

In [1]:

```
#HTML을 파이썬으로 가져오기 : requests
import requests
from bs4 import BeautifulSoup

#특정 주소 naver 서버에 대화를 시도
response = requests.get('https://movie.naver.com/movie/running/current.naver')

#naver 에서 html 가져오기
html = response.text

#html.parser (html 번역기로 해석)
soup = BeautifulSoup(html, 'html.parser')

#html 출력
print(soup)
```

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta content="no" http-equiv="imagetoolbar"/>
<title>현재 상영영화 : 네이버 영화</title>
<meta content="https://ssl.pstatic.net/imgmovie/today/naverme/naverme_profile.jpg"
property="me2:image">
<meta content="네이버영화 현재상영작" property="me2:post_tag">
<meta content="네이버영화" property="me2:category1">
<meta content="현재상영작" property="me2:category2"/>
<meta content="현재상영작 : 네이버 영화" property="og:title"/>
<meta content="article" property="og:type"/>
<meta content="https://movie.naver.com/movie/running/current.naver" property="og:url"/>
<meta content="https://ssl.pstatic.net/static/m/movie/icons/OG_270_270.png" proper
... "..." />
```

In [2]:

```
rating = soup.select("dd:nth-child(2) > div > a > span.num")
print(rating)
```

```
[<span class="num">6.59</span>, <span class="num">9.44</span>, <span class="num">8.18</span>, <span class="num">7.74</span>, <span class="num">8.05</span>, <span class="num">8.41</span>, <span class="num">0.00</span>, <span class="num">8.57</span>, <span class="num">9.44</span>, <span class="num">7.70</span>, <span class="num">3.91</span>, <span class="num">9.55</span>, <span class="num">9.02</span>, <span class="num">9.28</span>, <span class="num">7.93</span>, <span class="num">8.91</span>, <span class="num">7.60</span>, <span class="num">9.78</span>, <span class="num">8.91</span>, <span class="num">8.63</span>, <span class="num">9.03</span>, <span class="num">9.38</span>, <span class="num">6.88</span>, <span class="num">6.52</span>, <span class="num">6.69</span>, <span class="num">7.62</span>, <span class="num">7.69</span>, <span class="num">6.42</span>, <span class="num">8.88</span>, <span class="num">5.52</span>, <span class="num">8.48</span>, <span class="num">7.95</span>, <span class="num">8.10</span>, <span class="num">7.43</span>, <span class="num">6.24</span>, <span class="num">9.12</span>, <span class="num">8.32</span>, <span class="num">7.07</span>, <span class="num">8.22</span>, <span class="num">7.25</span>, <span class="num">9.44</span>, <span class="num">9.43</span>, <span class="num">8.82</span>, <span class="num">8.15</span>, <span class="num">7.27</span>, <span class="num">7.81</span>, <span class="num">8.90</span>, <span class="num">7.54</span>, <span class="num">9.24</span>, <span class="num">8.40</span>, <span class="num">7.12</span>, <span class="num">9.13</span>, <span class="num">8.12</span>, <span class="num">8.75</span>, <span class="num">8.95</span>, <span class="num">7.67</span>, <span class="num">9.24</span>, <span class="num">7.44</span>, <span class="num">8.46</span>, <span class="num">8.85</span>, <span class="num">8.52</span>, <span class="num">8.60</span>, <span class="num">7.74</span>, <span class="num">8.32</span>, <span class="num">7.23</span>, <span class="num">8.57</span>, <span class="num">8.63</span>, <span class="num">8.31</span>, <span class="num">9.04</span>, <span class="num">7.99</span>, <span class="num">7.36</span>, <span class="num">6.33</span>, <span class="num">7.95</span>, <span class="num">7.88</span>, <span class="num">9.18</span>, <span class="num">8.10</span>, <span class="num">7.72</span>, <span class="num">8.76</span>, <span class="num">7.64</span>, <span class="num">8.36</span>, <span class="num">9.26</span>, <span class="num">8.50</span>, <span class="num">8.94</span>]
```

In [3]:

```
for number in rating:  
    print(number.text)
```

6.59
9.44
8.18
7.74
8.05
8.41
0.00
8.57
9.44
7.70
3.91
9.55
9.02
9.28
7.93
8.91
7.60
9.78
8.91
8.63
9.03
9.38
6.88
6.52
6.69
7.62
7.69
6.42
8.88
5.52
8.48
7.95
8.10
7.43
6.24
9.12
8.32
7.07
8.22
7.25
9.44
9.43
8.82
8.15
7.27
7.81
8.90
7.54
9.24
8.40
7.12
9.13
8.12
8.75
8.95
7.67

9.24
7.44
8.46
8.85
8.52
8.60
7.74
8.32
7.23
8.57
8.63
8.31
9.04
7.99
7.36
6.33
7.95
7.88
9.18
8.10
7.72
8.76
7.64
8.36
9.26
8.50
8.94

In [4]:

```
for i in range(10):  
    print(rating[i].text)
```

6.59
9.44
8.18
7.74
8.05
8.41
0.00
8.57
9.44
7.70

In [5]:

```
name = soup.select("dl > dt > a")
print(name)
```

[토르: 러브 앤 썬더, 빅샤크4 : 바다공룡 대모험, 큐어, 컴온 컴온, 프리! 더 파이널 스트로크 후편, 보통의 용기, 눈감은 여름, 헤어질 결심, 위대한 침묵, 미친 능력, 고스트랜드, 니얼굴, 극장판 월벤저스 : 수상한 캠핑 대소동, 모어, 우스운게 딱! 좋아!, 디어 헌터, 데쓰 캘린더, 탐견: 매버릭, 룸 쉐어링, 감동주의보, 경아의 딸, 그대라는 기억 연속씨, 뱀: 비컴 휴먼, 아부쟁이, 마녀(魔女) Part2. The Other One, 버즈 라이트이어, 실종, 더 렛지, 베르네 부인의 장미정원, 브로커, 윤시내가 사라졌다, 올리 마키의 가장 행복한 날, 애프터 양, 극장판 포켓몬스터DP: 기라티나와 하늘의 꽃다발 쉐이미, 주라기 월드: 도미니언, 카시오페아, 오마주, 그대가 조국, 플레이그라운드, 더 노비스, 범죄도시2, 아치의 노래, 정태춘, 나를 만나는 길, 우연과 상상, 닥터 스트레인지: 대혼돈의 멀티버스, 소설가의 영화, 중경삼림, 레드로켓, 극장판 주술회전 0, 드라이브 마이 카, 호수의 이방인, 코다, 릴리 슈슈의 모든 것, 스쿨 오브 락(樂), 남매의 여름밤, 모리의 정원, 피아니스트의 전설, 지구 최후의 밤, 녹색 광선, 천로역정: 천국을 찾아서, 그녀, 막다른 골목의 추억, 철벽선생, 신은 죽지 않았다 3: 어둠 속의 빛, 은혼, 분노, 걸어도 걸어도, 순응자, 올드보이, 마스터, 신은 죽지 않았다 3: 어둠 속의 빛]

er?code=95251">홀리 모터스, 박쥐, 밀양, 성냥공장 소녀, 슬픔은 그대 가슴에, 여름 이야기, 트윈 픽스, 펠프 픽션, 7년만의 외출, 말 없는 사나이, 로마의 휴일, 율리시즈, 레베카]

In [6]:

```
for title in name:  
    print(title.text)
```

토르: 러브 앤 썬더
빅샤크4 : 바다공룡 대모험
큐어
컴온 컴온
프리! 더 파이널 스트로크 후편
보통의 용기
눈감은 여름
헤어질 결심
위대한 침묵
미친 능력
고스트랜드
니얼굴
극장판 월벤저스 : 수상한 캠핑 대소동
모어
우스운게 딱! 좋아!
디어 헌터
데쓰 캘린더
탐건: 매버릭
룸 쉐어링
감동주의보
경아의 딸
그대라는기억 연속씨
뽀 : 비컴 휴먼
아부쟁이
마녀(魔女) Part2. The Other One
버즈 라이트이어
실종
더 렛지
베르네 부인의 장미정원
브로커
윤시내가 사라졌다
올리 마키의 가장 행복한 날
애프터 양
극장판 포켓몬스터DP: 기라티나와 하늘의 꽃다발 쉐이미
쥬라기 월드: 도미니언
카시오페아
오마주
그대가 조국
플레이그라운드
더 노비스
범죄도시2
아치의 노래, 정태춘
나를 만나는 길
우연과 상상
닥터 스트레인지: 대혼돈의 멀티버스
소설가의 영화
중경삼림
레드 로켓
극장판 주술회전 0
드라이브 마이 카
호수의 이방인
코다
릴리 슈슈의 모든 것
스쿨 오브 락(樂)
남매의 여름밤
모리의 정원

피아니스트의 전설
지구 최후의 밤
녹색 광선
천로역정: 천국을 찾아서
그녀
막다른 골목의 추억
철벽선생
신은 죽지 않았다 3: 어둠 속의 빛
은혼
분노
걸어도 걸어도
순응자
올드보이
마스터
홀리 모터스
박쥐
밀양
성냥공장 소녀
슬픔은 그대 가슴에
여름 이야기
트윈 픽스
펄프 픽션
7년만의 외출
말 없는 사나이
로마의 휴일
율리시즈
레베카

In [7]:

```
for i in range(10):  
    print(name[i].text)
```

토르: 러브 앤 썬더
빅샤크4 : 바다공룡 대모험
큐어
컴온 컴온
프리! 더 파이널 스트로크 후편
보통의 용기
눈감은 여름
헤어질 결심
위대한 침묵
미친 능력

In [8]:

```
for i in range(0,10):  
    print("%d. %s / %s" %(i+1,name[i].text, rating[i].text))
```

1. 토르: 러브 앤 썬더 / 6.59
2. 빅샤크4 : 바다공룡 대모험 / 9.44
3. 큐어 / 8.18
4. 컴온 컴온 / 7.74
5. 프리! 더 파이널 스트로크 후편 / 8.05
6. 보통의 용기 / 8.41
7. 눈감은 여름 / 0.00
8. 헤어질 결심 / 8.57
9. 위대한 침묵 / 9.44
10. 미친 능력 / 7.70

In [9]:

```
import pandas as pd
```

In [10]:

```
mName=[]
mRating=[]
rank_list=[]

for i in range(10):
    mName.append(name[i].text)
    mRating.append(rating[i].text)
    #rank_list.append(i+1)
```

In [11]:

```
MovieDataSet=list(zip(mName,mRating))
df = pd.DataFrame(data = MovieDataSet, columns=['영화제목', '평점'])
df[0:10]
```

Out[11]:

	영화제목	평점
0	토르: 러브 앤 썬더	6.59
1	빅샤크4 : 바다공룡 대모험	9.44
2	큐어	8.18
3	컴온 컴온	7.74
4	프리! 더 파이널 스트로크 후편	8.05
5	보통의 용기	8.41
6	눈감은 여름	0.00
7	헤어질 결심	8.57
8	위대한 침묵	9.44
9	미친 능력	7.70

In [12]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sb
import numpy as np
```

In [47]:

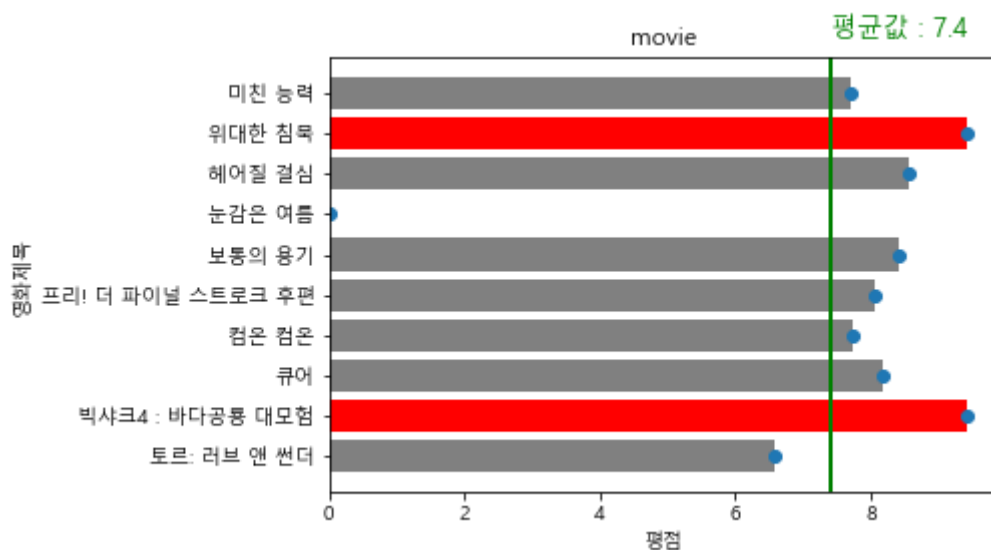
```
plt.rc("font", family="Malgun Gothic")

x = df['평점']
y = df['영화제목']
df['평점'] = pd.to_numeric(df['평점']) # 문자열인 '평점' 데이터를 실수형으로 변환

avrg = df.groupby(['영화제목'])['평점'].sum().mean()
print(avrg)
avrg1 = "{}".format(round(avrg, 1))
plt.axvline(avrg, linewidth=2, color='green')
plt.text(avrg, avrg*1.4, f'평균값 : {avrg1}', fontsize=13, color='green')

colors = ['r' if (bar == max(df['평점'])) else 'grey' for bar in df['평점']]
plt.barh(y, x, align='center', color=colors) # 막대그래프
plt.scatter(x, y) # 산점도
plt.xlabel('평점') # x축 레이블 표기
# plt.xticks(range(11), range(11))
plt.ylabel('영화제목') # y축 레이블 표기
plt.title('movie') # 타이틀
plt.show()
```

7.412000000000001



In []:

