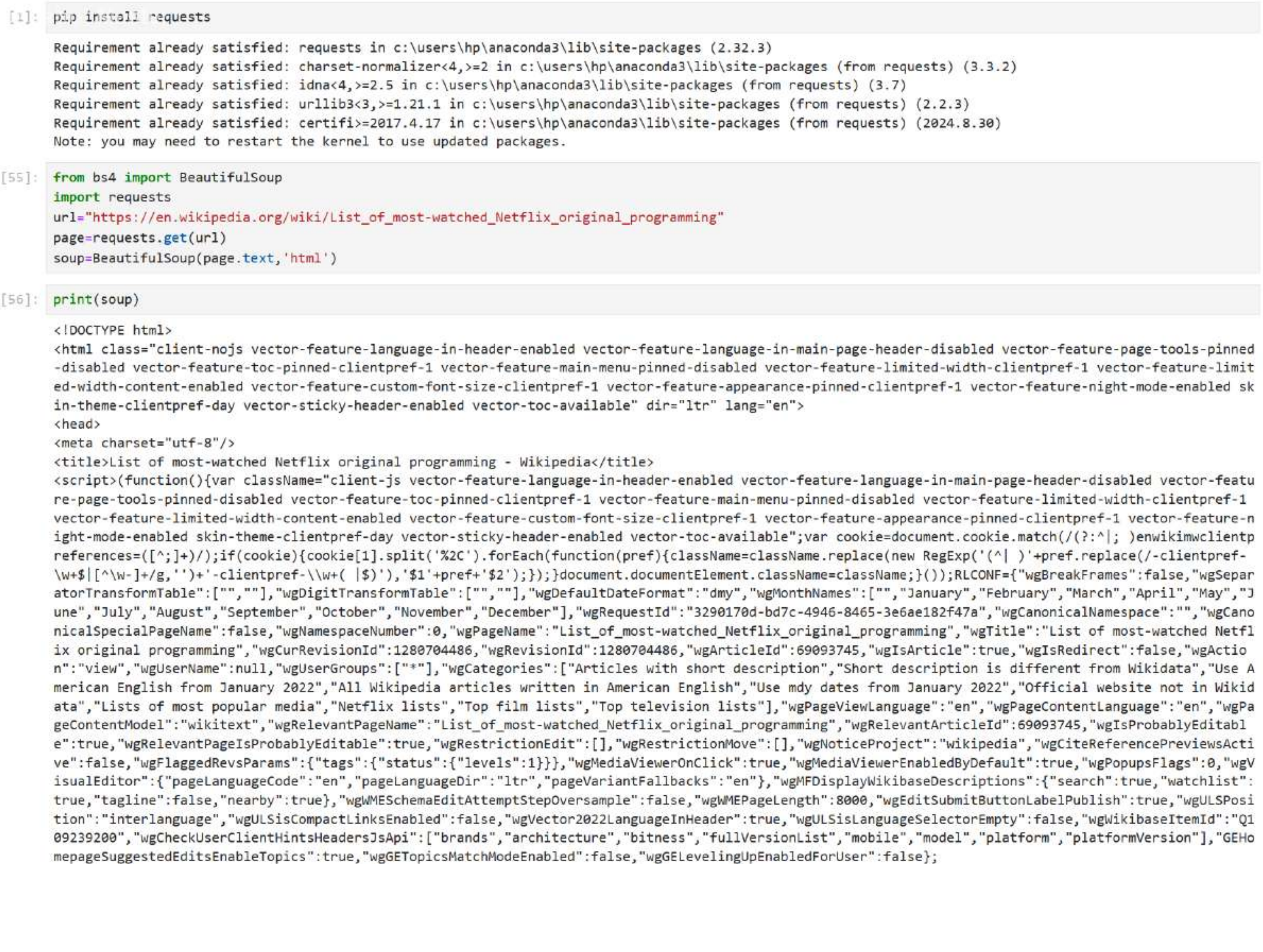


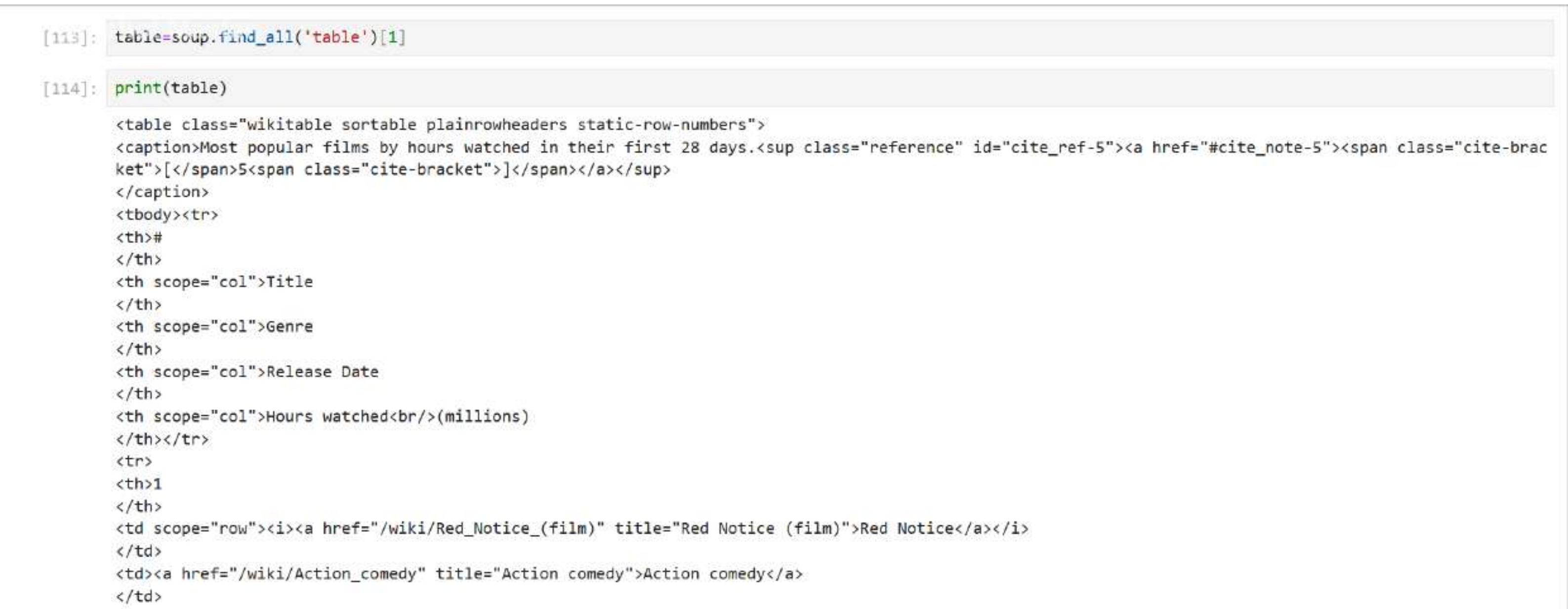
```
[1]: pip install requests
```

```
Requirement already satisfied: requests in c:\users\hp\anaconda3\lib\site-packages (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hp\anaconda3\lib\site-packages (from requests) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests) (2024.8.30)
Note: you may need to restart the kernel to use updated packages.
