Importing necessary modules

In [39]:
```
import json
import spotipy
import pandas as pd
from spotipy.oauth2 import SpotifyClientCredentials
```

In [40]:
```
client_id = '28364a36677e49c8811b8f78cab9a27f'
client_secret = '5a03d8fdb5234460acf6a8491ae8ee0e'
```

In [41]:
```
client_credentials_manager = SpotifyClientCredentials(client_id, client_secret)
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)
```

In [42]:
```
playlist_id='spotify:playlist:4tNnLzd4WsIvbE58PswNl4'
results = sp.playlist(playlist_id)
```

Creating dataset from existing Spotify playlist

In [43]:
```
# Create a list of song ids
ids = []

for item in results['tracks']['items']:
    track = item['track']['id']
    ids.append(track)

song_meta = {'id':[],'album':[],'name':[],'artist':[],'explicit':[],'popularity':[]}

for song_id in ids:
    # song's meta data
    meta = sp.track(song_id)

    # song id
    song_meta['id'].append(song_id)

    # album name
    album = meta['album']['name']
    song_meta['album'] += [album]

    # song name
    song = meta['name']
    song_meta['name'] += [song]

    # artist's name
    s = ', '
    artist=s.join([singer_name['name'] for singer_name in meta['artists']])
    song_meta['artist']+=[artist]

    # explicit: lyrics could be considered offensive or unsuitable for children
    explicit=meta['explicit']
    song_meta['explicit'].append(explicit)

    # song popularity
    popularity=meta['popularity']
    song_meta['popularity'].append(popularity)

song_meta_df=pd.DataFrame.from_dict(song_meta)

# check the song feature
features = sp.audio_features(song_meta['id'])
# change dictionary to dataframe
features_df=pd.DataFrame.from_dict(features)

# convert milliseconds to mins
# duration_ms: The duration of the track in milliseconds.
# 1 minute = 60 seconds = 60 × 1000 milliseconds = 60,000 ms
features_df['duration_ms']=features_df['duration_ms']/60000

# combine two dataframe
final_df=song_meta_df.merge(features_df)
```

In [44]:
```
final_df
```

Out[44]:

	id	album	name	artist	explicit	popularity	danceability	energy	key	loudness	...	instrumentalne
0	3eR23VReFzcdmS7TYCrhCe	It Ain't Me (with Selena Gomez)	It Ain't Me (with Selena Gomez)	Kygo, Selena Gomez	False	76	0.640	0.533	0	-6.596	...	0.000
1	7qiZfU4dY1lWllzX7mPBI3	÷ (Deluxe)	Shape of You	Ed Sheeran	False	84	0.825	0.652	1	-3.183	...	0.000
2	4iLqG9SeJSnt0cSPICSjxv	Attention	Attention	Charlie Puth	False	7	0.774	0.626	3	-4.432	...	0.000
3	79cuOz3SPQTuFrp8WgfAu	Illuminate (Deluxe)	There's Nothing Holdin' Me Back	Shawn Mendes	False	4	0.857	0.800	2	-4.035	...	0.000
4	0dA2Mk56wEzDgegdC6R17g	Stay	Stay (with Alessia Cara)	Zedd, Alessia Cara	False	2	0.679	0.634	5	-5.024	...	0.000
...	...	...	...	...	...	...	...	...	...	...	...	...
95	48nnRcGWEO5ySlqE17IBYB	Game Of Thrones: Season 7 (Music from the HBO®...	Main Titles	Ramin Djawadi	False	0	0.324	0.766	8	-10.919	...	0.876
96	1j4kHkkpqZRBwE0A4CN4Yv	Dusk Till Dawn (Radio Edit)	Dusk Till Dawn - Radio Edit	ZAYN, Sia	False	82	0.258	0.437	11	-6.593	...	0.000
97	6KdmNK9MogmGcnO3wNZHhp	Funk Wav Bounces Vol.1	Hard to Love (feat. Jessie Reyez)	Calvin Harris, Jessie Reyez, Funk Wav	True	54	0.753	0.785	2	-6.530	...	0.000
98	043MJ8zk9VQLvXXV01UbH6	Game On: 2 Player Mode	Fairy Tail Theme	Taylor Davis, Lara de Wit	False	48	0.289	0.360	5	-13.008	...	0.880
99	3JBUv4SVXFAbZVlnchORpU	You Don't Do It For Me Anymore	You Don't Do It For Me Anymore	Demi Lovato	False	0	0.546	0.421	5	-6.339	...	0.000

