Handling country names using pycountry python package

pycountry Python package provides the ISO databases to match the standards for: Languages, Countries, Deleted countries, Subdivisions of countries, Currencies and Scripts. The ISO codes are as follow s:

- 3166 - Countries
- 639 - Languages
- 3166-3 - Deleted countries
- 3166-2 - Subdivisions of countries
- 4217 - Currencies
- 15924 - Scripts

The package includes a copy from Debian's pkg-isocodes and makes the data accessible through a Python API.

Translation files for the various strings are included as w ell.

Matching Country ISO codes

COUNTRIES (ISO 3166)

Countries are accessible through a database object that is already configured upon import of pycountry (https://pypi.org/project/pycountry/) and w orks as an iterable.

In the first query to *pycountry* let's import tha package and check how many countries are on the database (249 in total), secondly w ill check w hich is the first country on DB (Aruba), and last w e w ill iterate over all records, w ill visualize the parameters for each country.

As w e can easily inspect the alpha_2 parameter defines ISO 3166 Alpha-2 code as described in the International Standard (https://w w w.iban.com/country-codes), as w ell as alpha_3, and then the short name, the numeric code and the official name.

In [31]:
```
import pycountry
len(pycountry.countries)
```

Out[31]:
```
249
```

In [32]:
```
list(pycountry.countries)[0]
```

Out[32]:
```
Country(alpha_2='AW', alpha_3='ABW', name='Aruba', numeric='533')
```

In [33]:
```
for country in pycountry.countries:
    print(country)
```

```
Country(alpha_2='AW', alpha_3='ABW', name='Aruba', numeric='533')
Country(alpha_2='AF', alpha_3='AFG', name='Afghanistan', numeric='004',
official_name='Islamic Republic of Afghanistan')
Country(alpha_2='AO', alpha_3='AGO', name='Angola', numeric='024', offic
ial_name='Republic of Angola')
Country(alpha_2='AI', alpha_3='AIA', name='Anguilla', numeric='660')
Country(alpha_2='AX', alpha_3='ALA', name='Åland Islands', numeric='248
')
Country(alpha_2='AL', alpha_3='ALB', name='Albania', numeric='008', offi
cial_name='Republic of Albania')
Country(alpha_2='AD', alpha_3='AND', name='Andorra', numeric='020', offi
cial_name='Principality of Andorra')
Country(alpha_2='AE', alpha_3='ARE', name='United Arab Emirates', numeri
c='784')
Country(alpha_2='AR', alpha_3='ARG', name='Argentina', numeric='032', of
ficial_name='Argentine Republic')
Country(alpha_2='AM', alpha_3='ARM', name='Armenia', numeric='051', offi
cial_name='Republic of Armenia')
Country(alpha_2='AS', alpha_3='ASM', name='American Samoa', numeric='016
')
Country(alpha_2='AQ', alpha_3='ATA', name='Antarctica', numeric='010')
Country(alpha_2='TF', alpha_3='ATF', name='French Southern Territories',
numeric='260')
Country(alpha_2='AG', alpha_3='ATG', name='Antigua and Barbuda', numeric
='028')
Country(alpha_2='AU', alpha_3='AUS', name='Australia', numeric='036')
Country(alpha_2='AT', alpha_3='AUT', name='Austria', numeric='040', offi
cial_name='Republic of Austria')
Country(alpha_2='AZ', alpha_3='AZE', name='Azerbaijan', numeric='031', o
fficial_name='Republic of Azerbaijan')
Country(alpha_2='BI', alpha_3='BDI', name='Burundi', numeric='108', offi
cial_name='Republic of Burundi')
Country(alpha_2='BE', alpha_3='BEL', name='Belgium', numeric='056', offi
cial_name='Kingdom of Belgium')
Country(alpha_2='BJ', alpha_3='BEN', name='Benin', numeric='204', offici
al_name='Republic of Benin')
Country(alpha_2='BQ', alpha_3='BES', name='Bonaire, Sint Eustatius and S
aba', numeric='535', official_name='Bonaire, Sint Eustatius and Saba')
Country(alpha_2='BF', alpha_3='BFA', name='Burkina Faso', numeric='854')
Country(alpha_2='BD', alpha_3='BGD', name='Bangladesh', numeric='050', o
fficial_name="People's Republic of Bangladesh")
Country(alpha_2='BG', alpha_3='BGR', name='Bulgaria', numeric='100', off
icial_name='Republic of Bulgaria')
Country(alpha_2='BH', alpha_3='BHR', name='Bahrain', numeric='048', offi
cial_name='Kingdom of Bahrain')
Country(alpha_2='BS', alpha_3='BHS', name='Bahamas', numeric='044', offi
cial_name='Commonwealth of the Bahamas')
Country(alpha_2='BA', alpha_3='BIH', name='Bosnia and Herzegovina', nume
ric='070', official_name='Republic of Bosnia and Herzegovina')
Country(alpha_2='BL', alpha_3='BLM', name='Saint Barthélemy', numeric='6
52')
Country(alpha_2='BY', alpha_3='BLR', name='Belarus', numeric='112', offi
cial_name='Republic of Belarus')
Country(alpha_2='BZ', alpha_3='BLZ', name='Belize', numeric='084')
Country(alpha_2='BM', alpha_3='BMU', name='Bermuda', numeric='060')
Country(alpha_2='BO', alpha_3='BOL', common_name='Bolivia', name='Bolivi
a, Plurinational State of', numeric='068', official_name='Plurinational
State of Bolivia')
Country(alpha_2='BR', alpha_3='BRA', name='Brazil', numeric='076', offic
ial_name='Federative Republic of Brazil')
Country(alpha_2='BB', alpha_3='BRB', name='Barbados', numeric='052')
Country(alpha_2='BN', alpha_3='BRN', name='Brunei Darussalam', numeric='
096')
Country(alpha_2='BT', alpha_3='BTN', name='Bhutan', numeric='064', offic
```

Specific countries can be looked up by their various codes and provide the information included in the standard as attributes. I will do the example with my Country of Birth, looking for the alpha_2 and 3 codes, numeric code, name and official_name, last one is the name I am not very proud of... but anyway...

In [34]:
```
Venezuela = pycountry.countries.get(alpha_2='VE')
print(Venezuela)
print(Venezuela.alpha_2)
print(Venezuela.alpha_3)
print(Venezuela.numeric)
print(Venezuela.name)
print(Venezuela.official_name)
```

```
Country(alpha_2='VE', alpha_3='VEN', common_name='Venezuela', name='Vene
zuela, Bolivarian Republic of', numeric='862', official_name='Bolivarian
Republic of Venezuela')
VE
VEN
862
Venezuela, Bolivarian Republic of
Bolivarian Republic of Venezuela
```

The historic_countries database contains former countries that have been removed from the standard and are now included in ISO 3166-3, in addition to the existing ones. For example, let's look for the information of the former URSS, now days know as Russia.

In [35]:
```
URSS = pycountry.historic_countries.get(alpha_2='SU')
URSS
```

Out[35]:
```
Country(alpha_2='SU', alpha_3='SUN', alpha_4='SUHH', name='USSR, Union o
f Soviet Socialist Republics', numeric='810', withdrawal_date='1992-08-3
0')
```

In [36]:
```
Russia = pycountry.countries.get(alpha_2='RU')
Russia
```

Out[36]:
```
Country(alpha_2='RU', alpha_3='RUS', name='Russian Federation', numeric=
'643')
```

COUNTRY SUBDIVISIONS (ISO 3166-2)

The country subdivisions are a little more complex than the countries itself because they provide a nested and typed structure.

All subdivisons can be accessed directly, first let's check how many subdivisions exists for all the ISO Countries, and then which would be the first subdivision on that list, it corresponds to country_code **'AD'**, Andorra, and it is Canillo parish.

In [37]:
```
len(pycountry.subdivisions)
```

Out[37]:
```
4844
```

In [38]:
```
list(pycountry.subdivisions)[0]
```

Out[38]:
```
Subdivision(code='AD-02', country_code='AD', name='Canillo', parent_code
=None, type='Parish')
```

Subdivisions can be accessed using their unique code and provide at least their code, name and type, in this example w ill query one of the venezuelan subdivisions, w here specifically locates the capital city Caracas (https://en.w ikipedia.org/w iki/Caracas) in *Distrito Federal*.

In [39]:
```
venezuelan_state = pycountry.subdivisions.get(code='VE-A')
print(venezuelan_state)
```

```
Subdivision(code='VE-A', country_code='VE', name='Distrito Federal', par
ent_code=None, type='Federal District')
```

SCRIPTS (ISO 15924)

Scripts are available from a database similar to the countries, first w ill check how many scripts w orldw ide and their corresponding information.

