

I give you a URL and you
give me back LEGO bricks

or how to use an LLM to write your first scraper

⚠️⚠️⚠️ This is NOT bricklink.com, NOT affiliated with LEGO® or Bricklink™. Test/demo page. Intended for HTML+CSS workshop ⚠️⚠️⚠️

Keyword Search: Condition: Min Qty: Min Price: Max Price: Available Items Only

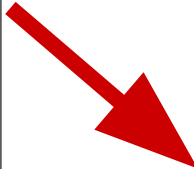
[More Options](#) ▼

9,185 Items Found Showing prices in Euro (EUR) [\(more info\)](#) Sort:

- Overview (9,185)
- Part (7,843)
- Gear (503)
- Set (142)
- Original Box (142)
- Minifigure (122)
- Book (80)
- Instruction (67)
- Catalog (1)
- Custom Items (285)

25 per page ...

Part (25 out of 7,843)	Condition	Qty	Sellers	Price
 Brick 1 x 1 Brick : 3005	New	15,730,337	6,663	EUR 0.0001+
	Used	4,921,039	87,937	EUR 0.0013+
 Brick 1 x 2 Brick : 3004	New	16,800,431	6,779	EUR 0.0001+
	Used	7,076,722	101,667	EUR 0.0009+
 Brick 1 x 3 Brick : 3622	New	3,867,766	5,893	EUR 0.0001+
	Used	1,311,312	47,376	EUR 0.001+
 Brick 1 x 4 Brick : 3010	New	5,709,796	6,332	EUR 0.0001+



ID, URL, Name

- 3005, /v2/catalog/catalogitem.page?P=3005&name=Brick%201%20x%201&category=%5BBrick%5D,Brick 1 x 1
- 3004, /v2/catalog/catalogitem.page?P=3004&name=Brick%201%20x%202&category=%5BBrick%5D,Brick 1 x 2
- 3622, /v2/catalog/catalogitem.page?P=3622&name=Brick%201%20x%203&category=%5BBrick%5D,Brick 1 x 3
- 3010, /v2/catalog/catalogitem.page?P=3010&name=Brick%201%20x%204&category=%5BBrick%5D,Brick 1 x 4
- 3009, /v2/catalog/catalogitem.page?P=3009&name=Brick%201%20x%206&category=%5BBrick%5D,Brick 1 x 6
- 3008, /v2/catalog/catalogitem.page?P=3008&name=Brick%201%20x%208&category=%5BBrick%5D,Brick 1 x 8
- 3003, /v2/catalog/catalogitem.page?P=3003&name=Brick%202%20x%202&category=%5BBrick%5D,Brick 2 x 2
- 3002, /v2/catalog/catalogitem.page?P=3002&name=Brick%202%20x%203&category=%5BBrick%5D,Brick 2 x 3
- 3001, /v2/catalog/catalogitem.page?P=3001&name=Brick%202%20x%204&category=%5BBrick%5D,Brick 2 x 4
- 2456, /v2/catalog/catalogitem.page?P=2456&name=Brick%202%20x%206&category=%5BBrick%5D,Brick 2 x 6

1. Short intro to Large Language Models (LLMs)
2. Setup the development environment
3. Let's write the scraper

Generic Question:

Have you read/written python before?

Generic Question:

Are you familiar with HTML elements (example `<div>`)?

Generic Question:

Have you used Chrome dev tools before?

What is an LLM?

They are a very trendy topic

It's a topic that is here to stay
and it could be very useful

Hype means overselling pros
understating limitations

So... what are LLMs?

Very very large statistical models, created on huge amounts of text.



They can generate text that feels very natural
(amazing human-machine interface)



They can produce text that makes sense

Example: “the cat sits on the _____”



They can produce text that looks correct

Example: “the cat sits on the _____”

elephant, sun, chair



Imagine someone who has read **a lot** about a topic
but are not domain experts



Imagine someone who has read about a topic
but are not domain experts

That's an LLM



Imagine someone who has read about a topic
but are not domain experts

They can say the wrong thing



They can say the wrong thing

With a lot of confidence



They can say the wrong thing

With a lot of confidence

Due to hype people expect LLMs to be “smart”



Is an LLM good as a teaching assistant for programming?



The LLM has read a lot of existing good material



Simple technical issues are not controversial



LLMs can say the wrong thing

With a lot of confidence



They might nudge you to a non optimal direction

Worth a try!



