Scraping List of Blockchain Companies from Wikipedia for Further Analysis

Web Scraping with BeautifulSoup and Data Presentation in Pandas

Overview:

In this project, I scrape blockchain data from Wikipedia using `BeautifulSoup`, clean the extracted data, and store it in a structured `pandas` DataFrame. Finally, I saved the data as a CSV file for further exploratory analysis.

[1]:
```python
#Import the necessary libraries
from bs4 import BeautifulSoup
import requests
```

[2]:
```python
url = 'https://en.wikipedia.org/wiki/List_of_blockchains' #add the webpage to be scraped
page = requests.get(url)
soup = BeautifulSoup(page.text, 'html')
```

[5]:
```python
print(soup)
```

```
<li class="user-links-collapsible-item mw-list-item user-links-collapsible-item" id="pt-createaccount-2"><a class="" data-mw="interface" href="/w/index.php?title=Special:CreateAccount&amp;returnto=List+of+blockchains" title="You are encouraged to create an account and log in; however, it is not mandatory"><span>Create account</span></a>
</li>
<li class="user-links-collapsible-item mw-list-item user-links-collapsible-item" id="pt-login-2"><a accesskey="o" class="" data-mw="interface" href="/w/index.php?title=Special:UserLogin&amp;returnto=List+of+blockchains" title="You're encouraged to log in; however, it's not mandatory. [o]"><span>Log in</span></a>
</li>
</ul>
</div>
</div>
</div>
<div class="vector-dropdown vector-user-menu vector-button-flush-right vector-user-menu-logged-out" id="vector-user-links-dropdown" title="Log in and more options">
<input aria-haspopup="true" aria-label="Personal tools" class="vector-dropdown-checkbox" data-event-name="ui.dropdown-vector-user-links-dropdown" id="vector-user-links-dropdown-checkbox" role="button" type="checkbox"/>
<label aria-hidden="true" class="vector-dropdown-label cdx-button cdx-button--fake-button cdx-button--fake-button--enabled cdx-button--weight-quiet cdx-button--icon-only" for="vector-user-links-dropdown-checkbox" id="vector-user-links-dropdown-label"><span class="vector-icon mw-ui-icon-ellipsis mw-ui-ic
```

[7]:
```python
#Find the table in html of the webpage
soup.find('table')
```

[7]:
```
<table class="plainlinks metadata ambox mbox-small-left ambox-notice" role="presentation" style="width: auto;"><tbody><tr><td class="mbox-image"><span typeof="mw:File"><span><img alt="" class="mw-file-element" data-file-height="620" data-file-width="620" decoding="async" height="20" src="//upload.wikimedia.org/wikipedia/en/thumb/1/1d/Information_icon4.svg/20px-Information_icon4.svg.png" srcset="//upload.wikimedia.org/wikipedia/en/thumb/1/1d/Information_icon4.svg/30px-Information_icon4.svg.png 1.5x, //upload.wikimedia.org/wikipedia/en/thumb/1/1d/Information_icon4.svg/40px-Information_icon4.svg.png 2x" width="20"/></span></span></td><td class="mbox-text" style="width: auto;"><div class="mbox-text-span">This list is <a href="/wiki/Wikipedia:WikiProject_Lists#Incomplete_lists" title="Wikipedia:WikiProject Lists">incomplete</a>; you can help by <a class="external text" href="https://en.wikipedia.org/w/index.php?title=List_of_blockchains&amp;action=edit">adding missing items</a>.  <span class="date-container"><i>(<span class="date">July 2022</span>)</i></span></div></td></tr></tbody></table>
```

[9]:
```python
soup.find_all('table')[1]
```

[9]:
```
<table class="wikitable sortable">
<caption>
</caption>
<tbody><tr>
<th>Name
</th>
<th>Date created
</th>
<th>Created by
</th>
<th>Native cryptocurrency
</th>
<th>Consensus algorithm
</th>
<th>Programmable?
</th>
<th>Private?<sup class="reference" id="cite_ref-ISO_1-0"><a href="#cite_note-ISO-1"><span class="cite-bracket">[</span>Note 1<span class="cite-bracket">]</span></a></sup>
```

[11]:
```python
soup.find('table', class_ = 'wikitable sortable')

#table class="wikitable sortable jquery-tablesorter">
#<caption>
```

[11]:
```
<table class="wikitable sortable">
<caption>
</caption>
<tbody><tr>
<th>Name
</th>
<th>Date created
</th>
<th>Created by
</th>
<th>Native cryptocurrency
</th>
<th>Consensus algorithm
</th>
<th>Programmable?
</th>
<th>Private?<sup class="reference" id="cite_ref-ISO_1-0"><a href="#cite_note-ISO-1"><span class="cite-bracket">[</span>Note 1<span class="cite-bracket">]</span></a></sup>
```

