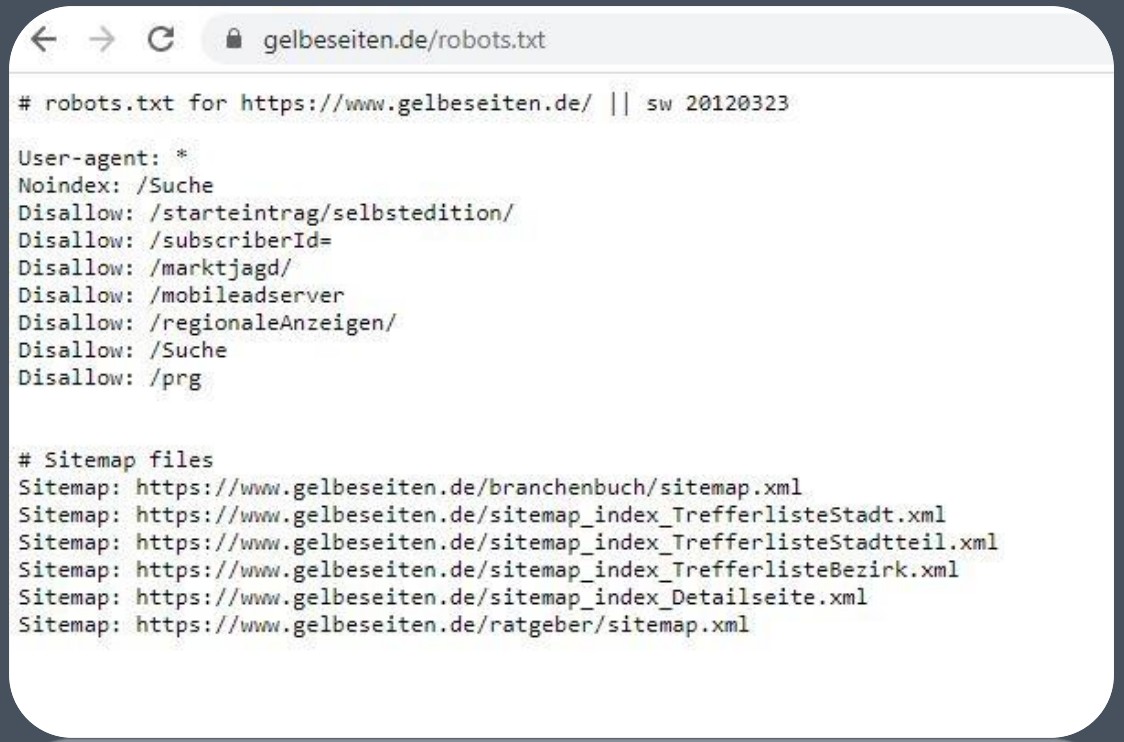

TRANSFORM8
TRANSFORM8
WEBCRAWLER-BAUANLEITUNG
BSP: GELBE SEITEN / APOTHEKEN / MÜNCHEN

Webcrawling erlaubt?

Um die Wünsche des Websiteadministrators bezüglich Crawler beachten zu können, öffnen wir www.website.de/robots.txt im Browser

Im Beispiel der Gelben Seiten ist crawlen nicht verboten und es gibt auch keine offensichtliche Spezifikation, dass man sich identifizieren muss als Webcrawler.