Let's get to work!

vkatsikaros.com

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```



```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```



```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://example-blog.com'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```

2: I am trying to scrape the url
<https://vkatsikaros.github.io/dataharvest24-www.github.io/>
what should I change?



+ Code + Text



```
import requests
from bs4 import BeautifulSoup
```



```
url = 'https://vkatsikaros.github.io/dataharvest24-www.github.io/'
```



```
response = requests.get(url)
```



```
if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')
```

```
    for title in titles:
        print(title.get_text())
```

```
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```





+ Code + Text



✓
1s



```
import requests
from bs4 import BeautifulSoup

url = 'https://vkatsikaros.github.io/dataharvest24-www.github.io/'

response = requests.get(url)

if response.status_code == 200:
    print('Retrieved the webpage. Status code:', response.status_code)
    soup = BeautifulSoup(response.content, 'html.parser')
    titles = soup.find_all('h2')

    for title in titles:
        print(title.get_text())
else:
    print('Failed to retrieve the webpage. Status code:', response.status_code)
```



Retrieved the webpage. Status code: 200



Keyword Search Condition All Min Qty Min Price Max Price Available Items Only




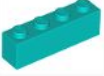


[More Options](#) ▼

9,185 Items Found Showing prices in Euro (EUR) ([more info](#))

Sort: Best match ▼

Overview (9,185) Part (7,843) Gear (503) Set (142) Original Box (142) Minifigure (122) Book (80) Instruction (67) Catalog (1) Custom Items (285)

25 per page

Part (25 out of 7,843)	Condition	Qty	Sellers	Price
 Brick 1 x 1 Brick : 3005	New	15,730,337	6,663	EUR 0.0001+
	Used	4,921,039	87,937	EUR 0.0013+
 Brick 1 x 2 Brick : 3004	New	16,800,431	6,779	EUR 0.0001+
	Used	7,076,722	101,667	EUR 0.0009+
 Brick 1 x 3 Brick : 3622	New	3,867,766	5,893	EUR 0.0001+
	Used	1,311,312	47,376	EUR 0.001+
 Brick 1 x 4 Brick : 3010	New	5,709,796	6,332	EUR 0.0001+
	Used	1,851,694	68,725	EUR 0.0027+
 Brick 1 x 6 Brick : 3009	New	2,288,682	5,834	EUR 0.0001+
	Used	829,247	45,917	EUR 0.0013+
 Brick 1 x 8 Brick : 3008	New	1,101,907	5,121	EUR 0.0001+
	Used	350,629	27,114	EUR 0.005+

Back Alt+Left Arrow

Forward Alt+Right Arrow


Reload Ctrl+R

Save as... Ctrl+S

Print... Ctrl+P

Cast...

Search images with Google

 Send to your devices

 Create QR Code for this page

Translate to English

Open in reading mode **NEW**

View page source Ctrl+U

Inspect

All Items Search

Guide Color Guide Inventories Appears In Relationships Download Add or Change Logs Credits Stores

All Item Search: Results for "brick"



⚠⚠⚠ This is NOT bricklink.com. NOT affiliated with LEGO® or Bricklink™. Test/demo page. Intended for HTML+CSS workshop.

Keyword Search Condition All Min Qty Min Price Max Price Available Items Only

9,185 Items Found Showing prices in Euro (EUR) [\(more info\)](#) Sort: Best ma

Overview (9,185) Part (7,843) Gear (503) Set (142) Original Box (142) Minifigure (122) Book (80) Instruction (67) Catalog (1) Cl

25 per page « 1 2 3 4 5 6 7 8 9 ... 314 »

Part (25 out of 7,843)	Condition	Qty	Sellers
 Brick 1 x 1 Brick : 3005	New	15,730,337	6,663
	Used	4,921,039	87,937
 Brick 1 x 2 Brick : 3004	New	16,800,431	6,779
	Used	7,076,722	101,667
 Brick 1 x 3 Brick : 3622	New	3,867,766	5,893
	Used	1,311,312	47,376
 Brick 1 x 4 Brick : 3010	New	5,709,796	6,332
	Used	1,851,694	68,725

Elements
Console
Sources
Network
Performance
Memory
Application
17
28
21