[13]:
```python
table = soup.find_all('table')[1]
```

[15]:
```python
print(table)
```

```
<table class="wikitable sortable">
<caption>
</caption>
<tbody><tr>
<th>Name
</th>
<th>Date created
</th>
<th>Created by
</th>
<th>Native cryptocurrency
</th>
<th>Consensus algorithm
</th>
<th>Programmable?
</th>
<th>Private?<sup class="reference" id="cite_ref-ISO_1-0"><a href="#cite_note-ISO-1"><span class="cite-bracket">[</span>Note 1<span class="cite-bracket">]</span></a></sup>
```

[17]:
```python
soup.find_all('th') #Find the elements that have the "th" label
```

[17]:
```
[<th>Name
 </th>,
 <th>Date created
 </th>,
 <th>Created by
 </th>,
 <th>Native cryptocurrency
 </th>,
 <th>Consensus algorithm
 </th>,
 <th>Programmable?
 </th>,
 <th>Private?<sup class="reference" id="cite_ref-ISO_1-0"><a href="#cite_note-ISO-1"><span class="cite-bracket">[</span>Note 1<span class="cite-bracket">]</span></a></sup>
 </th>,
 <th>Permissioned?<sup class="reference" id="cite_ref-ISO_1-1"><a href="#cite_note-ISO-1"><span class="cite-bracket">[</span>Note 1<span class="cite-bracket">]</span></a></sup>
 </th>,
 <th>Finality
 </th>,
 <th>Ledger state
 </th>,
 <th>Notes
 </th>,
 <th>Refs.
 </th>]
```

[19]:
```python
block_titles = table.find_all ('th')
```

[21]:
```python
print (block_titles)
```

```
[<th>Name
</th>, <th>Date created
</th>, <th>Created by
</th>, <th>Native cryptocurrency
</th>, <th>Consensus algorithm
</th>, <th>Programmable?
</th>, <th>Private?<sup class="reference" id="cite_ref-ISO_1-0"><a href="#cite_note-ISO-1"><span class="cite-bracket">[</span>Note 1<span class="cite-bracket">]</span></a></sup>
</th>, <th>Permissioned?<sup class="reference" id="cite_ref-ISO_1-1"><a href="#cite_note-ISO-1"><span class="cite-bracket">[</span>Note 1<span class="cite-bracket">]</span></a></sup>
</th>, <th>Finality
</th>, <th>Ledger state
</th>, <th>Notes
</th>, <th>Refs.
</th>]
```

[23]:
```python
block_table_titles = [title.text.strip() for title in block_titles]
print(block_table_titles)
```

```
['Name', 'Date created', 'Created by', 'Native cryptocurrency', 'Consensus algorithm', 'Programmable?', 'Private?[Note 1]', 'Permissioned?[Note 1]', 'Finality', 'Ledger state', 'Notes', 'Refs.']
```

[25]:
```python
#I want to replace some column titles in the list
block_table_titles = [block_table_title.replace('Refs.', ' ') for block_table_title in block_table_titles]
block_table_titles = [block_table_title.replace('Private?[Note 1]', 'Private?') for block_table_title in block_table_titles]
block_table_titles = [block_table_title.replace('Permissioned?[Note 1]', 'Permissioned?') for block_table_title in block_table_titles]
#I also need to remove the links
print(block_table_titles)
```

```
['Name', 'Date created', 'Created by', 'Native cryptocurrency', 'Consensus algorithm', 'Programmable?', 'Private?', 'Permissioned?', 'Finality', 'Ledger state', 'Notes', ' ']
```

[27]:
```python
#next, we want to put the column titles into our pandas dataframe
import pandas as pd
```

[29]:
```python
df = pd.DataFrame(columns = block_table_titles)
df
```

[29]:

Name	Date created	Created by	Native cryptocurrency	Consensus algorithm	Programmable?	Private?	Permissioned?	Finality	Ledger state	Notes

