (after much research) as the server not responding fast enough and when it wouldn't respond within a certain timeframe, it would produce an error. So I then implanted a timeout argument of "timeout=1000" to give the server a window of 1000 seconds to respond. This helped a lot. I was able to increase the block of iterations back up to 52 at a time to finish out the data. With each iteration, we saved

the data extracted to a dictionary and that dictionary to a list. Then we created a DataFame out of our list of dictionaries and saved the DataFrame to a csv file. This data included the chart date, song title, artist, and number of weeks on the top 100 chart.

In [86]:
```
chart_date = "2009-11-07"
top100_chart = billboard.ChartData('hot-100', date=chart_date, timeout=1000)
counter = 52

for i in range (1, counter+1):
    top_song = top100_chart[0]  # Get no. 1 song on chart
    title = top_song.title
    artist = top_song.artist
    weeks = top_song.weeks

    top100_data_dict = {"Chart Date": chart_date, "Title": title, "Artist": artist,
                        "Number of Weeks In Top 100": weeks}

    top100_data.append(top100_data_dict)

    sleep(8)
    chart_date = top100_chart.previousDate
    top100_chart = billboard.ChartData('hot-100', date=chart_date, timeout=1000)

print(top100_data)
```

```
00': 6}, {'Chart Date': '2009-03-14', 'Title': 'Right Round', 'Artist': 'Flo Rida', 'Number of Weeks In Top 100': 5},
{'Chart Date': '2009-03-07', 'Title': 'Right Round', 'Artist': 'Flo Rida', 'Number of Weeks In Top 100': 4}, {'Chart
Date': '2009-02-28', 'Title': 'Right Round', 'Artist': 'Flo Rida', 'Number of Weeks In Top 100': 3}, {'Chart Date':
'2009-02-21', 'Title': 'Crack A Bottle', 'Artist': 'Eminem, Dr. Dre & 50 Cent', 'Number of Weeks In Top 100': 4}, {'C
hart Date': '2009-02-14', 'Title': 'My Life Would Suck Without You', 'Artist': 'Kelly Clarkson', 'Number of Weeks In
Top 100': 3}, {'Chart Date': '2009-02-07', 'Title': 'My Life Would Suck Without You', 'Artist': 'Kelly Clarkson', 'Nu
mber of Weeks In Top 100': 2}, {'Chart Date': '2009-01-31', 'Title': 'Just Dance', 'Artist': "Lady Gaga Featuring Col
by O'Donis", 'Number of Weeks In Top 100': 24}, {'Chart Date': '2009-01-24', 'Title': 'Just Dance', 'Artist': "Lady G
aga Featuring Colby O'Donis", 'Number of Weeks In Top 100': 23}, {'Chart Date': '2009-01-17', 'Title': 'Just Dance',
'Artist': "Lady Gaga Featuring Colby O'Donis", 'Number of Weeks In Top 100': 22}, {'Chart Date': '2009-01-10', 'Titl
e': 'Single Ladies (Put A Ring On It)', 'Artist': 'Beyonce', 'Number of Weeks In Top 100': 11}, {'Chart Date': '2009-
01-03', 'Title': 'Single Ladies (Put A Ring On It)', 'Artist': 'Beyonce', 'Number of Weeks In Top 100': 10}, {'Chart
Date': '2008-12-27', 'Title': 'Single Ladies (Put A Ring On It)', 'Artist': 'Beyonce', 'Number of Weeks In Top 100':
9}, {'Chart Date': '2008-12-20', 'Title': 'Live Your Life', 'Artist': 'T.I. Featuring Rihanna', 'Number of Weeks In T
op 100': 11}, {'Chart Date': '2008-12-13', 'Title': 'Single Ladies (Put A Ring On It)', 'Artist': 'Beyonce', 'Number
of Weeks In Top 100': 7}, {'Chart Date': '2008-12-06', 'Title': 'Live Your Life', 'Artist': 'T.I. Featuring Rihanna',
'Number of Weeks In Top 100': 9}, {'Chart Date': '2008-11-29', 'Title': 'Live Your Life', 'Artist': 'T.I. Featuring R
ihanna', 'Number of Weeks In Top 100': 8}, {'Chart Date': '2008-11-22', 'Title': 'Live Your Life', 'Artist': 'T.I. Fe
aturing Rihanna', 'Number of Weeks In Top 100': 7}, {'Chart Date': '2008-11-15', 'Title': 'Live Your Life', 'Artist':
'T.I. Featuring Rihanna', 'Number of Weeks In Top 100': 6}]
```