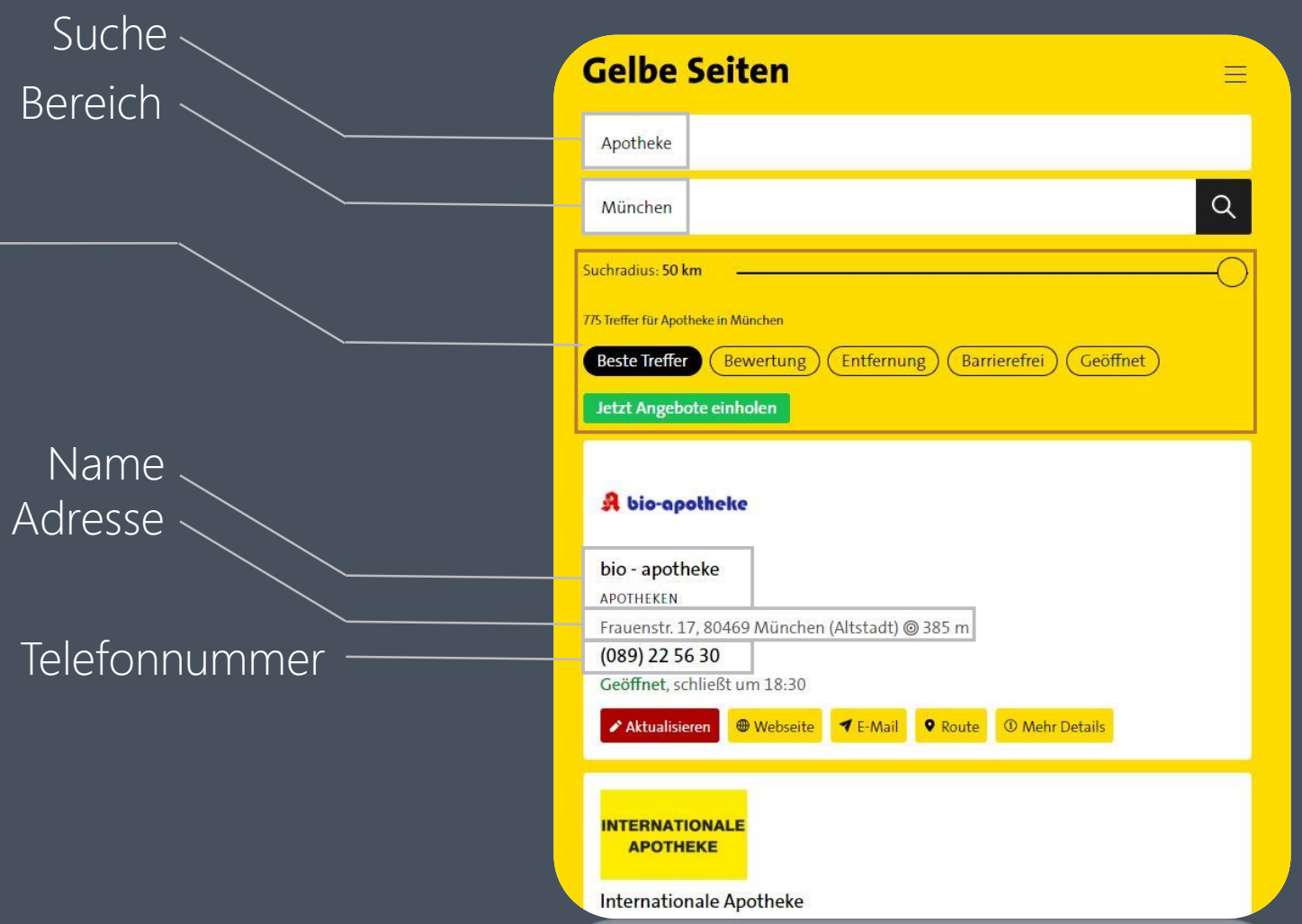
Wir können also zum nächsten Schritt, der Datensichtung, übergehen.



