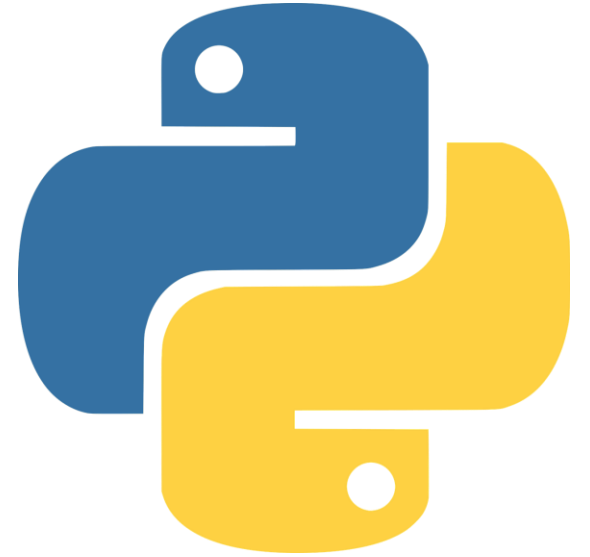


# Web Scraping with Python Using BeautifulSoup

I'll be going over how we can navigate through the HTML tree.

Specifically, how to do this so we can find prices of cryptocurrencies.

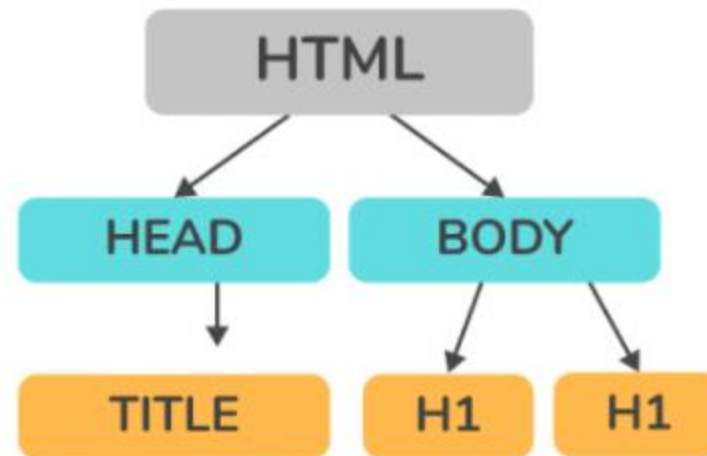


BeautifulSoup

# Tree Structure

## Tree Structure

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <h1></h1>  
    <h1></h1>  
  </body>  
</html>
```



# Tree Structure

coinmarketcap.com

Apps СоцСети Фотошоп QA java C# & SQL Marvel Python NodeJS Elastic MONGO DB Projects for Itay - G... Скачать файл | iLov... BigData FCM Vue

2 GAMINGDOGE GAMINGDOGE -29.84% 2 Chainbing CBG -439.08% 2 Tremendous Coin TMDS \$5.31  
3 Retromoon RETRO -27.73% 3 MetaCat METACAT -215.69% 3 GLOW GLOW \$0.00000001504

Watchlist Portfolio Cryptocurrencies Categories DeFi NFT Metaverse Polkadot BSC Solana Avalanche Show rows 100 Filters

#	Name	Price	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply	Last
1	Bitcoin BTC Buy	\$50,302.32	-2.12%	-1.53%	\$951,503,051,547	\$22,313,362,284 443,201 BTC	18,899,293 BTC	
2	Ethereum ETH Buy	\$4,161.14	-2.81%	-1.05%	\$493,723,424,245	\$14,773,486,881 3,552,226 ETH	118,713,840 ETH	
3	Binance Coin BNB Buy	\$571.26	-1.29%	-2.15%	\$95,317,511,341	\$1,314,135,240 2,299,675 BNB	166,801,148 BNB	
4	Tether USDT Buy	\$1.00	-0.01%	-0.03%	\$76,194,332,857	\$46,903,496,393 46,880,086,746 USDT	76,156,304,084 USDT	
5	Solana SOL Buy	\$174.59	-1.78%	-11.78%	\$53,636,557,565	\$1,201,936,759 6,894,039 SOL	307,647,226 SOL	
6	Cardano ADA	\$1.36	-0.72%	-2.53%	\$45,179,226,842	\$1,497,200,201 1107,807,756 ADA	33,428,994,929 ADA	

Elements Console Sources Network Performance