In the second example w ill get the complete information of our mother tongue script, w hich is *Latin*

In [40]:
```
len(pycountry.scripts)
```

Out[40]:
```
182
```

In [41]:
```
for scripts in pycountry.scripts:
    print(scripts)
```

```
Script(alpha_4='Adlm', name='Adlam', numeric='166')
Script(alpha_4='Afak', name='Afaka', numeric='439')
Script(alpha_4='Aghb', name='Caucasian Albanian', numeric='239')
Script(alpha_4='Ahom', name='Ahom, Tai Ahom', numeric='338')
Script(alpha_4='Arab', name='Arabic', numeric='160')
Script(alpha_4='Aran', name='Arabic (Nastaliq variant)', numeric='161')
Script(alpha_4='Armi', name='Imperial Aramaic', numeric='124')
Script(alpha_4='Armn', name='Armenian', numeric='230')
Script(alpha_4='Avst', name='Avestan', numeric='134')
Script(alpha_4='Bali', name='Balinese', numeric='360')
Script(alpha_4='Bamu', name='Bamum', numeric='435')
Script(alpha_4='Bass', name='Bassa Vah', numeric='259')
Script(alpha_4='Batk', name='Batak', numeric='365')
Script(alpha_4='Beng', name='Bengali', numeric='325')
Script(alpha_4='Bhks', name='Bhaiksuki', numeric='334')
Script(alpha_4='Blis', name='Blissymbols', numeric='550')
Script(alpha_4='Bopo', name='Bopomofo', numeric='285')
Script(alpha_4='Brah', name='Brahmi', numeric='300')
Script(alpha_4='Brai', name='Braille', numeric='570')
Script(alpha_4='Bugi', name='Buginese', numeric='367')
Script(alpha_4='Buhd', name='Buhid', numeric='372')
Script(alpha_4='Cakm', name='Chakma', numeric='349')
Script(alpha_4='Cans', name='Unified Canadian Aboriginal Syllabics', num
eric='440')
Script(alpha_4='Cari', name='Carian', numeric='201')
Script(alpha_4='Cham', name='Cham', numeric='358')
Script(alpha_4='Cher', name='Cherokee', numeric='445')
Script(alpha_4='Cirt', name='Cirth', numeric='291')
Script(alpha_4='Copt', name='Coptic', numeric='204')
Script(alpha_4='Cprt', name='Cypriot', numeric='403')
Script(alpha_4='Cyrl', name='Cyrillic', numeric='220')
Script(alpha_4='Cyrs', name='Cyrillic (Old Church Slavonic variant)', nu
meric='221')
Script(alpha_4='Deva', name='Devanagari (Nagari)', numeric='315')
Script(alpha_4='Dsrt', name='Deseret (Mormon)', numeric='250')
Script(alpha_4='Dupl', name='Duployan shorthand, Duployan stenography',
numeric='755')
Script(alpha_4='Egyd', name='Egyptian demotic', numeric='070')
Script(alpha_4='Egyh', name='Egyptian hieratic', numeric='060')
Script(alpha_4='Egyp', name='Egyptian hieroglyphs', numeric='050')
Script(alpha_4='Elba', name='Elbasan', numeric='226')
Script(alpha_4='Ethi', name='Ethiopic (Ge‘ez)', numeric='430')
Script(alpha_4='Geok', name='Khutsuri (Asomtavruli and Nuskhuri)', numer
ic='241')
Script(alpha_4='Geor', name='Georgian (Mkhedruli)', numeric='240')
Script(alpha_4='Glag', name='Glagolitic', numeric='225')
Script(alpha_4='Goth', name='Gothic', numeric='206')
Script(alpha_4='Gran', name='Grantha', numeric='343')
Script(alpha_4='Grek', name='Greek', numeric='200')
Script(alpha_4='Gujr', name='Gujarati', numeric='320')
Script(alpha_4='Guru', name='Gurmukhi', numeric='310')
Script(alpha_4='Hanb', name='Han with Bopomofo (alias for Han + Bopomof
o)', numeric='503')
Script(alpha_4='Hang', name='Hangul (Hangŭl, Hangeul)', numeric='286')
Script(alpha_4='Hani', name='Han (Hanzi, Kanji, Hanja)', numeric='500')
Script(alpha_4='Hano', name='Hanunoo (Hanunóo)', numeric='371')
Script(alpha_4='Hans', name='Han (Simplified variant)', numeric='501')
Script(alpha_4='Hant', name='Han (Traditional variant)', numeric='502')
Script(alpha_4='Hatr', name='Hatran', numeric='127')
Script(alpha_4='Hebr', name='Hebrew', numeric='125')
Script(alpha_4='Hira', name='Hiragana', numeric='410')
Script(alpha_4='Hluw', name='Anatolian Hieroglyphs (Luwian Hieroglyphs,
Hittite Hieroglyphs)', numeric='080')
Script(alpha_4='Hmng', name='Pahawh Hmong', numeric='450')
```

