

Utilizando el lenguaje de programación que desee (Python o R), ingrese al sitio

https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population y extraiga la tabla 'List of countries and dependencies by population'.

```

1 # Importamos las librerías que necesitamos
2 import requests
3 from bs4 import BeautifulSoup
4 import pandas as pd
5
6 # Ponemos la URL dentro de una variable
7 url = 'https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population'
8
9 # Definimos una función para extraer los datos
10
11 def extrer_tabla(url):
12
13     res = requests.get(url)
14     soup = BeautifulSoup(res.content, 'html.parser')
15     table = soup.find('table', {'class': 'wikitable'})
16
17     # Convertimos la tabla a un DataFrame de pandas
18
19     df = pd.read_html(str(table))[0]
20
21     return df
22
23 # Llamamos a la función y almacenamos el resultado en un DataFrame
24
25 tabla_res = extrer_tabla(url)
26
27 # Mostramos las primeras filas del DataFrame
28
29 print(tabla_res.head())
30

```

```

↳ Unnamed: 0      Location  Population % of world      Date \
0      NaN      World  8109452000      100%  29 May 2024
1      [b]      China  1409670000      NaN    31 Dec 2023
2      [b]      India  1400744000      NaN    1 Mar 2024
3      3  United States  335893238      NaN    1 Jan 2024
4      4      Indonesia  279118866      NaN    1 Jul 2023

```

```

Source (official or from the United Nations) Unnamed: 6
0      UN projection[3]      NaN
1      Official estimate[5]    [c]
2      Official projection[6]  [d]
3      Official estimate[7]    [e]
4      National annual projection[8]  NaN

```

```

1 #ahora voy a hacer un poco de data cleaning para que el data frame quede como yo quisiera.
2
3
4 tabla_res = tabla_res.drop(columns=['Source (official or from the United Nations)'])
5 tabla_res = tabla_res.drop(columns=['Unnamed: 6'])
6 tabla_res = tabla_res.drop(columns=['Unnamed: 0'])
7
8
9 #sacamos estas dos columnas que no vamos a necesitar para hacer lo que sigue de la entrega.
10
11 #printeamos el df para ver el resultado final
12
13 print(tabla_res)

```

```

↳
      Location  Population % of world      Date
0      World  8109452000      100%  29 May 2024
1      China  1409670000      NaN    31 Dec 2023
2      India  1400744000      NaN    1 Mar 2024
3      United States  335893238      NaN    1 Jan 2024
4      Indonesia  279118866      NaN    1 Jul 2023
..      ...      ...      ...      ...
236      Niue      1689      NaN    11 Nov 2022
237      Tokelau (NZ)      1647      NaN    1 Jan 2019
238      Vatican City      764      NaN    26 Jun 2023
239      Cocos (Keeling) Islands (Australia)      593      NaN    30 Jun 2020
240      Pitcairn Islands (UK)      47      NaN    1 Jul 2021


```

[241 rows x 4 columns]

```

1 #sacamos la primera linea que pertenece a "Mundo"
2
3 tabla_res = tabla_res.drop(index=0)
4
5 print(tabla_res)

```



	Location	Population	% of world	Date
1	China	1409670000	NaN	31 Dec 2023
2	India	1400744000	NaN	1 Mar 2024
3	United States	335893238	NaN	1 Jan 2024
4	Indonesia	279118866	NaN	1 Jul 2023
5	Pakistan	241499431	NaN	1 Mar 2023
..
236	Niue	1689	NaN	11 Nov 2022
237	Tokelau (NZ)	1647	NaN	1 Jan 2019
238	Vatican City	764	NaN	26 Jun 2023
239	Cocos (Keeling) Islands (Australia)	593	NaN	30 Jun 2020
240	Pitcairn Islands (UK)	47	NaN	1 Jul 2021

[240 rows x 4 columns]

1. (1 punto) ¿Cuál es el país con la mayor población?

```

1 #usamos la funcion idx, no vista en clase que nos da el indice de la fila que tiene mayor poblacion
2
3 max_population_idx = tabla_res['Population'].idxmax()
4
5 #usamos el resultado anterior para obtener el nombre del pais con mayor poblacion de acuerdo al indice brindado
6
7 country_with_max_population = tabla_res.loc[max_population_idx, 'Location']
8
9 print(country_with_max_population)
10


```

 China

```

1 #si solo se podrian usar recursos vistos en clase, podemos sortear la tabla en orden descendente
2
3 pais_mas_poblado = tabla_res.sort_values('Population', ascending = False).head()
4 pais_mas_poblado.iloc[0]

```



Location	China
Population	1409670000
% of world	NaN
Date	31 Dec 2023
Name: 1, dtype: object	

