



MORE

 $\frac{1}{4}$

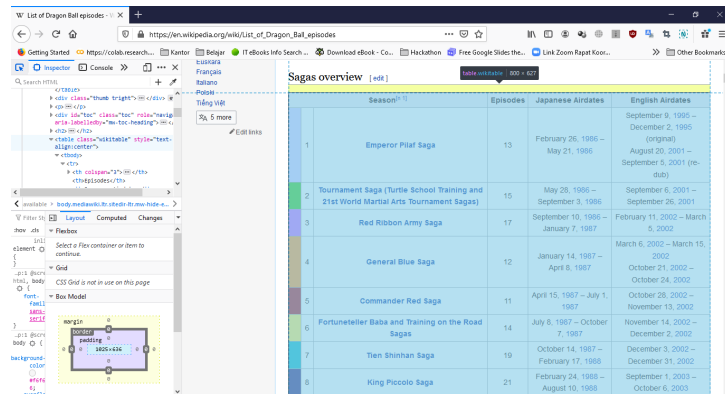
Setelah berhasil menginstall dua library diatas, lakukan import library untuk menguji apakah library tersebut sudah terinstall dengan baik dan selanjutnya bisa kita gunakan dalam kode python kita.

```
import pandas as pd
from bs4 import BeautifulSoup
from urllib.request import Request, urlopen
```

Langkah Kedua

Lakukan pengecekan melalui browser, pada elemen manakah tabel yang akan kita ekstrak tersebut berada. Apabila menggunakan Firefox/ Chrome tekan CTR+Shift+I atau dengan melakukan klik kanan pada mouse dan pilih *Inspect Element*.

Dari informasi yang ditampilkan pada jendela *Inspect Element* tersebut diketahui bahwa tabel yang akan kita ekstrak berada pada elemen table class="wikitable".



Langkah Ketiga

Setelah mengetahui di elemen mana informasi tersebut akan diambil, saatnya kita melakukan pemanggilan dan ekstraksi dari halaman tersebut. Panggil halaman menggunakan library request:

```
alamat = "https://en.wikipedia.org/wiki/List_of_Dragon_Ball_episodes"
req = Request(alamat, headers={'User-Agent': 'Mozilla/5.0'})
```

Perintah diatas akan melakukan pemanggilan pada alamat wikipedia. Argumen headers digunakan agar panggilan kita seolah-olah berasal dari sebuah web browser (yaitu 'Mozilla/5.0'). Argumen headers ini sifatnya opsional, beberapa website tidak mensyaratkan hal tersebut tetapi ada beberapa website lainnya yang melakukan pengecekan agar halaman website mereka hanya bisa dibuka menggunakan browser yang *legitimate*.

Langkah Keempat

Setelah halaman web berhasil dibuka, jalankan BeautifulSoup untuk melakukan ekstraksi tabel yang akan kita ambil.

```
html = urlopen(req).read()
data = BeautifulSoup(html, 'html.parser')
table = data.find_all('table', {'class': 'wikitable'})[0]
```

Variabel *table* menyimpan hasil pengambilan elemen tabel pertama yang merupakan tabel yang ingin kita ambil dari halaman website tersebut. Apabila kita coba tampilkan variabel tersebut penggunaan fungsi print maka akan menghasilkan tampilan seperti berikut:

```
<table class="wikitable" style="text-align:center">
<tbody><tr>
<th colspan="3">Season<sup class="reference" id="cite_ref-1"><a
href="#cite_note-1">[n 1]</a></sup>
</th>
<th>Episodes
</th>
<th>Japanese Airdates
</th>
<th>English Airdates
</th></tr>
<tr>
<td bgcolor="#CCF" width="3%">
</td>
<td width="3%">1
</td>
<td><b><a href="#Season_1:_Emperor_Pilaf_Saga_(1986)">Emperor Pilaf
Saga</a></b>
</td>
<td>13
</td>
<td>February 26, <a href="/wiki/1986_in_television" title="1986 in
television">1986</a> - May 21, <a href="/wiki/1986_in_television"
```

```
title="1986 in television">1986</a>
</td>
<td>September 9, <a href="/wiki/1995_in_television" title="1995 in
television">1995</a> - December 2, <a href="/wiki/1995_in_television"
title="1995 in television">1995</a> (original)<br/>August 20, <a
href="/wiki/2001_in_television" title="2001 in television">2001</a> -
September 5, <a href="/wiki/2001_in_television" title="2001 in
television">2001</a> (re-dub)
</td></tr>
<tr>
<td bgcolor="#2FCE1F">
</td>
<td>2
</td>
<td><b><a
href="#Season_2:_Tournament_Saga_(Turtle_School_Training_and_21st_World
_Martial_Arts_Tournament_Sagas)_(1986)">Tournament Saga (Turtle School
Training and 21st World Martial Arts Tournament Sagas)</a></b>
</td>
<td>15
</td>
<td>May 28, <a href="/wiki/1986_in_television" title="1986 in
television">1986</a> - September 3, <a href="/wiki/1986_in_television"
title="1986 in television">1986</a>
</td>
<td>September 6, <a href="/wiki/2001_in_television" title="2001 in
television">2001</a> - September 26, <a href="/wiki/2001_in_television"
title="2001 in television">2001</a>
</td></tr>
<tr>
...

```

Selanjutnya kita akan coba simpan masing-masing baris dari tabel tersebut kedalam sebuah list bernama rows.

```
rows = table.findAll('tr')
```


Lakukan iterasi pada variabel row untuk memecah masing-masing kolom menjadi sebuah list yang kemudian disimpan kedalam variabel hasil.

```
hasil = []
for row in rows:
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    hasil.append(cols)
```

Buat dataframe dari variabel list hasil dan tampilkan hasilnya.

```
df_hasil = pd.DataFrame(hasil)
df_hasil
```

Maka akan menghasilkan tampilan seperti berikut ini.



| | 0 | 1 | | 2 | 3 | | 4 | | 5 |
|---|------|---|----------------------|--------------------|--------------------------------------|----------------------------------|---|--|------|
| 0 | None | None | | None | None | | None | | None |
| 1 | 1 | | | Emperor Pilaf Saga | 13 | February 26, 1986 – May 21, 1986 | September 9, 1995 – December 2, 1995 (original... | | |
| 2 | 2 | Tournament Saga (Turtle School Training and 21... | | 15 | May 28, 1986 – September 3, 1986 | | September 6, 2001 – September 26, 2001 | | |
| 3 | 3 | | Red Ribbon Army Saga | 17 | September 10, 1986 – January 7, 1987 | | February 11, 2002 – March 5, 2002 | | |
| 4 | 4 | | General Blue Saga | 12 | January 14, 1987 – April 8, 1987 | March 6, 2002 – March 15, 2002 | October 21, 2002... | | |
| 5 | 5 | | Commander Red Saga | 11 | April 15, 1987 – July 1, 1987 | | October 28, 2002 – November 13, 2002 | | |
| 6 | 6 | Fortuneteller Baba and Training on the Road Sagas | | 14 | July 8, 1987 – October 7, 1987 | | November 14, 2002 – December 2, 2002 | | |
| 7 | 7 | | Tien Shinhan Saga | 19 | October 14, 1987 – February 17, 1988 | | December 3, 2002 – December 31, 2002 | | |
| 8 | 8 | | King Piccolo Saga | 21 | February 24, 1988 – August 10, 1988 | | September 1, 2003 – October 6, 2003 | | |
| 9 | 9 | Heavenly Training, Piccolo Jr. and The Wedding... | | 31 | August 17, 1988 – April 19, 1989 | | October 7, 2003 – December 1, 2003 | | |

Sudah lumayan menyerupai data yang diambil dari wikipedia bukan? Tetap ada beberapa hal yang harus diperbaiki dari dataframe yang telah dihasilkan, pertama adalah hapus kolom satu karena tidak menyimpan informasi apapun. Kemudian selanjutnya adalah memberi judul masing-masing kolom dari tabel kita. Terakhir adalah menghapus baris awal dari tabel kita karena terlihat disana tidak memiliki data/ none.

Jalankan perintah berikut untuk melakukan 3 hal tersebut.

```
df_hasil = df_hasil.drop([0], axis=1) # menghapus kolom pertama
df_hasil = df_hasil.drop([0], axis=0) # menghapus baris pertama
df_hasil.columns = ['No', 'Season', 'Episodes', 'Japanese Airdates', 'English Airdates',
'English Airdates'] # memberi judul kolom tabel
```

Maka dari perintah diatas akan menghasilkan dataframe seperti gambar dibawah ini.

| No | Season | Episodes | Japanese Airdates | English Airdates |
|----|--------|---|---|---|
| 1 | 1 | Emperor Pilaf Saga | 13 February 26, 1986 – May 21, 1986 | September 9, 1995 – December 2, 1995 (original... |
| 2 | 2 | Tournament Saga (Turtle School Training and 21... | 15 May 28, 1986 – September 3, 1986 | September 6, 2001 – September 26, 2001 |
| 3 | 3 | Red Ribbon Army Saga | 17 September 10, 1986 – January 7, 1987 | February 11, 2002 – March 5, 2002 |
| 4 | 4 | General Blue Saga | 12 January 14, 1987 – April 8, 1987 | March 6, 2002 – March 15, 2002 |
| 5 | 5 | Commander Red Saga | 11 April 15, 1987 – July 1, 1987 | October 28, 2002 – November 13, 2002 |
| 6 | 6 | Fortuneteller Baba and Training on the Road Sagas | 14 July 8, 1987 – October 7, 1987 | November 14, 2002 – December 2, 2002 |
| 7 | 7 | Tien Shinhan Saga | 19 October 14, 1987 – February 17, 1988 | December 3, 2002 – December 31, 2002 |
| 8 | 8 | King Piccolo Saga | 21 February 24, 1988 – August 10, 1988 | September 1, 2003 – October 6, 2003 |
| 9 | 9 | Heavenly Training, Piccolo Jr. and The Wedding... | 31 August 17, 1988 – April 19, 1989 | October 7, 2003 – December 1, 2003 |

Coba bandingkan hasilnya dengan halaman website wikipedia.

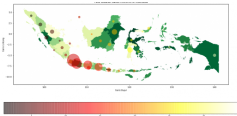
| | Season ^[n 1] | Episodes | Japanese Airdates | English Airdates | |
|--|-------------------------|---|-------------------|--------------------------------------|---|
| | 1 | Emperor Pilaf Saga | 13 | February 26, 1986 – May 21, 1986 | September 9, 1995 – December 2, 1995 (original) August 20, 2001 – September 5, 2001 (re-dub) |
| | 2 | Tournament Saga (Turtle School Training and 21st World Martial Arts Tournament Sagas) | 15 | May 26, 1986 – September 3, 1986 | September 6, 2001 – September 26, 2001 |
| | 3 | Red Ribbon Army Saga | 17 | September 10, 1986 – January 7, 1987 | February 11, 2002 – March 5, 2002 |
| | 4 | General Blue Saga | 12 | January 14, 1987 – April 8, 1987 | March 6, 2002 – March 15, 2002 October 21, 2002 – October 24, 2002 |
| | 5 | Commander Red Saga | 11 | April 15, 1987 – July 1, 1987 | October 28, 2002 – November 13, 2002 |
| | 6 | Fortuneteller Baba and Training on the Road Sagas | 14 | July 8, 1987 – October 7, 1987 | November 14, 2002 – December 2, 2002 |
| | 7 | Tien Shinhan Saga | 19 | October 14, 1987 – February 17, 1988 | December 3, 2002 – December 31, 2002 |
| | 8 | King Piccolo Saga | 21 | February 24, 1988 – August 10, 1988 | September 1, 2003 – October 6, 2003 |
| | 9 | Heavenly Training, Piccolo Jr. and The Wedding Dress in Flames Sagas | 31 | August 17, 1988 – April 19, 1989 | October 7, 2003 – December 1, 2003 |

Langkah terakhir adalah menyimpan hasil tersebut kedalam file excel.

```
df_hasil.to_excel('data_serial_dragonball.xlsx', index=False)
```

Tags: python

YOU MAY ALSO LIKE...



Melakukan Visualisasi Data Kasus COVID19 di Indonesia
MARCH 15, 2021



Membuat Wordcloud Dengan Python
MARCH 1, 2021



Webinar Sejarah Internet Indonesia
JULY 20, 2020

