In [1]:
```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException
from IPython.display import clear_output
import pandas as pd
```
executed in 302ms, finished 15:44:36 2021-02-21

In [2]:
```
# set up selenium webscraping browser

option = webdriver.ChromeOptions()
option.add_argument('incognito')

browser = webdriver.Chrome(options=option)
```
executed in 1.17s, finished 15:44:37 2021-02-21

In [3]:

```
# scrape esrb.org for game information

pg = 1
games_list = []

while True:
    browser.get("https://www.esrb.org/search/?searchKeyword=" \
                "&platform=Nintendo%20Switch%2CPlayStation%204%2CXbox%20One" \
                "&rating=E%2CE10%2B%2CT%2CM%2CAO&descriptor=All%20Content" \
                "&pg={}&searchType=All".format(pg))
    try:
        # make selenium wait until games are loaded before moving on
        element = WebDriverWait(browser, 10).until(
            EC.presence_of_element_located((By.CLASS_NAME, 'game')))

        # games on current page
        results = browser.find_elements_by_xpath("//div[@class='game']")

        titles, consoles, ratings, descriptors = [], [], [], []

        # pull relevant content from results
        for x in results:
            # titles
            title = x.find_element_by_css_selector('h2').text
            titles.append(title)

            # consoles
            console = x.find_element_by_class_name('platforms').text
            consoles.append(console)

            # ratings are displayed on the page as an image,
            # so we have to pull the rating from the image's url
            xpath = x.find_element_by_css_selector('img')
            rating = xpath.get_attribute('src')[58:-4]
            ratings.append(rating)

            # descriptors
            desc = x.find_elements_by_css_selector('td')[1].text
            descriptors.append(desc)

        # collect the results as a list of dictionaries
        for i in range(len(titles)):
            game_dict = {}
            game_dict['title'] = titles[i]
            game_dict['consoles'] = consoles[i]
            game_dict['rating'] = ratings[i]
            game_dict['descriptors'] = descriptors[i]
            games_list.append(game_dict)

        # progress indicator
        clear_output(wait=True)
        print('Page: {}'.format(pg))
        pg += 1

    # stop running when no more games are found
    except:
        print('Stopped on page {}'.format(pg))
        break
```

executed in 17m 59s, finished 16:02:35 2021-02-21

```
Page: 504
Stopped on page 505
```

In [4]:
```
# preview of games_list
games_list[:5]
```

executed in 13ms, finished 16:02:35 2021-02-21

Out[4]:
```
[{'title': 'Blizzard Arcade Collection',
  'consoles': 'Windows PC, PlayStation 4, Nintendo Switch, Xbox One',
  'rating': 'T',
  'descriptors': 'Blood, Fantasy Violence, Language, Use of Tobacco'},
 {'title': 'Rez Infinite',
  'consoles': 'PlayStation 4',
  'rating': 'E10plus',
  'descriptors': 'Fantasy Violence'},
 {'title': 'Hotshot Racing',
  'consoles': 'PlayStation 4, Nintendo Switch',
  'rating': 'E10plus',
  'descriptors': 'Alcohol Reference, Language, Mild Violence'},
 {'title': "Sea of Solitude : The Director's Cut",
  'consoles': 'Nintendo Switch',
  'rating': 'T',
  'descriptors': 'Fantasy Violence, Language'},
 {'title': 'Ape Out',
  'consoles': 'Nintendo Switch',
  'rating': 'T',
  'descriptors': 'Blood and Gore, Violence'}]
```

In [5]:
```
# dataframe of collected data
df = pd.DataFrame(games_list)
df.head()
```

executed in 30ms, finished 16:02:35 2021-02-21

Out[5]:

	title	consoles	rating	descriptors
0	Blizzard Arcade Collection	Windows PC, PlayStation 4, Nintendo Switch, Xb...	T	Blood, Fantasy Violence, Language, Use of Tobacco
1	Rez Infinite	PlayStation 4	E10plus	Fantasy Violence
2	Hotshot Racing	PlayStation 4, Nintendo Switch	E10plus	Alcohol Reference, Language, Mild Violence
3	Sea of Solitude : The Director's Cut	Nintendo Switch	T	Fantasy Violence, Language
4	Ape Out	Nintendo Switch	T	Blood and Gore, Violence

In [6]:
```
df.info()
```

executed in 13ms, finished 16:02:35 2021-02-21

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5034 entries, 0 to 5033
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   title        5034 non-null   object
 1   consoles     5034 non-null   object
 2   rating       5034 non-null   object
 3   descriptors  5034 non-null   object
dtypes: object(4)
memory usage: 157.4+ KB
```

In [7]:
```
# split consoles and descriptors columns
df.descriptors = df.descriptors.map(lambda x: x.split(', '))
df.consoles = df.consoles.map(lambda x: x.split(', '))

df.head()
```

executed in 15ms, finished 16:02:35 2021-02-21

Out[7]:

	title	consoles	rating	descriptors
0	Blizzard Arcade Collection	[Windows PC, PlayStation 4, Nintendo Switch, X...	T	[Blood, Fantasy Violence, Language, Use of Tob...
1	Rez Infinite	[PlayStation 4]	E10plus	[Fantasy Violence]
2	Hotshot Racing	[PlayStation 4, Nintendo Switch]	E10plus	[Alcohol Reference, Language, Mild Violence]
3	Sea of Solitude : The Director's Cut	[Nintendo Switch]	T	[Fantasy Violence, Language]
4	Ape Out	[Nintendo Switch]	T	[Blood and Gore, Violence]

In [8]:
```
# choose consoles
consoles_list = ['PlayStation 4', 'Xbox One', 'Nintendo Switch']
for i in range(len(df)):
    df.consoles[i] = [x for x in df.consoles[i] if x in consoles_list]

df.head()
```

executed in 424ms, finished 16:02:36 2021-02-21

Out[8]:

	title	consoles	rating	descriptors
0	Blizzard Arcade Collection	[PlayStation 4, Nintendo Switch, Xbox One]	T	[Blood, Fantasy Violence, Language, Use of Tob...
1	Rez Infinite	[PlayStation 4]	E10plus	[Fantasy Violence]
2	Hotshot Racing	[PlayStation 4, Nintendo Switch]	E10plus	[Alcohol Reference, Language, Mild Violence]
3	Sea of Solitude : The Director's Cut	[Nintendo Switch]	T	[Fantasy Violence, Language]
4	Ape Out	[Nintendo Switch]	T	[Blood and Gore, Violence]

In [9]:
```
# rearrange dataframe to put target variable at the end
df = df[['title', 'consoles', 'descriptors', 'rating']]
df.head()
```

executed in 13ms, finished 16:02:36 2021-02-21

Out[9]:

	title	consoles	descriptors	rating
0	Blizzard Arcade Collection	[PlayStation 4, Nintendo Switch, Xbox One]	[Blood, Fantasy Violence, Language, Use of Tob...	T
1	Rez Infinite	[PlayStation 4]	[Fantasy Violence]	E10plus
2	Hotshot Racing	[PlayStation 4, Nintendo Switch]	[Alcohol Reference, Language, Mild Violence]	E10plus
3	Sea of Solitude : The Director's Cut	[Nintendo Switch]	[Fantasy Violence, Language]	T
4	Ape Out	[Nintendo Switch]	[Blood and Gore, Violence]	T

In [11]:
```
# pickle it

df.to_pickle('esrb_ratings.pkl')
```

executed in 20ms, finished 16:08:50 2021-02-21