In [42]:
```
latin = pycountry.scripts.get(name='Latin')
print(latin)
```

```
Script(alpha_4='Latn', name='Latin', numeric='215')
```

LANGUAGES (ISO 639)

As well as the scripts, we can get the languages from *ISO 639* database. In a similar way as in the Scripts case, we will check how many languagues, and will iterate over the database, in order to save space in the Notebook, I will not print the result of the up to 7000 lnguages.

In [43]:
```
len(pycountry.languages)
```

Out[43]: 7847

We could look for Spanish language by setting the **alpha_2** parameter as *es*, as well as English looking for the **alpha_2** *en*

In [44]:
```
spanish = pycountry.languages.get(alpha_2='es')
print(spanish)
```

```
Language(alpha_2='es', alpha_3='spa', name='Spanish', scope='I', type='L
')
```

In [45]:
```
english = pycountry.languages.get(alpha_2='en')
print(english)
```

```
Language(alpha_2='en', alpha_3='eng', name='English', scope='I', type='L
')
```

Scraping list of countries

Now we will use the package called _pycountry*convert* in order to get the continent names for every country in a list of countries that we will extract from wikipedia.

First we will import the *requests* package that will connect to the wikipedia URL, then we will use the *BeautifulSoup* package to manipulate that data.

In [46]:
```
import requests
website_url = requests.get('https://en.wikipedia.org/wiki/List_of_countrie
s_by_GDP_(nominal)').text
```

In [47]:
```
from bs4 import BeautifulSoup
soup = BeautifulSoup(website_url,'lxml')
```

