Playlist Exercise - Requirements

The system should be able to:

1. Load the complete set of tracks in from ‘music.csv’
2. Print the complete list of tracks to the screen
3. Create a playlist from a list of track ids
    1. Sort a playlist
    2. Shuffle a playlist
    3. Calculate the total play time and store as an attribute
    4. Show the playlist
        1. Print the total play time
        2. List the tracks (ID, artist, title)
    5. Add tracks to the playlist (update play time)
    6. Remove tracks from the playlist (update play time)

music.csv

```
1,Taylor Swift,Everything Has Changed,4:12
2,Mumford and Sons,Little Lion Man,4:08
3,Hozier,Sedated,3:30
4,Mumford and Sons,Babel,4:05
5,Taylor Swift,I Knew You Were Trouble,3:40
6,Taylor Swift,We Are Never Ever Getting Back Together,3:16
7,Hozier,Jackie and Wilson,3:44
8,Hozier,Take Me To Church,4:02
9,Lily Allen,Not Fair,3:50
10,Hozier,Angel of Small Death & The Codeine Scene,3:34
11,Post Malone,Better Now,3:26
12,Miley Cyrus,Jolene,2:29
13,Florence + The Machine,Dog Days are Over,3:44
14,Rufus Wainwright,April Fools,4:03
15,Paul Simon,You Can Call me Al,4:35
16,Disturbed,The Sound of Silence,4:24
```

Assume there will be a Playlist Class

```
pl = Playlist(['1','2','3'],tlist)
pl.show_playlist()

Playing time 11:50
  1  Taylor Swift: Everything Has Changed
  2  Mumford and Sons: Little Lion Man
  3  Hozier: Sedated
In [24]:
pl.track_ids
Out[24]:
['1', '2', '3']
```

- `tlist` would be the full list of tracks

Playlist - add_track method

```
pl.add_track('7')
pl.add_track('12')
pl.add_track('8')
pl.show_playlist()

Playing time 22:05
  1  Taylor Swift: Everything Has Changed
  2  Mumford and Sons: Little Lion Man
  3  Hozier: Sedated
  7  Hozier: Jackie and Wilson
  12 Miley Cyrus: Jolene
  8  Hozier: Take Me To Church
```

More Methods

- remove_track
- shuffle_playlist
- sort_playlist

```
pl.add_track(13)
pl.remove_track(1)
pl.shuffle_playlist()
pl.show_playlist()
```

```
Playing time 21:37
  13  Florence + The Machine: Dog Days are Over
  8   Hozier: Take Me To Church
  12  Miley Cyrus: Jolene
  7   Hozier: Jackie and Wilson
  2   Mumford and Sons: Little Lion Man
  3   Hozier: Sedated
In [34]:
pl.sort_playlist()
pl.show_playlist()

Playing time 21:37
  13  Florence + The Machine: Dog Days are Over
  7   Hozier: Jackie and Wilson
  3   Hozier: Sedated
  8   Hozier: Take Me To Church
  12  Miley Cyrus: Jolene
  2   Mumford and Sons: Little Lion Man
```

What classes?

- A time class?
- A Full track list class?
  - also loads music.csv file
- A track class

Playlist
tot_time: TimeMS track_ids: tracklist: FullTrackList plist:
__init__(track_ids,tracklist) setup_list(tids) play_playlist() add_track(tid) remove_track(tid) sort_playlist() shuffle_playlist() calculate_time()

What classes?

Track
number: int title: str artist: str time: TimeMS
__init__(name,artist, tsting) __eq__(other) __lt__(other) __str__()

Playlist
tot_time: TimeMS track_ids: tracklist: FullTrackList plist:
__init__(track_ids,tracklist) setup_list(tids) play_playlist() add_track(tid) remove_track(tid) sort_playlist() shuffle_playlist() calculate_time()

FullTrackList
file_name: str tracks: Tracks list tracks_dict
__init__(file_name) load_csv('file_name') list_tracks()

TimeMS
mins: int secs: int
__init__(tstring) get_time() __str__() inc(tuple) reset()

Shuffle

- Shuffle a sequence (e.g. a list)
- Create new sequence drawing at random from original
- Or shuffle original list ***in-place***

```
from random import shuffle
In [2]:
a = [1,2,3,4,4,4,4,5,6]
In [15]:
shuffle(a)
In [16]:
a
Out[16]:
[4, 3, 4, 4, 4, 2, 6, 5, 1]
In [5]:
In [6]:
my_shuffle(a)
a
Out[6]:
[6, 4, 4, 1, 4, 5, 3, 4, 2]
```

Fisher-Yates Shuffle

- Shuffle a sequence (e.g. a list)
- Create new sequence drawing at random from original