```

<html>
<head> </head>
<body id="brick-link" class>
  <div class="bl-3 hidden" id="email_marketing_modal"> </div>
  <div id="blGlobalNavContainer" class="bl-3"> </div>
  <div class="bl-clone-support">
    <center> </center> == $0
  </div>
  <div class="bl-3" id="blGlobalFooter"> </div>
  <!-- .bl-3 -->
  <div id="_idLargeImgLayerTemplate" class="blLargeImgLayer" style="display: none;"> </div>
  <div class="bl-3" id="idcipItemImageDialogContainer"> </div>
  <div id="lbOverlay" style="display: none;"> </div>
  <div id="lbMain" style="display: none;"> </div>
</body>
</html>
    
```

Styles
Computed
Layout
Event Listeners

Filter :hov .cls +

```

element.style {
}

center {
  display: block;
  text-align: -webkit-center;
  unicode-bidi: isolate;
}

Inherited from div.bl-clone-support
.bl-clone-support {
  min-height: calc(100vh - 472px);
  padding-top: 20px;
  padding-bottom: 30px;
  background-color: #ddd;
  font-size: medium;
}

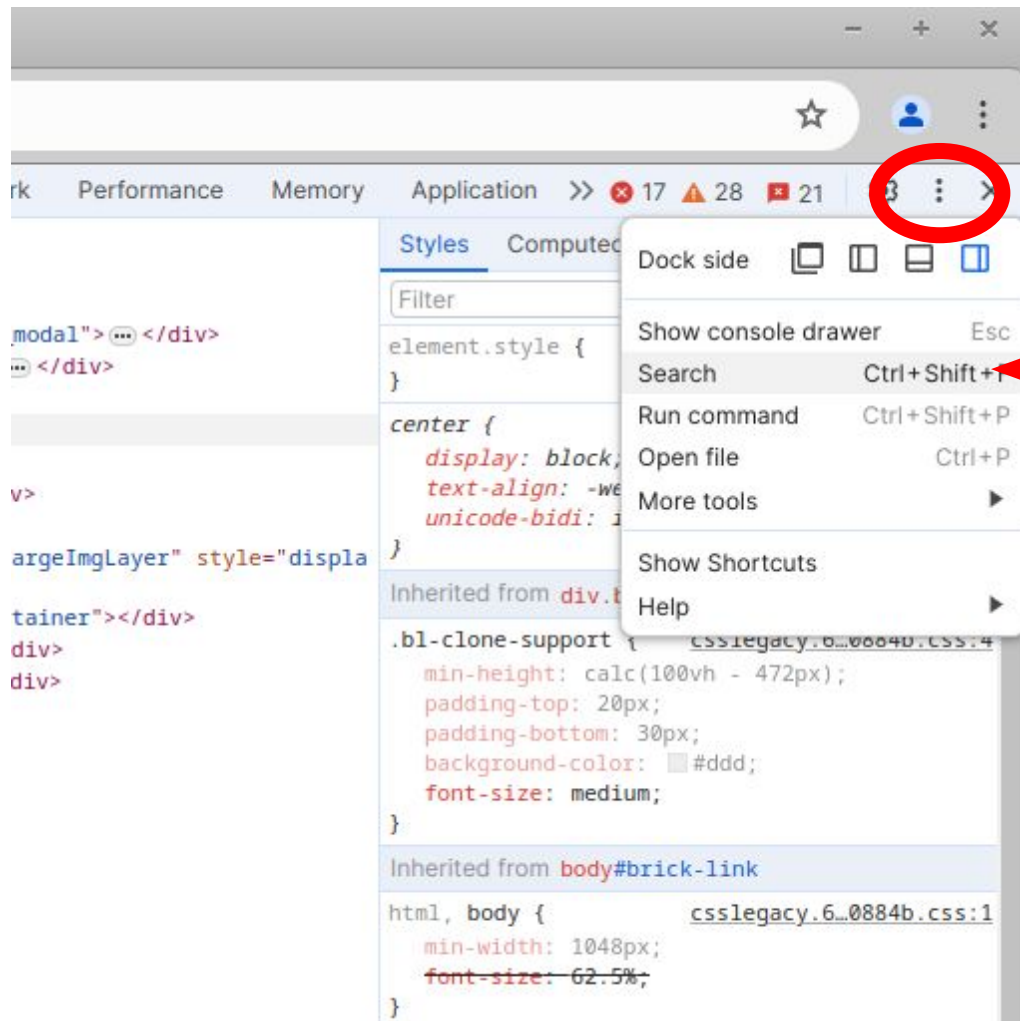
Inherited from body#brick-link
html, body {
  min-width: 1048px;
  font-size: 62.5%;
}