3. (1 punto) ¿Cuántos países tienen una población mayor a 100 millones?

```

1 tabla_res[tabla_res['Population'] > 100000000]['Location'].count()

```

 15

4. (1 punto) ¿Cuál es la población total de todos los países listados?

```

1 tabla_res['Population'].sum()

```

 7884748265

5. (1 punto) ¿Cuáles son los 5 países más poblados? Construya un DataFrame con las siguientes columnas: (País, Población).

```

1 df_mas_poblados = tabla_res.head(5)
2
3 df_mas_poblados = df_mas_poblados.drop(columns=['Date'])
4 df_mas_poblados = df_mas_poblados.drop(columns=['% of world'])
5
6 print(df_mas_poblados)


```



	Location	Population
1	China	1409670000
2	India	1400744000
3	United States	335893238
4	Indonesia	279118866
5	Pakistan	241499431


6. (1 punto) ¿Cuál es la población media de los países de África?

```
1 tabla_res.head()
```



	Location	Population	% of world	Date
1	China	1409670000	NaN	31 Dec 2023
2	India	1400744000	NaN	1 Mar 2024
3	United States	335893238	NaN	1 Jan 2024
4	Indonesia	279118866	NaN	1 Jul 2023
5	Pakistan	241499431	NaN	1 Mar 2023

```
1 #ponemos todos los paises africanos en una lista nueva
2
3 paises_africa = [
4     "Algeria", "Angola", "Benin", "Botswana", "Burkina Faso", "Burundi", "Cabo Verde",
5     "Cameroon", "Central African Republic", "Chad", "Comoros", "Congo", "Djibouti",
6     "Egypt", "Equatorial Guinea", "Eritrea", "Eswatini", "Ethiopia", "Gabon",
7     "Gambia", "Ghana", "Guinea", "Guinea-Bissau", "Ivory Coast", "Kenya", "Lesotho",
8     "Liberia", "Libya", "Madagascar", "Malawi", "Mali", "Mauritania", "Mauritius",
9     "Morocco", "Mozambique", "Namibia", "Niger", "Nigeria", "Rwanda", "Sao Tome and Principe",
10    "Senegal", "Seychelles", "Sierra Leone", "Somalia", "South Africa", "South Sudan",
11    "Sudan", "Tanzania", "Togo", "Tunisia", "Uganda", "Zambia", "Zimbabwe"
12 ]
13
14 # Definir la función para verificar si un país está en África
15 def es_pais_africano(pais):
16     if pais in paises_africa:
17         return True
18     else:
19         return False
20
21 #aplicamos la funcion al data frame para obtener solo los paises africanos
22 paises_africa_df = tabla_res[tabla_res['Location'].apply(es_pais_africano)]
23
24 #limpiamos el dataframe para tener solo la info que nos interesa
25 paises_africa_df = paises_africa_df.drop(columns=['% of world'])
26 paises_africa_df = paises_africa_df.drop(columns=['Date'])
27
28 print(paises_africa_df)
```



	Location	Population
6	Nigeria	223800000
13	Ethiopia	107334000
14	Egypt	105914499
23	South Africa	62027503
24	Tanzania	61741120
28	Kenya	51526000
32	Uganda	45562000
33	Algeria	45400000
35	Sudan	41984500
38	Morocco	37022000
42	Angola	34094077
45	Mozambique	32419747
48	Ghana	30832019
49	Ivory Coast	29389150
52	Cameroon	28088845
53	Madagascar	26923353
56	Niger	25369415
59	Burkina Faso	22752315
60	Mali	22395489
62	Malawi	21507723
65	Zambia	19610769
67	Senegal	18275743
68	Somalia	18143379
71	Chad	17414717
74	Zimbabwe	15178979

75	South Sudan	14746494
76	Guinea	13261638
77	Rwanda	13246394
78	Burundi	12837740
79	Benin	12606998
81	Tunisia	11850232
101	Sierra Leone	8494260
102	Togo	8095498
107	Libya	6931061
116	Central African Republic	5633412
125	Liberia	5248621
128	Mauritania	4475683
131	Eritrea	3748902
137	Namibia	3022401
144	Gambia	2417471
145	Botswana	2410338
146	Lesotho	2306000
147	Gabon	2233272
151	Guinea-Bissau	1781308
154	Equatorial Guinea	1558160
158	Mauritius	1261041
159	Eswatini	1223362
160	Djibouti	1001454
164	Comoros	758316
198	Seychelles	100447

```
1 paises_africa_df['Population'].mean()
```

 25719156.9

1 Start coding or [generate](#) with AI.