```

```
[55]: from bs4 import BeautifulSoup
      import requests
      url="https://en.wikipedia.org/wiki/List_of_most-watched_Netflix_original_programming"
      page=requests.get(url)
      soup=BeautifulSoup(page.text,'html')
```

```
[56]: print(soup)
```

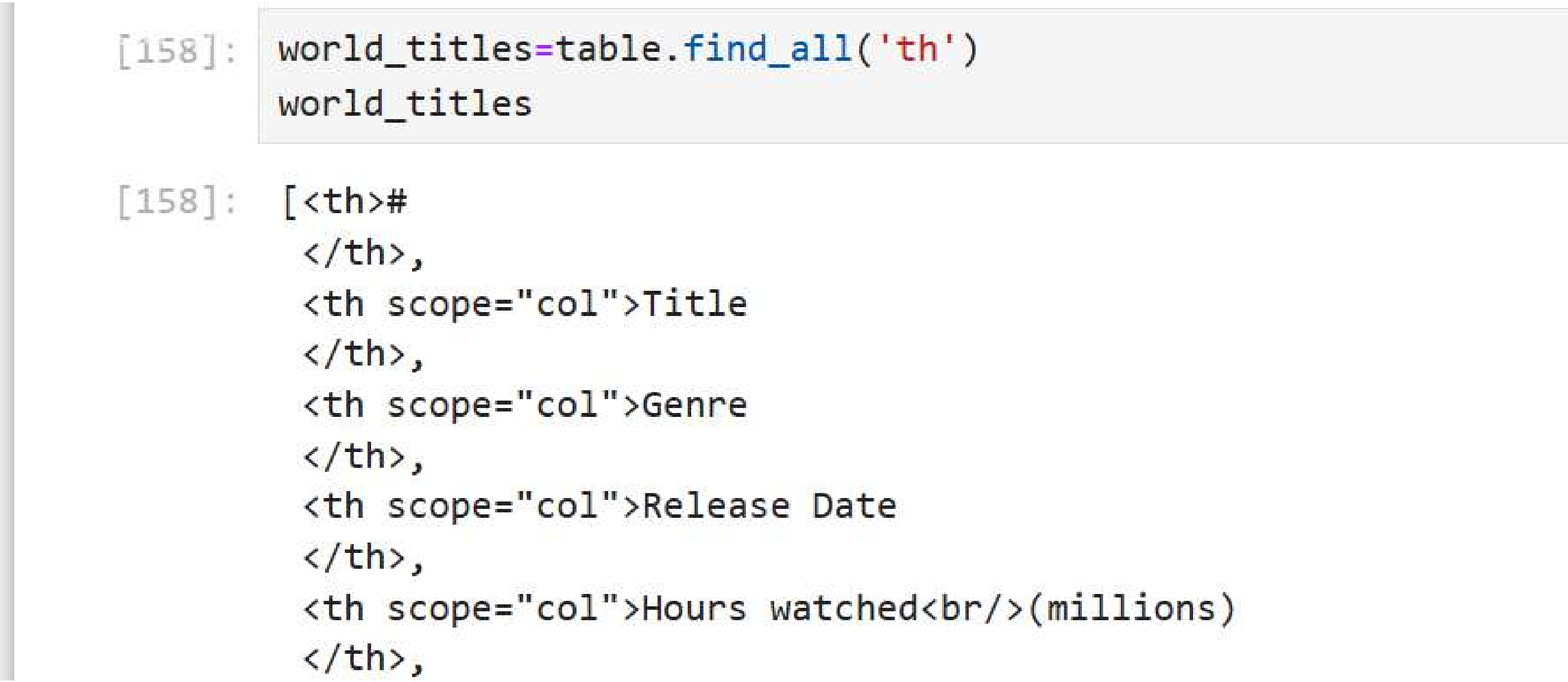
```
<!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-feature-page-tools-pinned
-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limit
ed-width-content-enabled vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1 vector-feature-night-mode-enabled sk
in-theme-clientpref-day vector-sticky-header-enabled vector-toc-available" dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>List of most-watched Netflix original programming - Wikipedia</title>
<script>(function(){var className="client-js vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-featu
re-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1
vector-feature-limited-width-content-enabled vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1 vector-feature-n
ight-mode-enabled skin-theme-clientpref-day vector-sticky-header-enabled vector-toc-available";var cookie=document.cookie.match(/(?:^|; )enwikimwclientp
references=([^;]+)/);if(cookie){cookie[1].split('%2C').forEach(function(pref){className=className.replace(new RegExp('(^| )'+pref.replace(/-clientpref-