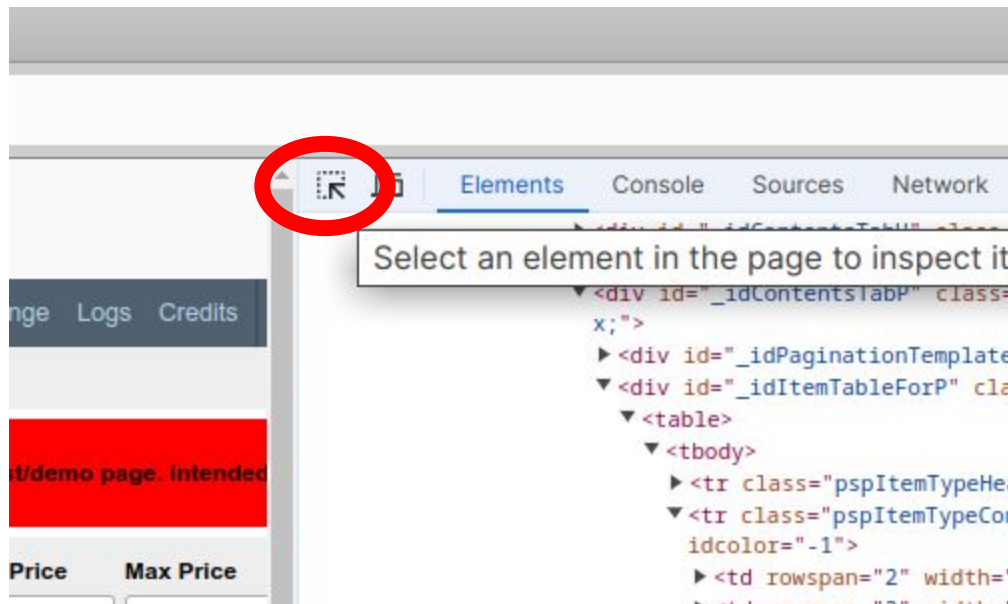
Inherited from html
:host, :root {
  --fa-font-thin: normal 100 1em/1 "Font Awesome 6 Pro";
}

:host, :root {
  --fa-font-solid: normal 900 1em/1 "Font Awesome 6 Pro";
}

:host, :root {
  --fa-font-regular: normal 400 1em/1 "Font Awesome 6 Pro";
}

:host, :root {
  --fa-font-light: normal 300 1em/1 "Font Awesome 6 Pro";
}
    
```



IS Search

Color Guide Inventories Appears In Relationships Download Add or Change Logs Credits

All Item Search: Results for "brick"

⚠⚠⚠ This is NOT bricklink.com. NOT affiliated with LEGO® or Bricklink™. Test/demo page. Intended

Keyword Search Condition Min Qty Min Price Max Price

9,185 Items Found

Overview (9,185) Part (7,843) Final Box (142) Minifigure (122) Book (80) Inst

25 per page

Part (25 out of 7,843)



a.pspitemnamelink 60.05 x 14
Color #00389A
Font 12px Tahoma, Arial
ACCESSIBILITY
Name Brick 1 x 1
Role link
Keyboard-focusable

Brick 1 x 1
Brick : 3005



Brick 1 x 2
Brick : 3004

Condition	Qty
New	15,730,33
Used	4,921,03
New	16,800,43
Used	7,076,72



Elements Console

```
style="font-s
▶ <div id="_idE
ay: none; bor
▶ <table id="_i
"0" cellpaddi
▶ <table id="_i
</table>
▶ <table id="id
cellpadding="
▶ <div id="_idT
one; padding:
▶ <div id="_idC
dding: 5px 10
▼ <div id="_idC
x;">
▶ <div id="_i
▼ <div id="_i
▼ <table>
▼ <tbody>
▶ <tr cl
▼ <tr cl
idcolc
▶ <td
▶ <td
▼ <td
<a
◀ table table tbody tr.pspitemTyp
: Console What's new Search
Aa .* h2
▼ dataharvest24-www.github.io/ vkal
348 ...="button" id="_idbtnSearch
```


It's easier to isolate and then
process the isolated parts

“divide and conquer”

Let's isolate the `<table>`

Talked about strengths and weaknesses of LLMs

Talked about strengths and weaknesses of LLMs

Took a look at python coding

Talked about strengths and weaknesses of LLMs

Took a look at python coding

Used an LLM to get answers for simple programming questions

Talked about strengths and weaknesses of LLMs

Took a look at python coding

Used an LLM to get answers for simple programming questions

Learned a few things about scraping

Questions?