Range	Roll	Scratch	Result
		1 2 3 4 5 6 7 8	
1-8	3	1 2 ~~3~~ 4 5 6 7 8	3
1–7	4	1 2 ~~3~~ 4 ~~5~~ 6 7 8	3 5
1–6	5	1 2 ~~3~~ 4 ~~5~~ 6 ~~7~~ 8	3 5 7

```
a = [4, 1, 4, 6, 4, 5, 3, 2, 4]

basic_fy_shuffle(a)
9 : [4, 1, 4, 6, 4, 5, 3, 2, 4]
8 : [4, 1, 4, 6, 4, 5, 3, 4]
7 : [1, 4, 6, 4, 5, 3, 4]
6 : [1, 4, 4, 5, 3, 4]
5 : [1, 4, 4, 5, 4]
4 : [1, 4, 5, 4]
3 : [1, 5, 4]
2 : [1, 5]
1 : [1]
Out[99]:
[2, 4, 6, 3, 4, 4, 4, 5, 1]
```

```
def basic_fy_shuffle(slist):
    # produce a copy of the original
    # list that we can pull apart.
    tlist = slist.copy()
    rlist = [] # target list
    for i in range(len(tlist),0,-1):
        print(i,":",tlist)
        sel = tlist[randint(0,i-1)]
        rlist.append(sel)
        tlist.remove(sel)
    return rlist
In [99]:
basic_fy_shuffle(a)
```

https://en.wikipedia.org/wiki/Fisher–Yates_shuffle

Fisher-Yates Shuffle

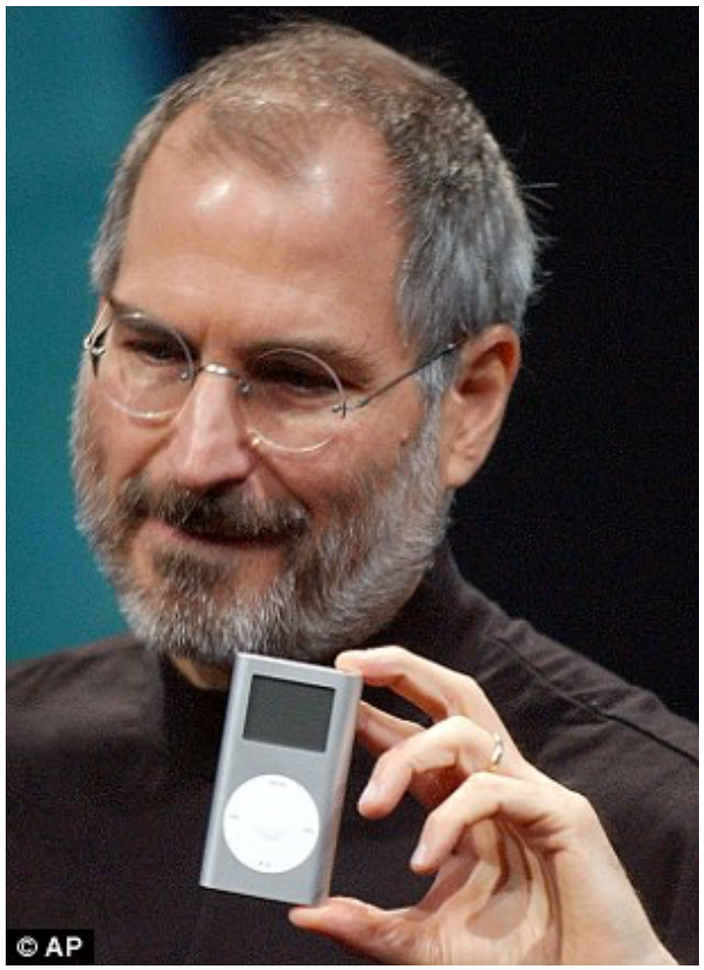
- Shuffle a sequence (e.g. a list)
- Create new sequence drawing at random from original
- The modern algorithm
  - in-place

```
-- To shuffle an array a of n elements (indices 0..n-1):
for i from n−1 downto 1 do
     j ← random integer such that 0 ≤ j ≤ i
     exchange a[j] and a[i]
```

https://en.wikipedia.org/wiki/Fisher–Yates_shuffle

Fun Fact

Steve Jobs
'We're making it (the shuffle) less random to make it feel more random’.



Spotify
"The problem is that, to humans, truly random does not feel random,
…
we updated it with a new algorithm that is intended to feel more random to a human.”

The new algorithm ironically makes the order of songs less random for it to feel more random to listeners.

Steps

1. Implement TimeMS class
2. Store it in a module L15Time

TimeMS
mins: int secs: int
__init__(tstring) get_time() __str__() inc(tuple) reset()

```
t1 = TimeMS('10:30')
In [109]:
t1.get_time()
Out[109]:
(10, 30)
In [112]:
print(t1)
Out[112]:
'10:30'
In [114]:
t1.inc((1,15))
print(t1)
Out[114]:
'11:45'
In [115]:
t1.reset()
print(t1)
Out[115]:
'0:00'
```

Steps

3. Implement the Track class

Track
number: int title: str artist: str time: TimeMS
__init__(name,artist, tsting) __eq__(other) __lt__(other) __str__()

```
lg1 = Track(45,'Lady Gaga','La Vie en Rose','3:00')
lg2 = Track(47,'Lady Gaga','Hair Body Face','3:23')

print(lg1)

  45  Lady Gaga: La Vie en Rose
In [33]:
lg = [lg1,lg2]
for t in lg: print(t)

  45  Lady Gaga: La Vie en Rose
  47  Lady Gaga: Hair Body Face
In [35]:
lg.sort()
for t in lg: print(t)

  47  Lady Gaga: Hair Body Face
  45  Lady Gaga: La Vie en Rose
```