\w+$|[^\w-]+/g,'')+'-clientpref-\\w+( |$)'),'$1'+pref+'$2');});}document.documentElement.className=className;}());RLCONF={"wgBreakFrames":false,"wgSepar
atorTransformTable":["",""],"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","March","April","May","J
une","July","August","September","October","November","December"],"wgRequestId":"3290170d-bd7c-4946-8465-3e6ae182f47a","wgCanonicalNamespace":"","wgCano
nicalSpecialPageName":false,"wgNamespaceNumber":0,"wgPageName":"List_of_most-watched_Netflix_original_programming","wgTitle":"List of most-watched Netfl
ix original programming","wgCurRevisionId":1280704486,"wgRevisionId":1280704486,"wgArticleId":69093745,"wgIsArticle":true,"wgIsRedirect":false,"wgActio
n":"view","wgUserName":null,"wgUserGroups":["*"],"wgCategories":["Articles with short description","Short description is different from Wikidata","Use A
merican English from January 2022","All Wikipedia articles written in American English","Use mdy dates from January 2022","Official website not in Wikid
ata","Lists of most popular media","Netflix lists","Top film lists","Top television lists"],"wgPageViewLanguage":"en","wgPageContentLanguage":"en","wgPa
geContentModel":"wikitext","wgRelevantPageName":"List_of_most-watched_Netflix_original_programming","wgRelevantArticleId":69093745,"wgIsProbablyEditabl
e":true,"wgRelevantPageIsProbablyEditable":true,"wgRestrictionEdit":[],"wgRestrictionMove":[],"wgNoticeProject":"wikipedia","wgCiteReferencePreviewsActi
ve":false,"wgFlaggedRevsParams":{"tags":{"status":{"levels":1}}},"wgMediaViewerOnClick":true,"wgMediaViewerEnabledByDefault":true,"wgPopupsFlags":0,"wgV
isualEditor":{"pageLanguageCode":"en","pageLanguageDir":"ltr","pageVariantFallbacks":"en"},"wgMFDisplayWikibaseDescriptions":{"search":true,"watchlist":
true,"tagline":false,"nearby":true},"wgWMESchemaEditAttemptStepOversample":false,"wgWMEPageLength":8000,"wgEditSubmitButtonLabelPublish":true,"wgULSPosi
tion":"interlanguage","wgULSisCompactLinksEnabled":false,"wgVector2022LanguageInHeader":true,"wgULSisLanguageSelectorEmpty":false,"wgWikibaseItemId":"Q1
09239200","wgCheckUserClientHintsHeadersJsApi":["brands","architecture","bitness","fullVersionList","mobile","model","platform","platformVersion"],"GEHo
mepageSuggestedEditsEnableTopics":true,"wgGETopicsMatchModeEnabled":false,"wgGELevelingUpEnabledForUser":false};
```



```
[113]: table=soup.find_all('table')[1]
```

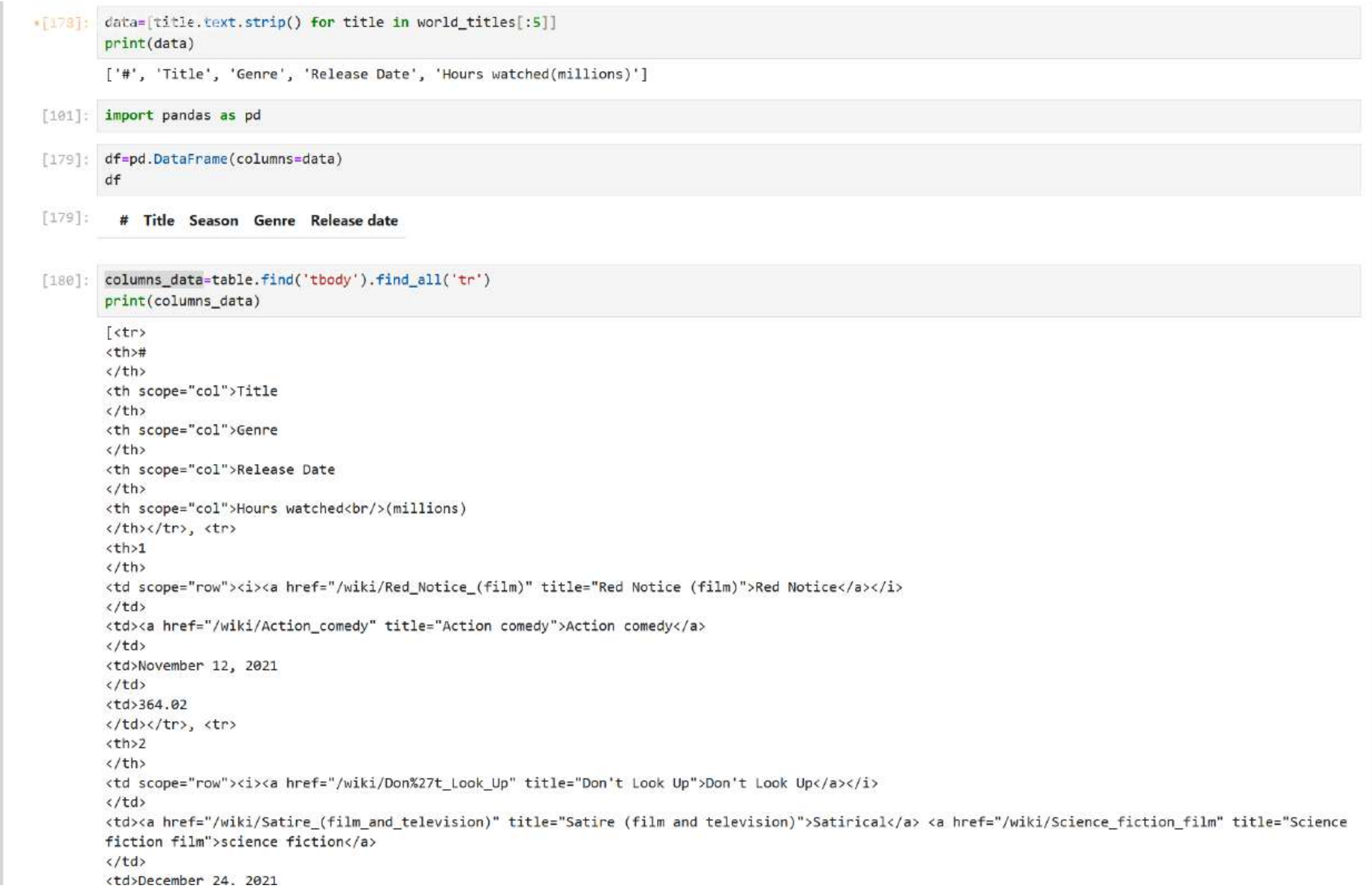
```
[114]: print(table)
```

```
<table class="wikitable sortable plainrowheaders static-row-numbers">
<caption>Most popular films by hours watched in their first 28 days.<sup class="reference" id="cite_ref-5"><a href="#cite_note-5"><span class="cite-brac
ket">[</span>5<span class="cite-bracket">]</span></a></sup>
</caption>
<tbody><tr>
<th>#
</th>
<th scope="col">Title
</th>
<th scope="col">Genre
</th>
<th scope="col">Release Date
</th>
<th scope="col">Hours watched<br/>(millions)
</th></tr>
<tr>
<th>1
</th>
<td scope="row"><i><a href="/wiki/Red_Notice_(film)" title="Red Notice (film)">Red Notice</a></i>
</td>
<td><a href="/wiki/Action_comedy" title="Action comedy">Action comedy</a>
</td>
```



```
[158]: world_titles=table.find_all('th')
       world_titles
```

```
[158]: [<th>#
        </th>,
        <th scope="col">Title
        </th>,
        <th scope="col">Genre
        </th>,
        <th scope="col">Release Date
        </th>,
        <th scope="col">Hours watched<br/>(millions)
        </th>,
```

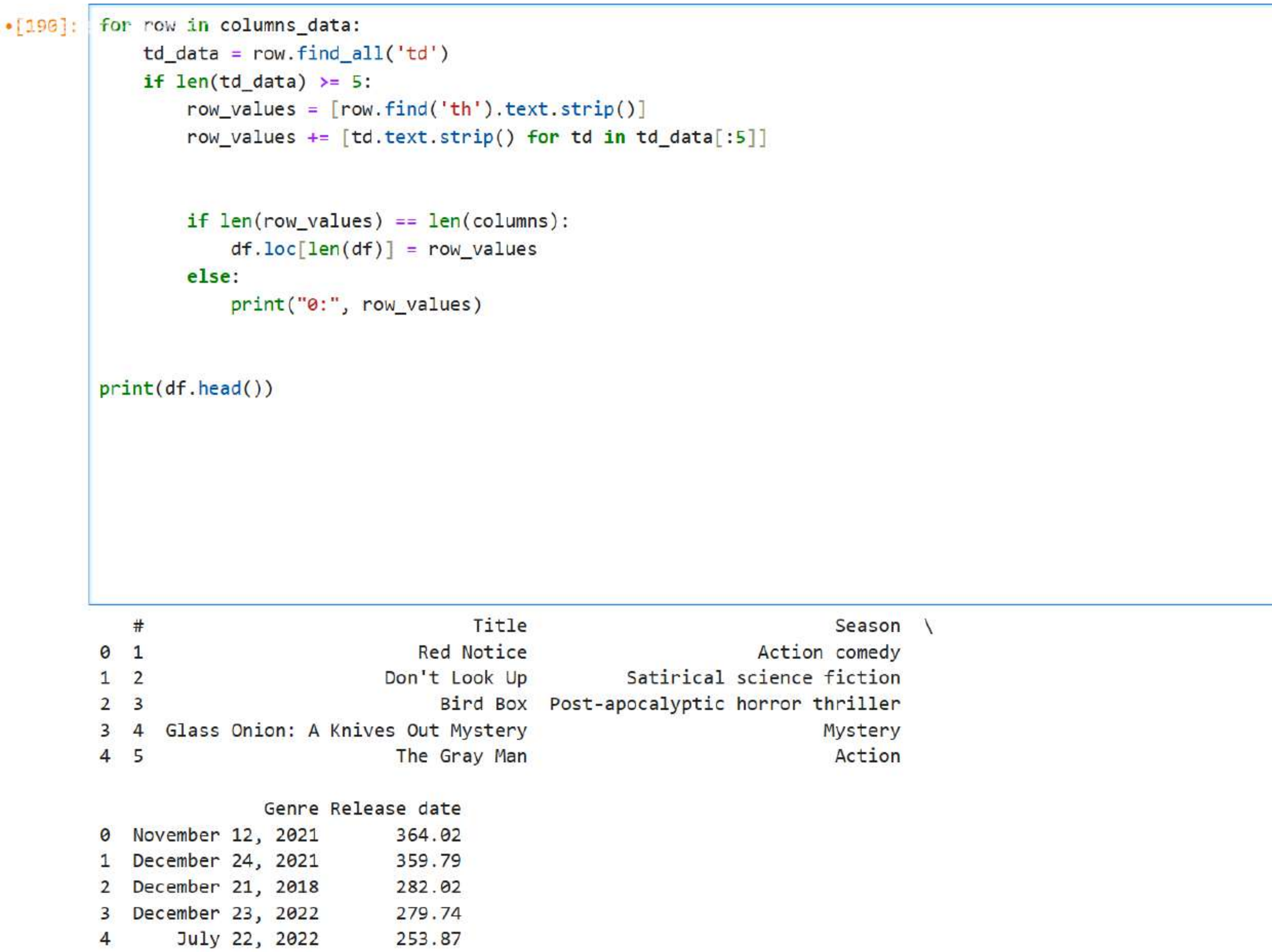


```
•[178]: data=[title.text.strip() for title in world_titles[:5]]
        print(data)

        ['#', 'Title', 'Genre', 'Release Date', 'Hours watched(millions)']

[101]:  import pandas as pd

[179]:  df=pd.DataFrame(columns=data)
        df
```

[179]:

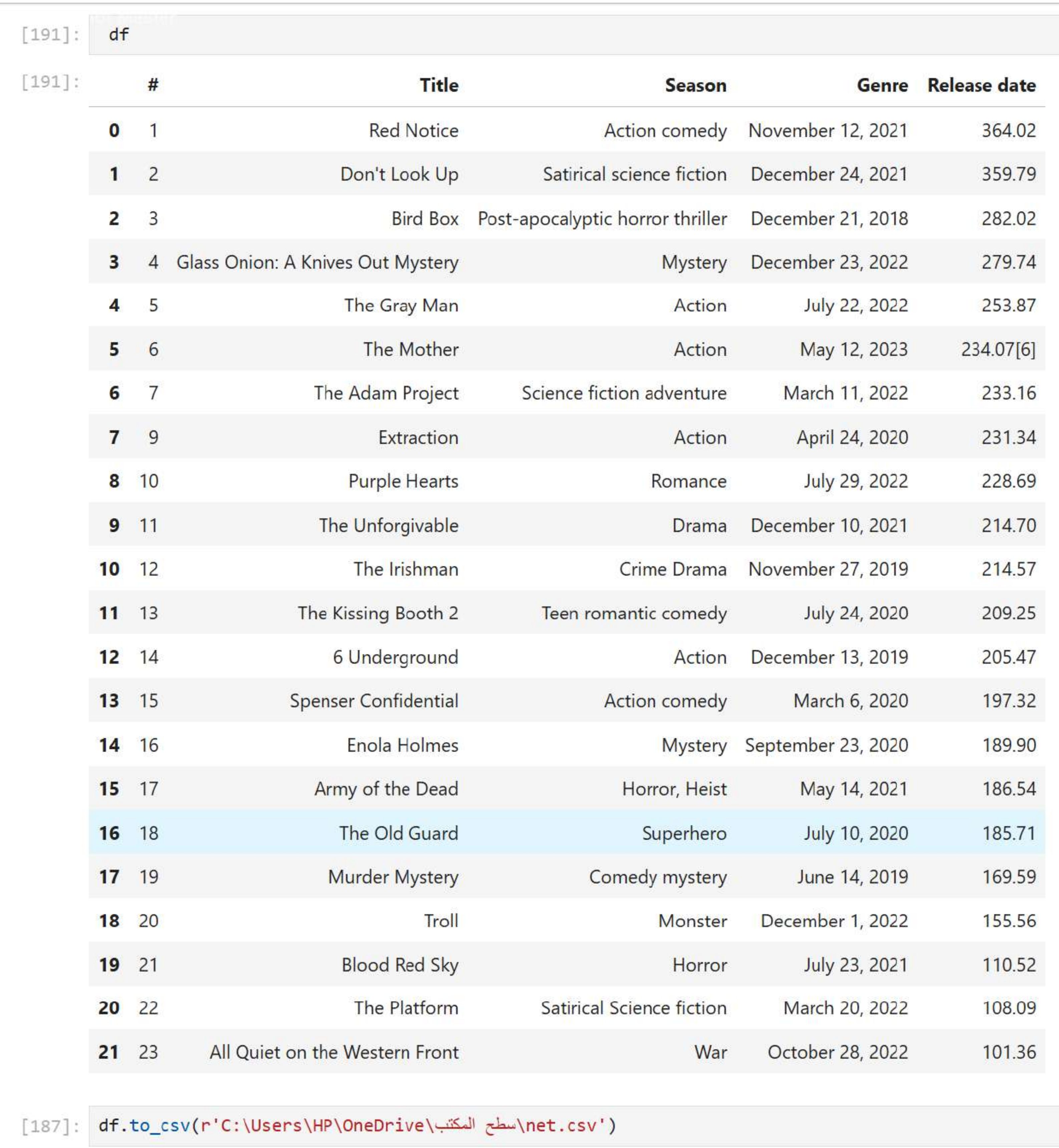
#	Title	Season	Genre	Release date

```
[180]:  columns_data=table.find('tbody').find_all('tr')
        print(columns_data)

[<tr>
<th>#
</th>
<th scope="col">Title
</th>
<th scope="col">Genre
</th>
<th scope="col">Release Date
</th>
<th scope="col">Hours watched<br/>(millions)
</th></tr>, <tr>
<th>1
</th>
<td scope="row"><i><a href="/wiki/Red_Notice_(film)" title="Red Notice (film)">Red Notice</a></i>
</td>
<td><a href="/wiki/Action_comedy" title="Action comedy">Action comedy</a>
</td>
<td>November 12, 2021
</td>
<td>364.02
</td></tr>, <tr>
<th>2
</th>
<td scope="row"><i><a href="/wiki/Don%27t_Look_Up" title="Don't Look Up">Don't Look Up</a></i>
</td>
<td><a href="/wiki/Satire_(film_and_television)" title="Satire (film and television)">Satirical</a> <a href="/wiki/Science_fiction_film" title="Science
fiction film">science fiction</a>
</td>
<td>December 24, 2021
```



```
•[190]: for row in columns_data:
            td_data = row.find_all('td')
            if len(td_data) >= 5:
                row_values = [row.find('th').text.strip()]
                row_values += [td.text.strip() for td in td_data[:5]]


                if len(row_values) == len(columns):
                    df.loc[len(df)] = row_values
                else:
                    print("0:", row_values)


        print(df.head())
```

```
   #                              Title                            Season  \
0  1                         Red Notice                     Action comedy   
1  2                      Don't Look Up         Satirical science fiction   
2  3                           Bird Box  Post-apocalyptic horror thriller   
3  4  Glass Onion: A Knives Out Mystery                           Mystery   
4  5                       The Gray Man                            Action   

               Genre Release date  
0  November 12, 2021       364.02  
1  December 24, 2021       359.79  
2  December 21, 2018       282.02  
3  December 23, 2022       279.74  
4      July 22, 2022       253.87  
```



```
[191]: df
```

[191]:

	#	Title	Season	Genre	Release date
0	1	Red Notice	Action comedy	November 12, 2021	364.02
1	2	Don't Look Up	Satirical science fiction	December 24, 2021	359.79
2	3	Bird Box	Post-apocalyptic horror thriller	December 21, 2018	282.02
3	4	Glass Onion: A Knives Out Mystery	Mystery	December 23, 2022	279.74
4	5	The Gray Man	Action	July 22, 2022	253.87
5	6	The Mother	Action	May 12, 2023	234.07[6]
6	7	The Adam Project	Science fiction adventure	March 11, 2022	233.16
7	9	Extraction	Action	April 24, 2020	231.34
8	10	Purple Hearts	Romance	July 29, 2022	228.69
9	11	The Unforgivable	Drama	December 10, 2021	214.70
10	12	The Irishman	Crime Drama	November 27, 2019	214.57
11	13	The Kissing Booth 2	Teen romantic comedy	July 24, 2020	209.25
12	14	6 Underground	Action	December 13, 2019	205.47
13	15	Spenser Confidential	Action comedy	March 6, 2020	197.32
14	16	Enola Holmes	Mystery	September 23, 2020	189.90
15	17	Army of the Dead	Horror, Heist	May 14, 2021	186.54
16	18	The Old Guard	Superhero	July 10, 2020	185.71
17	19	Murder Mystery	Comedy mystery	June 14, 2019	169.59
18	20	Troll	Monster	December 1, 2022	155.56
19	21	Blood Red Sky	Horror	July 23, 2021	110.52
20	22	The Platform	Satirical Science fiction	March 20, 2022	108.09
21	23	All Quiet on the Western Front	War	October 28, 2022	101.36

```
[187]: df.to_csv(r'C:\Users\HP\OneDrive\سطح المكتب\net.csv')
```