```
<div class="h7vnx2-1 bFzXgL" style="overflow-x: scroll;">
  <table class="h7vnx2-2 czTsgW cmc-table">
    <thead>
      <tr>
        <th>#</th>
        <th>Name</th>
        <th>Price</th>
        <th>24h %</th>
        <th>7d %</th>
        <th>Market Cap</th>
        <th>Volume(24h)</th>
        <th>Circulating Supply</th>
        <th>Last</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Bitcoin BTC Buy</td>
        <td>$50,302.32</td>
        <td>-2.12%</td>
        <td>-1.53%</td>
        <td>$951,503,051,547</td>
        <td>$22,313,362,284<br>443,201 BTC</td>
        <td>18,899,293 BTC</td>
        <td><img alt="Bitcoin price chart" /></td>
      </tr>
      <tr>
        <td>2</td>
        <td>Ethereum ETH Buy</td>
        <td>$4,161.14</td>
        <td>-2.81%</td>
        <td>-1.05%</td>
        <td>$493,723,424,245</td>
        <td>$14,773,486,881<br>3,552,226 ETH</td>
        <td>118,713,840 ETH</td>
        <td><img alt="Ethereum price chart" /></td>
      </tr>
      <tr>
        <td>3</td>
        <td>Binance Coin BNB Buy</td>
        <td>$571.26</td>
        <td>-1.29%</td>
        <td>-2.15%</td>
        <td>$95,317,511,341</td>
        <td>$1,314,135,240<br>2,299,675 BNB</td>
        <td>166,801,148 BNB</td>
        <td><img alt="Binance Coin price chart" /></td>
      </tr>
      <tr>
        <td>4</td>
        <td>Tether USDT Buy</td>
        <td>$1.00</td>
        <td>-0.01%</td>
        <td>-0.03%</td>
        <td>$76,194,332,857</td>
        <td>$46,903,496,393<br>46,880,086,746 USDT</td>
        <td>76,156,304,084 USDT</td>
        <td><img alt="Tether price chart" /></td>
      </tr>
      <tr>
        <td>5</td>
        <td>Solana SOL Buy</td>
        <td>$174.59</td>
        <td>-1.78%</td>
        <td>-11.78%</td>
        <td>$53,636,557,565</td>
        <td>$1,201,936,759<br>6,894,039 SOL</td>
        <td>307,647,226 SOL</td>
        <td><img alt="Solana price chart" /></td>
      </tr>
      <tr>
        <td>6</td>
        <td>Cardano ADA</td>
        <td>$1.36</td>
        <td>-0.72%</td>
        <td>-2.53%</td>
        <td>$45,179,226,842</td>
        <td>$1,497,200,201<br>1107,807,756 ADA</td>
        <td>33,428,994,929 ADA</td>
        <td><img alt="Cardano price chart" /></td>
      </tr>
    </tbody>
  </table>
</div>
```

html body.DAY div#\_next

tbody 1 of 1 Cancel

Console What's New X Issues Developer Resources

# Tree Structure