Steps

4. Implement the FullTrackList class

FullTrackList
file_name: str tracks: Tracks list tracks_dict
__init__(file_name) load_csv(‘file_name’) list_tracks()

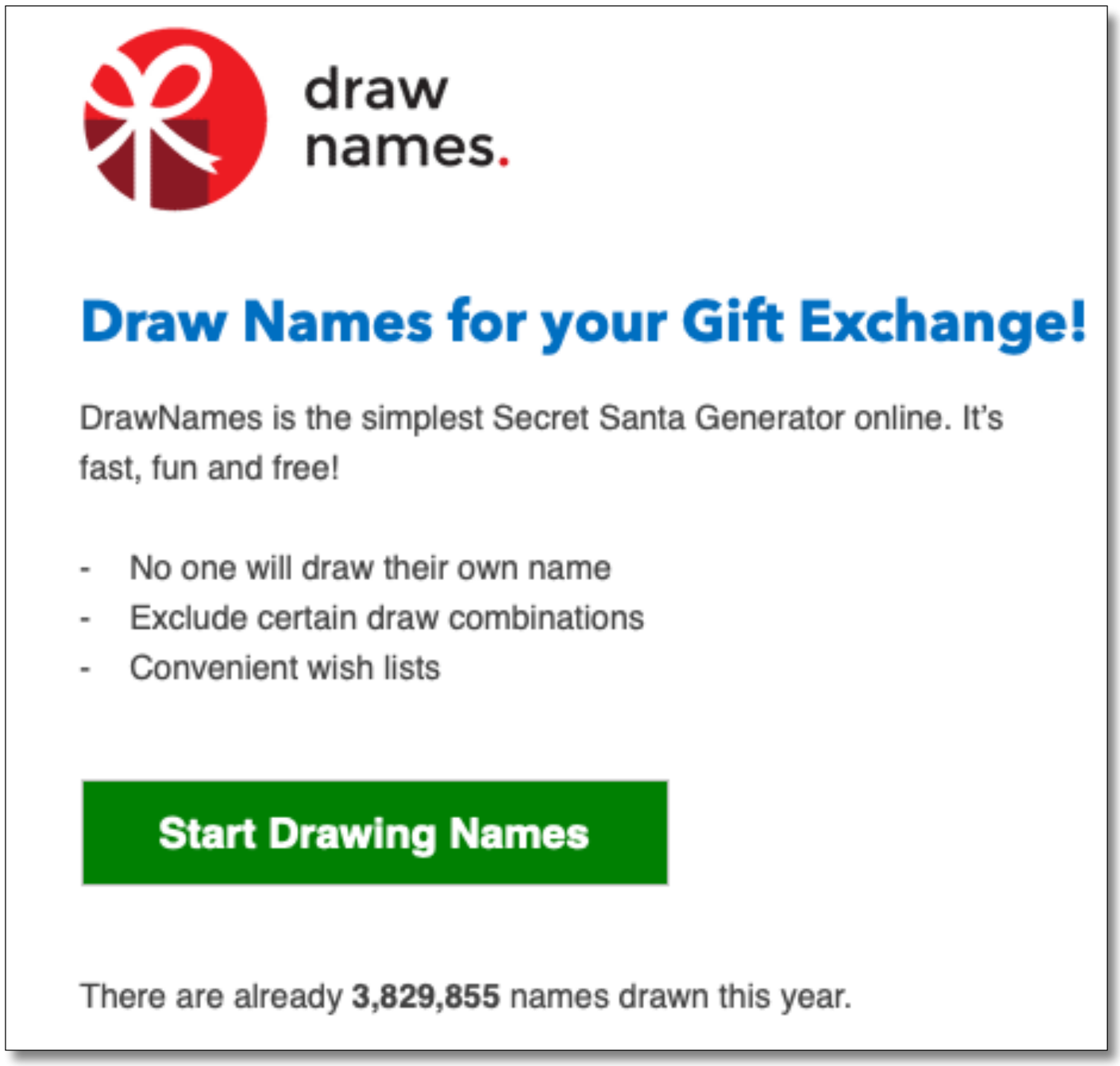
```
tlist = FullTrackList(file_name='music.csv')
print(tlist.tracks_dict['4'])

   4   Mumford and Sons: Babel
In [45]:
tlist.list_tracks()

   1   Taylor Swift: Everything Has Changed
   2   Mumford and Sons: Little Lion Man
   3   Hozier: Sedated
   4   Mumford and Sons: Babel
   5   Taylor Swift: I Knew You Were Trouble
  …
```

Steps

- The main event - Playlist class
  - Constructor

Playlist
tot_time: TimeMS track_ids: tracklist: FullTrackList plist:
__init__(tids,Tracklist) setup_list(tids) play_playlist() add_track(tid) remove_track(tid) sort_playlist() shuffle_playlist() calculate_time()

Secret Santa



- What would be the main objects in a Secret Santa System?