100 rows × 23 columns

In [45]:
```
final_df['artist']
```

Out[45]:
```
0                        Kygo, Selena Gomez
1                                Ed Sheeran
2                              Charlie Puth
3                              Shawn Mendes
4                         Zedd, Alessia Cara
                       ...
95                             Ramin Djawadi
96                                 ZAYN, Sia
97    Calvin Harris, Jessie Reyez, Funk Wav
98                 Taylor Davis, Lara de Wit
99                               Demi Lovato
Name: artist, Length: 100, dtype: object
```

In [46]:
```
final_df.shape
```

Out[46]:
```
(100, 23)
```

In [47]:
```
music_feature = features_df[['danceability','energy','loudness','speechiness','acousticness','instrumentalness','liveness','valence','tempo','duration_ms']]
```

In [48]:
```
music_feature.describe()
```

Out[48]:

	danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_m
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.00000
mean	0.640400	0.683370	-6.090910	0.085453	0.164188	0.045691	0.147080	0.515473	118.58673	3.65808
std	0.144982	0.157751	2.808741	0.070467	0.204452	0.187742	0.103024	0.236646	27.37181	0.96232
min	0.213000	0.054000	-23.420000	0.031000	0.000407	0.000000	0.037100	0.057800	60.38700	0.66600
25%	0.564000	0.617500	-6.702250	0.040075	0.037175	0.000000	0.088600	0.376000	99.98300	3.30205
50%	0.649500	0.715000	-5.642000	0.062050	0.081150	0.000001	0.107500	0.486000	118.48450	3.62750
75%	0.739750	0.788750	-4.421500	0.101500	0.213750	0.000046	0.154750	0.706250	127.98375	3.84927
max	0.965000	0.962000	-2.862000	0.425000	0.988000	0.933000	0.653000	0.966000	200.05600	8.90200

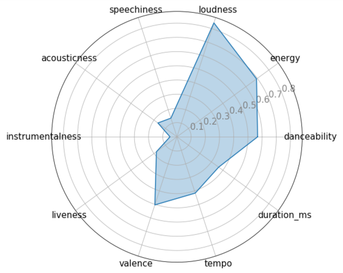
Feature Scaling

In [49]:
```
from sklearn.preprocessing import MinMaxScaler
min_max_scaler = MinMaxScaler()
music_feature.loc[:]=min_max_scaler.fit_transform(music_feature.loc[:])
```

```
C:\Users\Pranav.LAPTOP-HOVCQVL6\anaconda3\lib\site-packages\pandas\core\indexing.py:670: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_with_indexer(indexer, value)
C:\Users\Pranav.LAPTOP-HOVCQVL6\anaconda3\lib\site-packages\ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports until
```

Data Visualization

In [50]:
```
import matplotlib.pyplot as plt
from math import pi

# plot size
fig=plt.figure(figsize=(12,8))

# convert column names into a list
categories=list(music_feature.columns)
# number of categories
N=len(categories)

# create a list with the average of all features
value=list(music_feature.mean())

# repeat first value to close the circle
# the plot is a circle, so we need to "complete the loop"
# and append the start value to the end.
value+=value[:1]
# calculate angle for each category
angles=[n/float(N)*2*pi for n in range(N)]
angles+=angles[:1]

# plot
plt.polar(angles, value)
plt.fill(angles,value,alpha=0.3)



plt.xticks(angles[:-1],categories, size=15)
plt.yticks(color='grey',size=15)
plt.show()
```



From this chart one can tell that I like music that is loud and energy filled. Also the danceability of the songs is a plus.

In [ ]:
```

```