```
from bs4 import BeautifulSoup
import requests

url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

print(trs)
```

web\_scraping3.py

# Tree Siblings

- **previous\_sibling** is used to find the previous element of the given element
- **next\_sibling** is used to find the next element of the given element
- **previous\_siblings** is used to find all previous element of the given element
- **next\_siblings** is used to find all next element of the given element

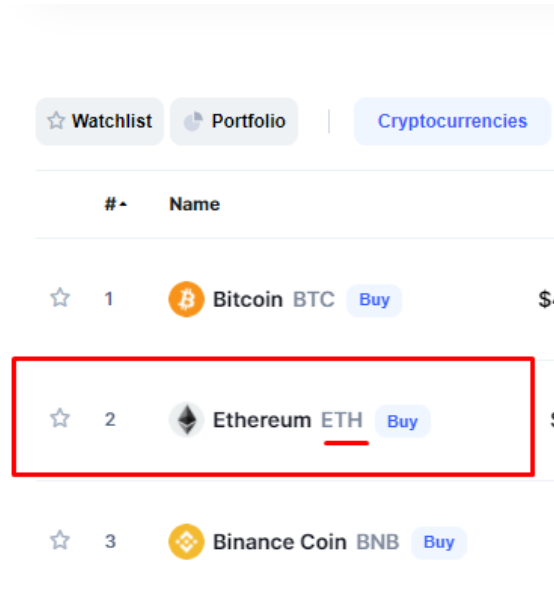
web\_scraping3.py

```
from bs4 import BeautifulSoup
import requests

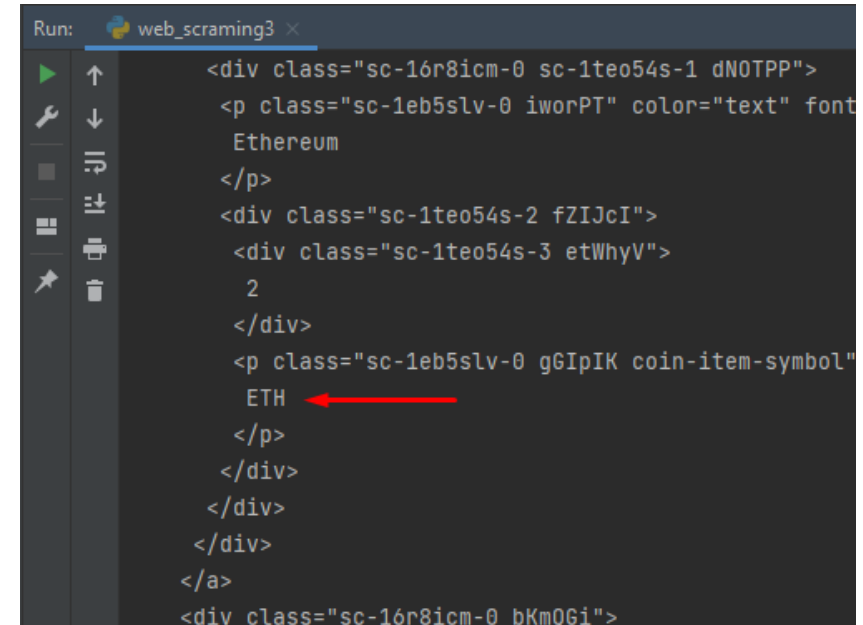
url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

print(trs[0].next_sibling.prettify())
```



☆ Watchlist		Portfolio		Cryptocurrencies	
#	Name				
☆ 1	Bitcoin BTC	Buy		\$4	
☆ 2	Ethereum ETH	Buy		\$	
☆ 3	Binance Coin BNB	Buy			



```
Run: web_scraming3 x
<div class="sc-16r8icm-0 sc-1teo54s-1 dNOTPP">
  <p class="sc-1eb5slv-0 iworPT" color="text" font
    Ethereum
  </p>
  <div class="sc-1teo54s-2 fZIJcI">
    <div class="sc-1teo54s-3 etWhyV">
      2
    </div>
    <p class="sc-1eb5slv-0 g6IpIK coin-item-symbol"
      ETH
    </p>
  </div>
</div>
</a>
<div class="sc-16r8icm-0 bKmqG6i">
```

# Tree Siblings

- **previous\_sibling** is used to find the previous element of the given element
- **next\_sibling** is used to find the next element of the given element
- **previous\_siblings** is used to find all previous element of the given element
- **next\_siblings** is used to find all next element of the given element

## web\_scraping3.py




