In [1]:
```python
import json
import re
from ast import literal_eval
```

Open json file with lyrics grouped by album

In [2]:
```python
with open('album_lyrics_1.json', 'r') as f:
    album_lyrics_obj = json.load(f)
album_lyrics_obj
```

...

Ran previously to delete unnecessary and duplicate elements, and combine album lyrics

In [ ]:
```python
for album, songs in album_lyrics_obj.items():
    if album == "taylor swift":
        del songs[9:18]
    if album == "speak now":
        del songs[11]
    if album == "red deluxe edition":
        del songs[17:21]
    if album == "1989 deluxe":
        del songs[12:15]
    if album == "reputation":
        del songs[14:16]

for album, songs in album_lyrics_obj.items():
    album_lyrics = []
    for song in songs:
        for word in song:
            album_lyrics.append(word)
    album_lyrics_obj[album] = album_lyrics

with open("album_lyrics_1.json", 'w') as f:
        json.dump(album_lyrics_obj, f, indent=4)
```

Define function to split camelCase lyric string

In [3]:
```
def split_camelcase(lyric):
    return re.findall(r'[A-Z]?[a-z]+|[A-Z]+(?=[A-Z]|$)', lyric)
```

Test the function to split camelCase string

In [4]:
```
print(split_camelcase('camelCaseXYZ'))
```

```
['camel', 'Case', 'XYZ']
```

Run function to split camelCase lyric strings

In [5]:
```
for album, lyrics in album_lyrics_obj.items():
    new_lyrics = []
    lyric = [lyric.replace(lyric, str(split_camelcase(lyric))) for lyric in lyrics]
    new_lyrics.extend(lyric)
    album_lyrics_obj[album] = new_lyrics
```

In [6]:
```
album_lyrics_obj
```

...

In [7]:
```
from ast import literal_eval
literal_eval("[1, 2, 3]")
```

Out[7]:
```
[1, 2, 3]
```

Remove double quotation marks

In [8]:
```
for album, lyrics in album_lyrics_obj.items():
    new_lyrics = []
    for lyric in lyrics:
        lyric = literal_eval(lyric)
        new_lyrics.append(lyric)
        album_lyrics_obj[album] = new_lyrics
```

```
In [9]:  album_lyrics_obj
```

```
...
```

Merge individual word lists to create one list for each album

```
In [10]: for album, lyrics in album_lyrics_obj.items():
             new_lyrics = []
             for lyric in lyrics:
                 for word in lyric:
                     new_lyrics.append(word)
                     album_lyrics_obj[album] = new_lyrics
```

```
In [11]: album_lyrics_obj
```

```
...
```

Save new json file with lyrics grouped by album

```
In [12]: with open("album_lyrics_2.json", 'w') as f:
                 json.dump(album_lyrics_obj, f, indent=4)
```

Define function to convert list of lyrics to string

```
In [13]: def list_to_string(lyrics):
             text = " "
             return (text.join(lyrics).lower())
```

Test the function to convert list of lyrics to string

```
In [14]: list_to_string(['Hello', 'good', 'morning'])
Out[14]: 'hello good morning'
```

In [15]:

```python
import os
from os import path
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
```

Define function to make wordcloud

In [16]:

```python
def make_wordcloud(album, lyrics, mask, color):
    stopwords = set(STOPWORDS)
    stopwords.update(["di", "n't", "oh", "ooh", "ai", "oooh", "wo", "mmmm", "mmmmmmm", "lyrics", "ch

    text = list_to_string(lyrics)
    wordcloud = WordCloud(font_path="Arial", color_func=lambda *args, **kwargs: color, min_font_size
    wordcloud.generate(text)

    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.figure()

    filename = album + "_wordcloud_1.png"
    wordcloud.to_file("/Users/lindsaytubbs/Documents/GitHub/ts-lyrics/" + filename )
```

Create an object to specify attributes for make_wordcloud function

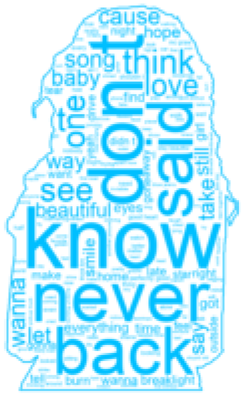
In [17]:

```python
matches = [{"mask": "selftitled", "album": "taylor swift", "color": "deepskyblue"},
            {"mask": "fearless", "album": "fearless", "color": "darkgoldenrod"},
            {"mask": "speaknow", "album": "speak now", "color": "darkmagenta"},
            {"mask": "red", "album": "red deluxe edition", "color": "darkred"},
            {"mask": "1989", "album": "1989 deluxe", "color": "blueviolet"},
            {"mask": "reputation", "album": "reputation", "color": "black"},
            {"mask": "lover", "album": "lover", "color": "hotpink"},
            {"mask": "folklore", "album": "folklore", "color": "dimgray"},
            {"mask": "evermore", "album": "evermore", "color": "forestgreen"}
            ]
```

Run function to make wordclouds for each element in the object

In [18]:

```
for match in matches:
    mask = np.array(Image.open("/Users/lindsaytubbs/Documents/python_projects/" + match["mask"] + "_
    make_wordcloud(album=match["album"], lyrics=album_lyrics_obj[match["album"]], mask=mask, color=r
```



fairytale
take
feel
way
dont
cause
know
back
thing