Datensichtung

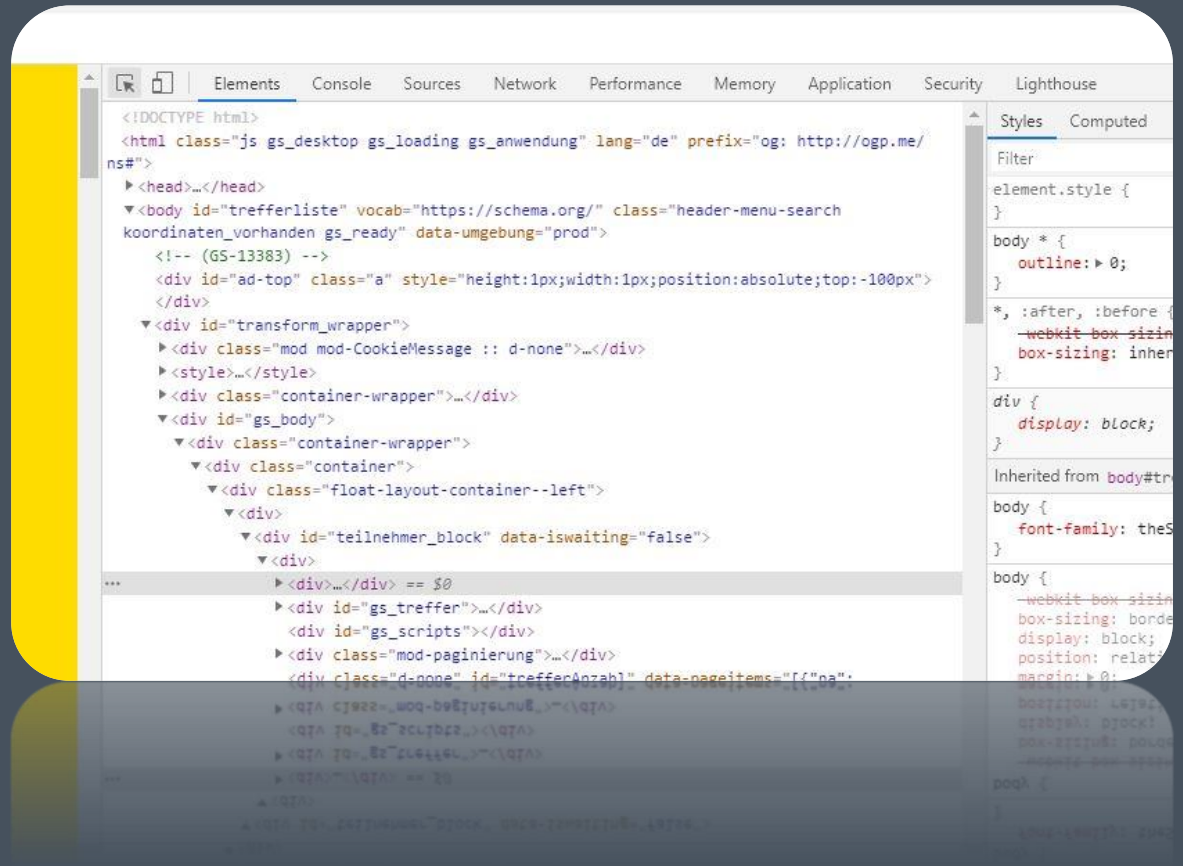
Alles was man auf einer Website sehen kann, kann gecrawlt werden.

Suchen Sie sich die Informationen aus, die Sie benötigen und definieren Sie den Bereich der Website, die Sie crawlen wollen