In [48]:
```
My_table = soup.find('table',{'class':'wikitable sortable'})
```

In [49]:
```
links = My_table.findAll('a')
links
```

Out[49]:
```
[<a href="/wiki/Gross_world_product" title="Gross world product">World</
a>,
 <a href="#cite_note-IMF_Groups-20">[19]</a>,
 <a href="/wiki/United_States" title="United States">United States</a>,
 <a href="/wiki/European_Union" title="European Union">European Union</a
>,
 <a href="#cite_note-24">[23]</a>,
 <a href="#cite_note-EU_note-26">[n 1]</a>,
 <a href="/wiki/China" title="China">China</a>,
 <a href="#cite_note-China-THM-27">[n 2]</a>,
 <a href="/wiki/Japan" title="Japan">Japan</a>,
 <a href="/wiki/Germany" title="Germany">Germany</a>,
 <a href="/wiki/India" title="India">India</a>,
 <a href="/wiki/United_Kingdom" title="United Kingdom">United Kingdom</a
>,
 <a href="/wiki/France" title="France">France</a>,
 <a href="/wiki/Italy" title="Italy">Italy</a>,
 <a href="/wiki/Brazil" title="Brazil">Brazil</a>,
 <a href="/wiki/Canada" title="Canada">Canada</a>,
 <a href="/wiki/Russia" title="Russia">Russia</a>,
 <a href="#cite_note-Russia-28">[n 3]</a>,
 <a href="/wiki/South_Korea" title="South Korea">Korea, South</a>,
 <a href="/wiki/Spain" title="Spain">Spain</a>,
 <a href="/wiki/Australia" title="Australia">Australia</a>,
 <a href="/wiki/Mexico" title="Mexico">Mexico</a>,
 <a href="/wiki/Indonesia" title="Indonesia">Indonesia</a>,
 <a href="/wiki/Netherlands" title="Netherlands">Netherlands</a>,
 <a href="/wiki/Saudi_Arabia" title="Saudi Arabia">Saudi Arabia</a>,
 <a href="/wiki/Turkey" title="Turkey">Turkey</a>,
 <a href="/wiki/Switzerland" title="Switzerland">Switzerland</a>,
 <a href="/wiki/Taiwan" title="Taiwan">Taiwan</a>,
 <a href="/wiki/Poland" title="Poland">Poland</a>,
 <a href="/wiki/Thailand" title="Thailand">Thailand</a>,
 <a href="/wiki/Sweden" title="Sweden">Sweden</a>,
 <a href="/wiki/Belgium" title="Belgium">Belgium</a>,
 <a href="/wiki/Iran" title="Iran">Iran</a>,
 <a href="/wiki/Austria" title="Austria">Austria</a>,
 <a href="/wiki/Nigeria" title="Nigeria">Nigeria</a>,
 <a href="/wiki/Argentina" title="Argentina">Argentina</a>,
 <a href="/wiki/Norway" title="Norway">Norway</a>,
 <a href="/wiki/United_Arab_Emirates" title="United Arab Emirates">Unite
d Arab Emirates</a>,
 <a href="/wiki/Israel" title="Israel">Israel</a>,
 <a href="/wiki/Republic_of_Ireland" title="Republic of Ireland">Ireland
</a>,
 <a href="/wiki/Hong_Kong" title="Hong Kong">Hong Kong</a>,
 <a href="/wiki/Malaysia" title="Malaysia">Malaysia</a>,
 <a href="/wiki/Singapore" title="Singapore">Singapore</a>,
 <a href="/wiki/South_Africa" title="South Africa">South Africa</a>,
 <a href="/wiki/Philippines" title="Philippines">Philippines</a>,
 <a href="/wiki/Denmark" title="Denmark">Denmark</a>,
 <a href="/wiki/Colombia" title="Colombia">Colombia</a>,
 <a href="/wiki/Bangladesh" title="Bangladesh">Bangladesh</a>,
 <a href="/wiki/Egypt" title="Egypt">Egypt</a>,
 <a href="/wiki/Chile" title="Chile">Chile</a>,
 <a href="/wiki/Pakistan" title="Pakistan">Pakistan</a>,
 <a href="/wiki/Finland" title="Finland">Finland</a>,
 <a href="/wiki/Vietnam" title="Vietnam">Vietnam</a>,
 <a href="/wiki/Czech_Republic" title="Czech Republic">Czech Republic</a
>,
 <a href="/wiki/Romania" title="Romania">Romania</a>,
 <a href="/wiki/Portugal" title="Portugal">Portugal</a>,
 <a href="/wiki/Peru" title="Peru">Peru</a>,
 <a href="/wiki/Iraq" title="Iraq">Iraq</a>,
```

We w ill store the result of the request operation in the list *Countries* and then w ill iterate each title on that list.