[31]:
```python
column_data = table.find_all('tr')
```

[33]:
```python
for row in column_data [1:]: #removed first empty bracket
   row_data = row.find_all('td')
   # Extract text from each table cell (td) instead of using raw Tag objects
   new_row_data = [td.get_text(strip=True) for td in column_data] #this extracts the text from each cell and shows it as text not a Beatysoup tag
   print(new_row_data)

   individual_row_data = [data.get_text(strip=True) for data in row_data] #removing this shows error "cannot set rows with mismatched column", adding it ba

   print(individual_row_data)
  # what you get are all individual lists, so we need to find a way to get them into our table
  # we cannot stuff everything into our data frame at once. So let's put it in one at a time. We will use Location in pandas

   length = len(df) #looking at length of current df. which is empty right now
   df.loc[length] = individual_row_data   #we are looping through individual row data and appending each row of info into next location
```

```
-2 on Ethereum[7][8]', 'StarkNet?StarkWareSTRKZK-rollupYesLayer-2 on Ethereum[9]', 'Polygon zkEVM?Polygon TechnologyETHZK-rollupYeszkEVM-compatibleLayer-2 on Ethereum', 'Ethereum ClassicJul 20, 2016ETCPoWYesNoNoSplit from Ethereum due toThe DAOhack earlier that month[1]', 'Bitcoin CashAug 1, 2017BCHPoW[10]split from Bitcoin', 'CardanoSep 27, 2017Charles Hoskinsonand Jeremy WoodADADPoSYesNoNoProbabilisticUTXO[1][11][12]', 'TRONJun 24, 2018TRONDPoSYesNo[1][13]', 'TezosJun 30, 2018Arthur and  Kathleen BreitmanXTZPoSYesNo[1][14]', 'Bitcoin SVNov 2018BSVPoWYes (scripts)Nosplit from Bitcoin Cash, itself split from Bitcoin[1]', 'Lightning Network[relevant?]2018n/a?Layer-2 on Bitcoin', 'XinFinJune 1, 2019XinFin Fintech, XDC FoundationXDCXDPOSYesNoNoimmediateXDC Network is an layer 1 EVM compatible, environmental friendly, near zero transaction cost with high speed settlement blockchain platform.', 'AlgorandJun 10, 2019Silvio Micaliand othersALGOPoSYesNoNoImmediateUses averifiable random functionto choose random validators for consensus[15]', '$PCSep 16, 2019$PCPoSYes (scripts)NoNo$PC future well-known blockchain[16]', 'SolanaMarch 16, 2020Anatoly Yakovenko and Raj GokalSOLPoSwith Proof of History (PoH)YesNoNoImmediateAccount-balance[17]', 'OasisJune 18, 2020Oasis Labs,Oasis Protocol FoundationROSEPoSYes (in ParaTimes)NoNoImmediateAccount-balanceParaTimes can useTEEforconfidential computing[18][19]', 'PolkadotMay 26, 2020Parity TechnologiesDOTStarted withPoAthen moved toPoSYes (in parachains)NoNoHybridAccount-balance[20][21]', 'AvalancheSeptember 10, 2020Emin Gün Sirer, Maofan "Ted" Yin andKevin SekniqiAVAXPoSYes (in C-Chain)NoNoImmediateUTXO[22]', 'MobileCoinDec 6, 2020MobileCoin Inc. (founded byJosh Goldbard and Shane Glynn)MOB', 'Internet Computer?DFINITY Foundation(founded byDominic WIlliams)ICPComputation is very cheap; can host websites', 'DESOJan 18, 2021Nader al-Naji (akadiamondhands) and othersDESO (formerlyBTCLT, CLOUT)social media; flagship appBitClout; name acquired in Sep 2021[23]', 'Terra Classic?Do Kwonand othersLUNC (formerlyLUNA), USTFormerly Terra until May 2022; ecosystem collapsed in May 2022 (UST depegged to near-zero and LUNA also went to near-zero)', 'Terra 2.0May 28, 2022LUNANew blockchain created following the collapse of Terra.[24]', 'StellarApr 6, 2016XLMBFT??Yes?[4]', 'EOS.IOJul 1, 2017?EOSDPoSYesNo??[4]', 'LBRY?LBC', 'RippleJune 2012Ripple LabsXRPBFTNoNoNoImmediateAccount-balanceBlockchain is known as XRP Ledger. Smart contract capabilities are being added.[25][26][27]', 'Stacks?STX', 'VertcoinJan 8, 2014VTC', 'Hedera Hashgraph?HBARYesNoYesAccount-balanceUses adirected acyclic graphinstead of a chainper se', 'ZcashOct 28, 2016[4]ZECPoWYesuseszero-knowledge proofsfor privacy', 'MoneroApr 18, 2014XMRPoWNoYes[1][4]', 'Bitcoin GoldBTGPoWYes (scripts)No[1]', 'DogecoinDOGEPoWNoNo[1]', 'Hyperledger Fabric?Linux FoundationN/
```

[35]:
```python
df
```

[35]:

	Name	Date created	Created by	Native cryptocurrency	Consensus algorithm	Programmable?	Private?	Permissioned?	Finality	Ledger state	Note
0	Bitcoin	January 3, 2009	Satoshi Nakamoto	BTC	PoWwith Nakamoto Consensus	Yes (scripts)	No	No	Probabilistic	UTXO	First and most well known blockchain o all; B..
1	Litecoin	Oct 8, 2011	Charlie Lee	LTC	PoW	Yes (scripts)	Yes				
2	Peercoin	Aug 19, 2012		PPC	PoW	Yes (scripts)	No				
3	Primecoin	Jul 7, 2013	Sunny King	XPM	PoW						Work is finding longCunningham chainsofprime n..
			Pavu Harris								Initially created fo

[255]:
```python
#we have now scraped successfully!! we can export to a CSV
df.to_csv (r'C:\Users\chiso\Downloads\MYSQL\block_scraped.csv', index= False)
```

[ ]:
```python
#The end
```

Summary:

- We successfully scraped blockchain data from Wikipedia.
- Cleaned the extracted table by handling inconsistent row lengths.
- Stored the data in a structured `pandas` DataFrame.
- Exported the cleaned data as a CSV file for further use.

Next Steps:

- Perform **exploratory data analysis (EDA)** on the dataset.
- Visualize blockchain data trends using **matplotlib** or **seaborn**.
- Automate data scraping to track changes over time.