Der HTML-Quellcode

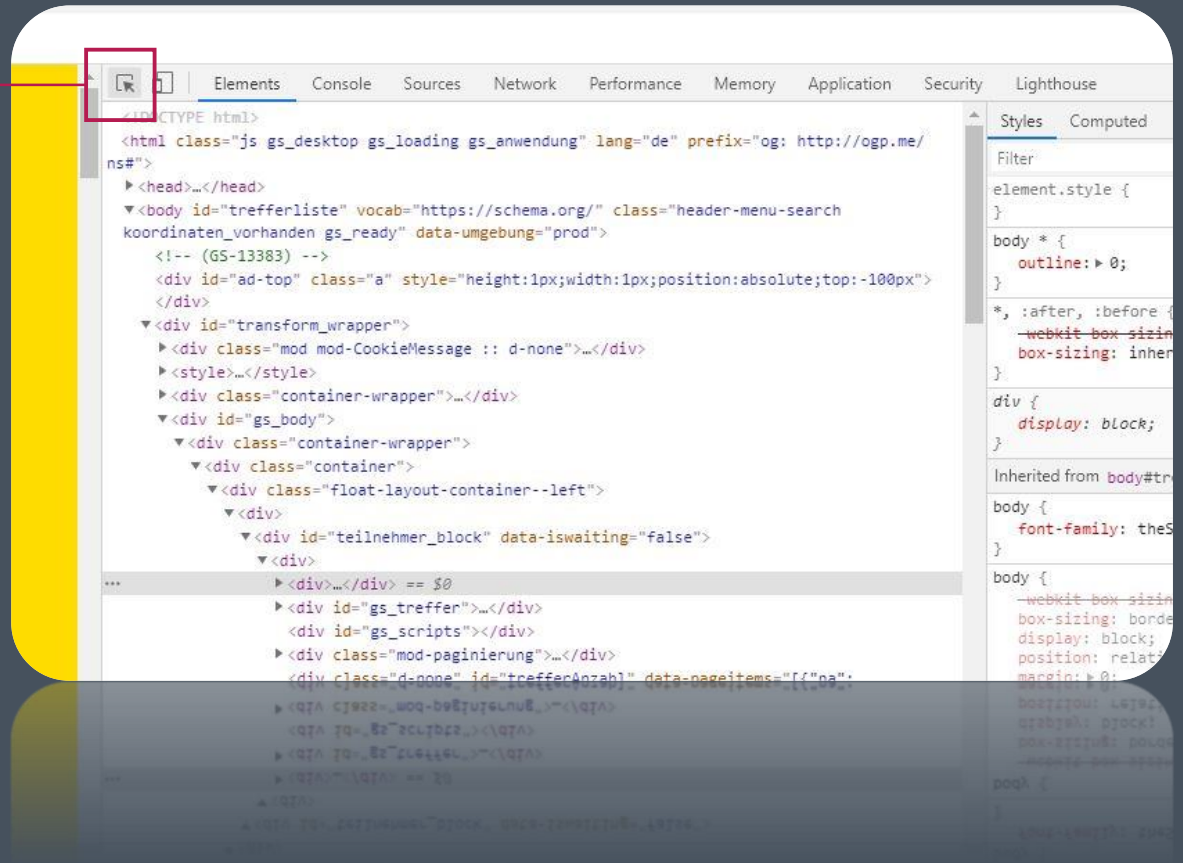
Drücken Sie im Browser F12 um sich den Quellcode der Website anzeigen zu lassen