In [50]:
```
Countries = []
for link in links:
    Countries.append(link.get('title'))
print(Countries)
```

```
['Gross world product', None, 'United States', 'European Union', None, N
one, 'China', None, 'Japan', 'Germany', 'India', 'United Kingdom', 'Fran
ce', 'Italy', 'Brazil', 'Canada', 'Russia', None, 'South Korea', 'Spain
', 'Australia', 'Mexico', 'Indonesia', 'Netherlands', 'Saudi Arabia', 'T
urkey', 'Switzerland', 'Taiwan', 'Poland', 'Thailand', 'Sweden', 'Belgiu
m', 'Iran', 'Austria', 'Nigeria', 'Argentina', 'Norway', 'United Arab Em
irates', 'Israel', 'Republic of Ireland', 'Hong Kong', 'Malaysia', 'Sing
apore', 'South Africa', 'Philippines', 'Denmark', 'Colombia', 'Banglades
h', 'Egypt', 'Chile', 'Pakistan', 'Finland', 'Vietnam', 'Czech Republic
', 'Romania', 'Portugal', 'Peru', 'Iraq', 'Greece', 'New Zealand', 'Qata
r', 'Algeria', 'Hungary', 'Kazakhstan', 'Ukraine', 'Kuwait', 'Morocco',
'Ecuador', 'Slovakia', 'Puerto Rico', 'Kenya', 'Angola', 'Ethiopia', 'Do
minican Republic', 'Sri Lanka', 'Guatemala', 'Oman', 'Venezuela', 'Luxem
bourg', 'Panama', 'Ghana', 'Bulgaria', 'Myanmar', 'Tanzania', 'Belarus',
'Costa Rica', 'Croatia', 'Uzbekistan', 'Syria', None, 'Uruguay', 'Lebano
n', 'Macau', 'Slovenia', 'Lithuania', 'Serbia', 'Democratic Republic of
the Congo', 'Azerbaijan', 'Turkmenistan', 'Ivory Coast', 'Jordan', 'Boli
via', 'Paraguay', 'Tunisia', 'Cameroon', 'Bahrain', 'Latvia', 'Libya', '
Estonia', 'Sudan', 'Uganda', 'Yemen', 'Nepal', 'El Salvador', 'Cambodia
', 'Honduras', 'Cyprus', 'Zambia', 'Senegal', 'Iceland', 'Papua New Guin
ea', 'Trinidad and Tobago', 'Bosnia and Herzegovina', 'Laos', 'Afghanist
an', 'Botswana', 'Mali', 'Gabon', 'Georgia (country)', 'Jamaica', 'Alban
ia', 'Mozambique', 'Malta', 'Burkina Faso', 'Mauritius', 'Benin', 'Namib
ia', 'Mongolia', 'Armenia', 'Guinea', 'Zimbabwe', 'North Macedonia', 'Th
e Bahamas', 'Madagascar', 'Nicaragua', 'Brunei', 'Equatorial Guinea', 'M
oldova', 'Republic of the Congo', 'Chad', 'Rwanda', 'Niger', 'Haiti', 'K
yrgyzstan', 'Tajikistan', 'Kosovo', 'Malawi', 'Maldives', 'Togo', 'Mauri
tania', 'Montenegro', 'Fiji', 'Barbados', 'Somalia', 'Eswatini', 'Sierra
Leone', 'Guyana', 'Suriname', 'South Sudan', 'Burundi', 'Liberia', 'Djib
outi', 'East Timor', 'Aruba', 'Bhutan', 'Lesotho', 'Central African Repu
blic', 'Eritrea', 'Belize', 'Saint Lucia', 'The Gambia', 'Antigua and Ba
rbuda', 'Seychelles', 'San Marino', 'Solomon Islands', 'Grenada', 'Comor
os', 'Saint Kitts and Nevis', 'Vanuatu', 'Samoa', 'Saint Vincent and the
Grenadines', 'Dominica', 'Tonga', 'São Tomé and Príncipe', 'Federated St
ates of Micronesia', 'Palau', 'Marshall Islands', 'Kiribati', 'Tuvalu']
```

Then w e w ill convert our list of countries in a DataFrame. Once it is a DataFrame, taking a look on that list w e note that there are names that shouldn't be there like *Gross world product* and *European Union* so w e w ill remove them from that list and the *None* as w ell.

In [51]:
```
import pandas as pd
df = pd.DataFrame()
df['Country'] = Countries
df
```

Out[51]:

	Country
0	Gross world product
1	None
2	United States
3	European Union
4	None
...	...
194	Federated States of Micronesia
195	Palau
196	Marshall Islands
197	Kiribati
198	Tuvalu

199 rows × 1 columns

In [52]:
```
World_Countries = df.loc[(df['Country'] !='Gross world product') & (df['Country'] !='European Union')]
World_Countries
```

Out[52]:

	Country
1	None
2	United States
4	None
5	None
6	China
...	...
194	Federated States of Micronesia
195	Palau
196	Marshall Islands
197	Kiribati
198	Tuvalu