```
from bs4 import BeautifulSoup
import requests

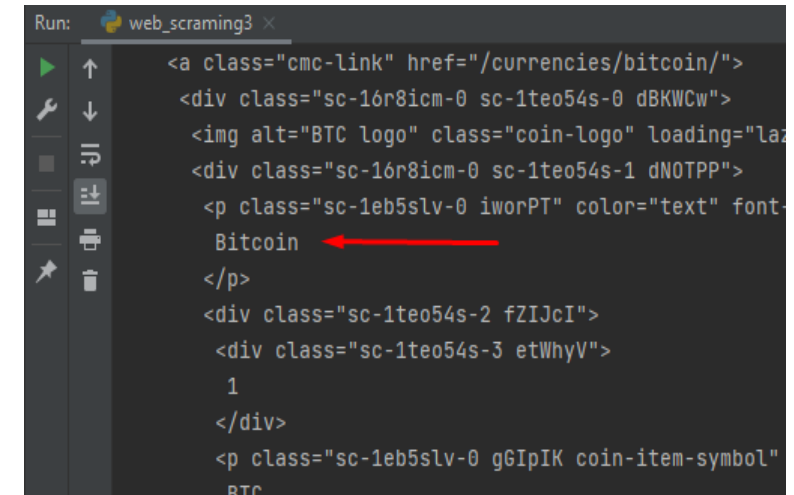
url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

print(trs[1].previous_sibling.prettify())
```

☆ Watchlist | Portfolio | Cryptocurrencies Categories

#	Name	Price
☆ 1	 Bitcoin BTC <a href="#">Buy</a>	\$49,220.49
☆ 2	 Ethereum ETH <a href="#">Buy</a>	\$4,028.50
☆ 3	 Binance Coin BNB <a href="#">Buy</a>	\$560.83



```
Run: web_scraming3 x
<a class="cmc-link" href="/currencies/bitcoin/">
  <div class="sc-16r8icm-0 sc-1teo54s-0 dBKWCw">
    <img alt="BTC logo" class="coin-logo" loading="lazy">
    <div class="sc-16r8icm-0 sc-1teo54s-1 dNOTPP">
      <p class="sc-1eb5slv-0 iworPT" color="text" font-size="14px">
        Bitcoin
      </p>
      <div class="sc-1teo54s-2 fZIJcI">
        <div class="sc-1teo54s-3 etWhyV">
          1
        </div>
      <p class="sc-1eb5slv-0 g6IpIK coin-item-symbol">
        BTC
```

# Tree Parents and Descendants

Continuing the “family tree” analogy, every tag and every string has a parent: the tag that contains it.

You can access an element’s parent with the `.parent` attribute.

`web_scraping3.py`

```
from bs4 import BeautifulSoup
import requests

url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

print(trs[1].parent.name)
```

# Tree Parents and Descendants

The `.contents` and `.children` attributes only consider a tag's direct children

The `.descendants` attribute lets you iterate over all of a tag's children, recursively: its direct children, the children of its direct children, and so on:

`web_scraping3.py`

```
from bs4 import BeautifulSoup
import requests

url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

print(list(trs[0].descendants))
```



# Getting Crypto Prices

Looking for all of the table data that's inside of that table

[web\\_scraping3.py](#)

```
from bs4 import BeautifulSoup
import requests

url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

prices = {}

for tr in trs:
    for td in tr.contents:
        print(td)
        print()
```

# Getting Crypto Prices

Print only name of cryptocurrency that's inside table

`web_scraping3.py`

```
from bs4 import BeautifulSoup
import requests

url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

prices = {}

for tr in trs[:10]:
    name, price = tr.contents[2:4]
    print(name.p.string)
    print()
```

# Getting Crypto Prices

Print name of cryptocurrency with its price

web\_scraping3.py

```
from bs4 import BeautifulSoup
import requests

url = "https://coinmarketcap.com/"
result = requests.get(url).text
doc = BeautifulSoup(result, "html.parser")

tbody = doc.tbody
trs = tbody.contents

prices = {}

for tr in trs[:10]:
    name, price = tr.contents[2:4]
    fixed_name = name.p.string
    fixed_price = price.a.string

    prices[fixed_name] = fixed_price

print(prices)
```