197 rows × 1 columns

In [53]:
```
World_Countries.Country.unique()
```

Out[53]:
```
array([None, 'United States', 'China', 'Japan', 'Germany', 'India',
       'United Kingdom', 'France', 'Italy', 'Brazil', 'Canada', 'Russia
',
       'South Korea', 'Spain', 'Australia', 'Mexico', 'Indonesia',
       'Netherlands', 'Saudi Arabia', 'Turkey', 'Switzerland', 'Taiwan',
       'Poland', 'Thailand', 'Sweden', 'Belgium', 'Iran', 'Austria',
       'Nigeria', 'Argentina', 'Norway', 'United Arab Emirates', 'Israel
',
       'Republic of Ireland', 'Hong Kong', 'Malaysia', 'Singapore',
       'South Africa', 'Philippines', 'Denmark', 'Colombia', 'Bangladesh
',
       'Egypt', 'Chile', 'Pakistan', 'Finland', 'Vietnam',
       'Czech Republic', 'Romania', 'Portugal', 'Peru', 'Iraq', 'Greece
',
       'New Zealand', 'Qatar', 'Algeria', 'Hungary', 'Kazakhstan',
       'Ukraine', 'Kuwait', 'Morocco', 'Ecuador', 'Slovakia',
       'Puerto Rico', 'Kenya', 'Angola', 'Ethiopia', 'Dominican Republic
',
       'Sri Lanka', 'Guatemala', 'Oman', 'Venezuela', 'Luxembourg',
       'Panama', 'Ghana', 'Bulgaria', 'Myanmar', 'Tanzania', 'Belarus',
       'Costa Rica', 'Croatia', 'Uzbekistan', 'Syria', 'Uruguay',
       'Lebanon', 'Macau', 'Slovenia', 'Lithuania', 'Serbia',
       'Democratic Republic of the Congo', 'Azerbaijan', 'Turkmenistan',
       'Ivory Coast', 'Jordan', 'Bolivia', 'Paraguay', 'Tunisia',
       'Cameroon', 'Bahrain', 'Latvia', 'Libya', 'Estonia', 'Sudan',
       'Uganda', 'Yemen', 'Nepal', 'El Salvador', 'Cambodia', 'Honduras
',
       'Cyprus', 'Zambia', 'Senegal', 'Iceland', 'Papua New Guinea',
       'Trinidad and Tobago', 'Bosnia and Herzegovina', 'Laos',
       'Afghanistan', 'Botswana', 'Mali', 'Gabon', 'Georgia (country)',
       'Jamaica', 'Albania', 'Mozambique', 'Malta', 'Burkina Faso',
       'Mauritius', 'Benin', 'Namibia', 'Mongolia', 'Armenia', 'Guinea',
       'Zimbabwe', 'North Macedonia', 'The Bahamas', 'Madagascar',
       'Nicaragua', 'Brunei', 'Equatorial Guinea', 'Moldova',
       'Republic of the Congo', 'Chad', 'Rwanda', 'Niger', 'Haiti',
       'Kyrgyzstan', 'Tajikistan', 'Kosovo', 'Malawi', 'Maldives', 'Togo
',
       'Mauritania', 'Montenegro', 'Fiji', 'Barbados', 'Somalia',
       'Eswatini', 'Sierra Leone', 'Guyana', 'Suriname', 'South Sudan',
       'Burundi', 'Liberia', 'Djibouti', 'East Timor', 'Aruba', 'Bhutan
',
       'Lesotho', 'Central African Republic', 'Eritrea', 'Belize',
       'Saint Lucia', 'The Gambia', 'Antigua and Barbuda', 'Seychelles',
       'San Marino', 'Solomon Islands', 'Grenada', 'Comoros',
       'Saint Kitts and Nevis', 'Vanuatu', 'Samoa',
       'Saint Vincent and the Grenadines', 'Dominica', 'Tonga',
       'São Tomé and Príncipe', 'Federated States of Micronesia', 'Palau
',
       'Marshall Islands', 'Kiribati', 'Tuvalu'], dtype=object)
```

In [54]:
```
World_Countries.shape
```

Out[54]:
```
(197, 1)
```

Calling pycountry_convert package

Now is time to call the pycountry_convert (https://pypi.org/project/pycountry-convert/) package, and let's demonstrate how to convert country names to country and continent codes.

In [55]:
```
import pycountry_convert as pc

country_code = pc.country_name_to_country_alpha2("China", cn_name_format="
default")
print(country_code)
continent_name = pc.country_alpha2_to_continent_code(country_code)
print(continent_name)
```

```
CN
AS
```

In [56]:
```
#Function to to map country names
def country_to_continent(country_name):
    country_alpha2 = pc.country_name_to_country_alpha2(country_name)
    country_continent_code = pc.country_alpha2_to_continent_code(country_a
lpha2)
    country_continent_name = pc.convert_continent_code_to_continent_name(c
ountry_continent_code)
    return country_continent_name
```

In [57]:
```
country_name = 'India'
print(country_to_continent(country_name))
```

```
Asia
```

Now let's create a function to map continent names to **country_alpha2_to_continent_code** and **country_name_to_country_alpha2** elements from pycountry, this will return as well a list of the continents where those countries are located.

After that we'll merge both list (countrie and continents) into one, and then convert it to a pandas DataFrame.

In [58]:
```
from pycountry_convert import country_alpha2_to_continent_code, country_name_to_country_alpha2

continents = {
    'NA': 'North America',
    'SA': 'South America',
    'EU': 'European Union',
    'AS': 'Asia',
    'OC': 'Australia',
    'AF': 'Africa',
}

countries = ['United States', 'China', 'Japan', 'Germany', 'India',
       'United Kingdom', 'France', 'Italy', 'Brazil', 'Canada', 'Russia',
       'South Korea', 'Spain', 'Australia', 'Mexico', 'Indonesia',
       'Netherlands', 'Saudi Arabia', 'Turkey', 'Switzerland', 'Taiwan',
       'Poland', 'Thailand', 'Sweden', 'Belgium', 'Iran', 'Austria',
       'Nigeria', 'Argentina', 'Norway', 'United Arab Emirates', 'Israel',
       'Ireland', 'Hong Kong', 'Malaysia', 'Singapore',
       'South Africa', 'Philippines', 'Denmark', 'Colombia', 'Bangladesh',
       'Egypt', 'Chile', 'Pakistan', 'Finland', 'Vietnam',
       'Czech Republic', 'Romania', 'Portugal', 'Peru', 'Iraq', 'Greece',
       'New Zealand', 'Qatar', 'Algeria', 'Hungary', 'Kazakhstan',
       'Ukraine', 'Kuwait', 'Morocco', 'Ecuador', 'Slovakia',
       'Puerto Rico', 'Kenya', 'Angola', 'Ethiopia', 'Dominican Republic',
       'Sri Lanka', 'Guatemala', 'Oman', 'Venezuela', 'Luxembourg',
       'Panama', 'Ghana', 'Bulgaria', 'Myanmar', 'Tanzania', 'Belarus',
       'Costa Rica', 'Croatia', 'Uzbekistan', 'Syria', 'Uruguay',
       'Lebanon', 'Macau', 'Slovenia', 'Lithuania', 'Serbia',
       'Azerbaijan', 'Turkmenistan','Ivory Coast', 'Jordan', 'Bolivia',
       'Paraguay', 'Tunisia', 'Cameroon', 'Bahrain', 'Latvia', 'Libya', 'Estonia', 'Sudan',
       'Uganda', 'Yemen', 'Nepal', 'El Salvador', 'Cambodia', 'Honduras',
       'Cyprus', 'Zambia', 'Senegal', 'Iceland', 'Papua New Guinea',
       'Trinidad and Tobago', 'Bosnia and Herzegovina', 'Laos',
       'Afghanistan', 'Botswana', 'Mali', 'Gabon', 'Georgia',
       'Jamaica', 'Albania', 'Mozambique', 'Malta', 'Burkina Faso',
       'Mauritius', 'Benin', 'Namibia', 'Mongolia', 'Armenia', 'Guinea',
       'Zimbabwe', 'North Macedonia', 'Bahamas', 'Madagascar',
       'Nicaragua', 'Brunei', 'Equatorial Guinea', 'Moldova',
       'Chad', 'Rwanda', 'Niger', 'Haiti',
       'Kyrgyzstan', 'Tajikistan', 'Malawi', 'Maldives', 'Togo',
       'Mauritania', 'Montenegro', 'Fiji', 'Barbados', 'Somalia',
       'Eswatini', 'Sierra Leone', 'Guyana', 'Suriname', 'South Sudan',
       'Burundi', 'Liberia', 'Djibouti', 'Aruba', 'Bhutan',
       'Lesotho', 'Central African Republic', 'Eritrea', 'Belize',
       'Saint Lucia', 'Gambia', 'Antigua and Barbuda', 'Seychelles',
       'San Marino', 'Solomon Islands', 'Grenada', 'Comoros',
       'Saint Kitts and Nevis', 'Vanuatu', 'Samoa',
       'Saint Vincent and the Grenadines', 'Dominica', 'Tonga',
       'Federated States of Micronesia', 'Palau']

continents = [continents[country_alpha2_to_continent_code(country_name_to_country_alpha2(country))] for country in countries]
```

In [59]:
```
countries_list = pd.DataFrame(
    {'Country': countries,
     'Continents': continents,
    })

countries_list
```

Out[59]:

	Country	Continents
0	United States	North America
1	China	Asia
2	Japan	Asia
3	Germany	European Union
4	India	Asia
...	...	...
178	Saint Vincent and the Grenadines	North America
179	Dominica	North America
180	Tonga	Australia
181	Federated States of Micronesia	Australia
182	Palau	Australia

183 rows × 2 columns

We can continue adding more features to any list of countries extracted from the web of a csv table imported to pandas, using the pycountry and pycountry-convert packages.

In [ ]:
```
```