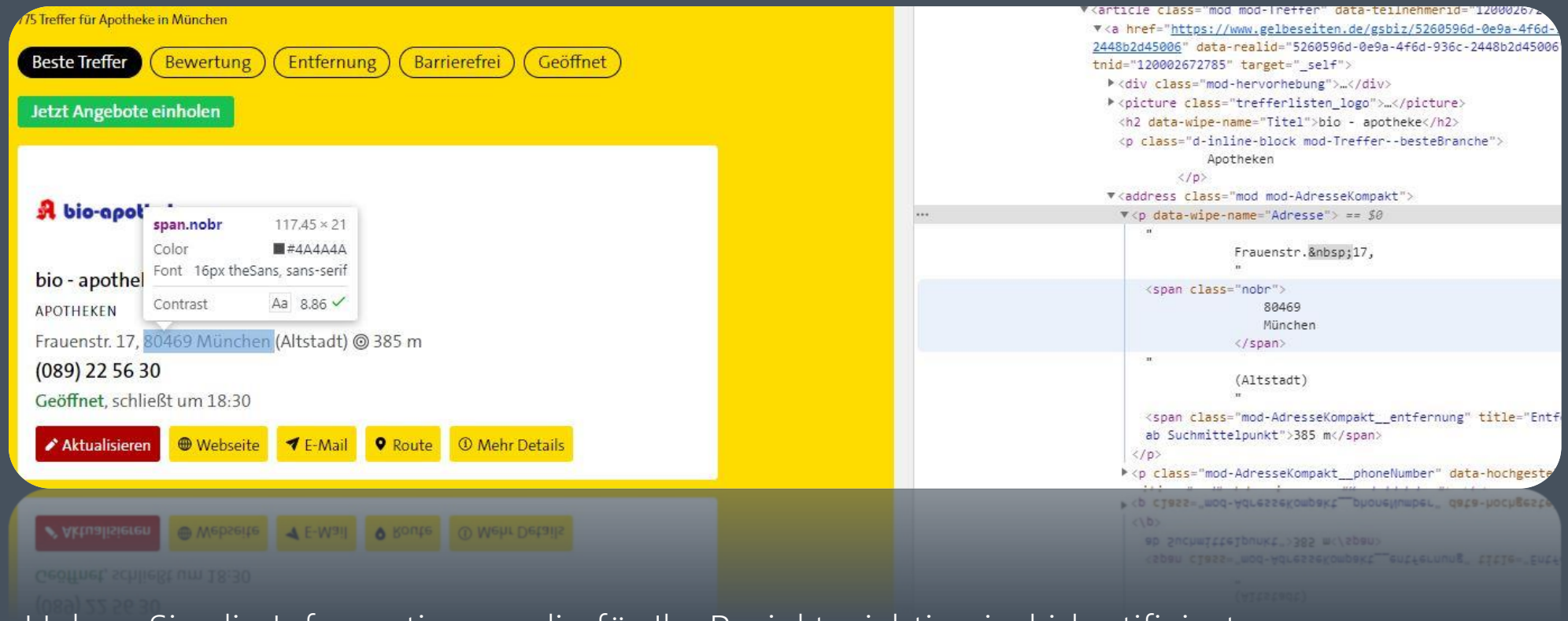


Der HTML-Quellcode

Drücken Sie auf diesen Button. Ist er gedrückt, können sie jetzt die Maus über die Website bewegen und im HTML Code wird die Stelle gehighlighted, die der Crawler abgreifen soll

Alternativ einen Rechtsklick auf die benötigte Information machen und dann „ Element untersuchen “ auswählen





Haben Sie die Informationen, die für Ihr Projekt wichtig sind identifiziert, können wir uns im HTML Code, die Stellen ansehen die für uns relevant sind.

Ziel ist es, dem Webcrawler sagen zu können, wo er im Code nach der Information suchen soll, um sie zu speichern.

TRANSFORM8

Python Code

Nötige Installationen bevor sie mit dem Coden anfangen können:

Anaconda (Download & Installation)

Anaconda ist eine Umgebung mit der Sie verschiedene Pythonversionen, Jupyter Notebook und Python-Pakete problemlos runterladen und installieren können.

Um alle für den Crawler relevanten Pakete und Programme runterzuladen,
geben Sie im Anaconda Prompt folgende commands nacheinander ein:

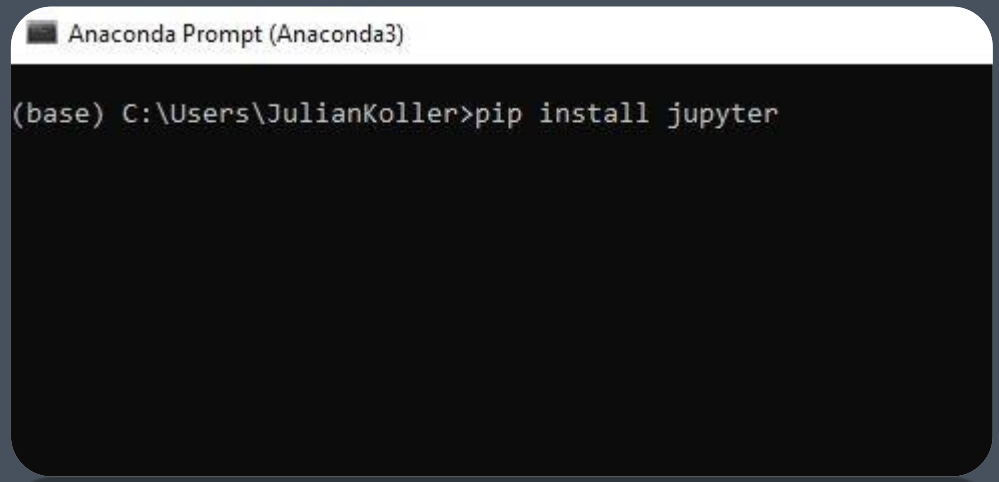
```
pip install --upgrade pip

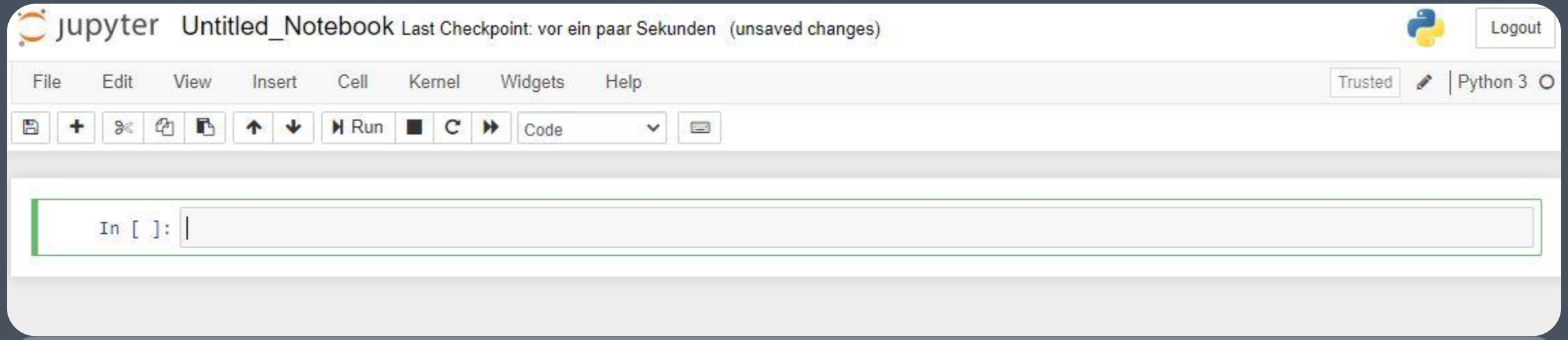
pip install jupyter

pip install beautifulsoup4

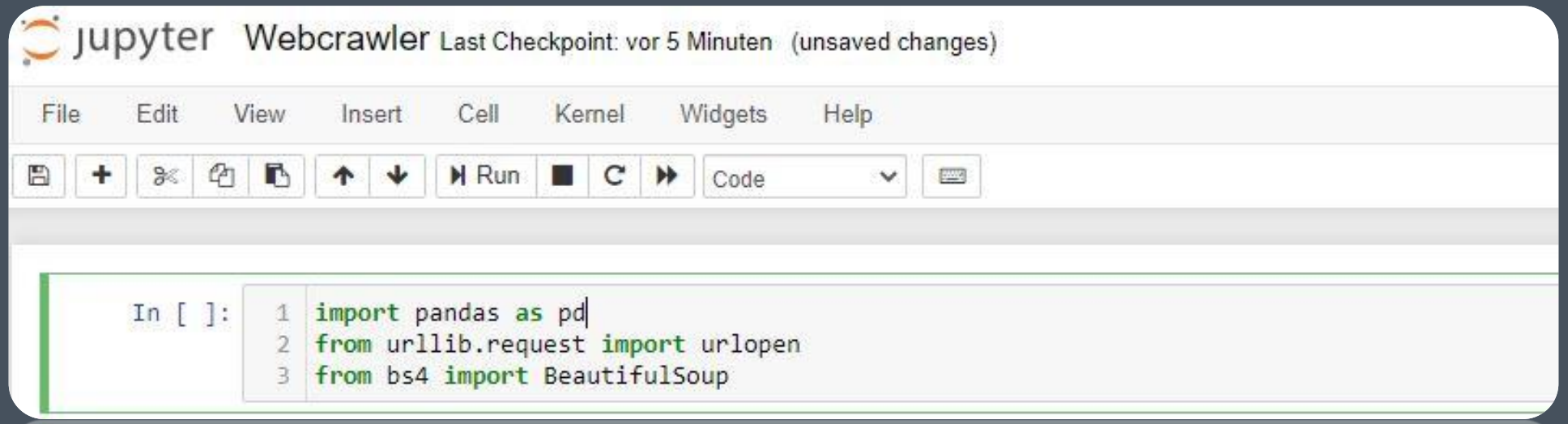
pip install Requests

pip install pandas
```



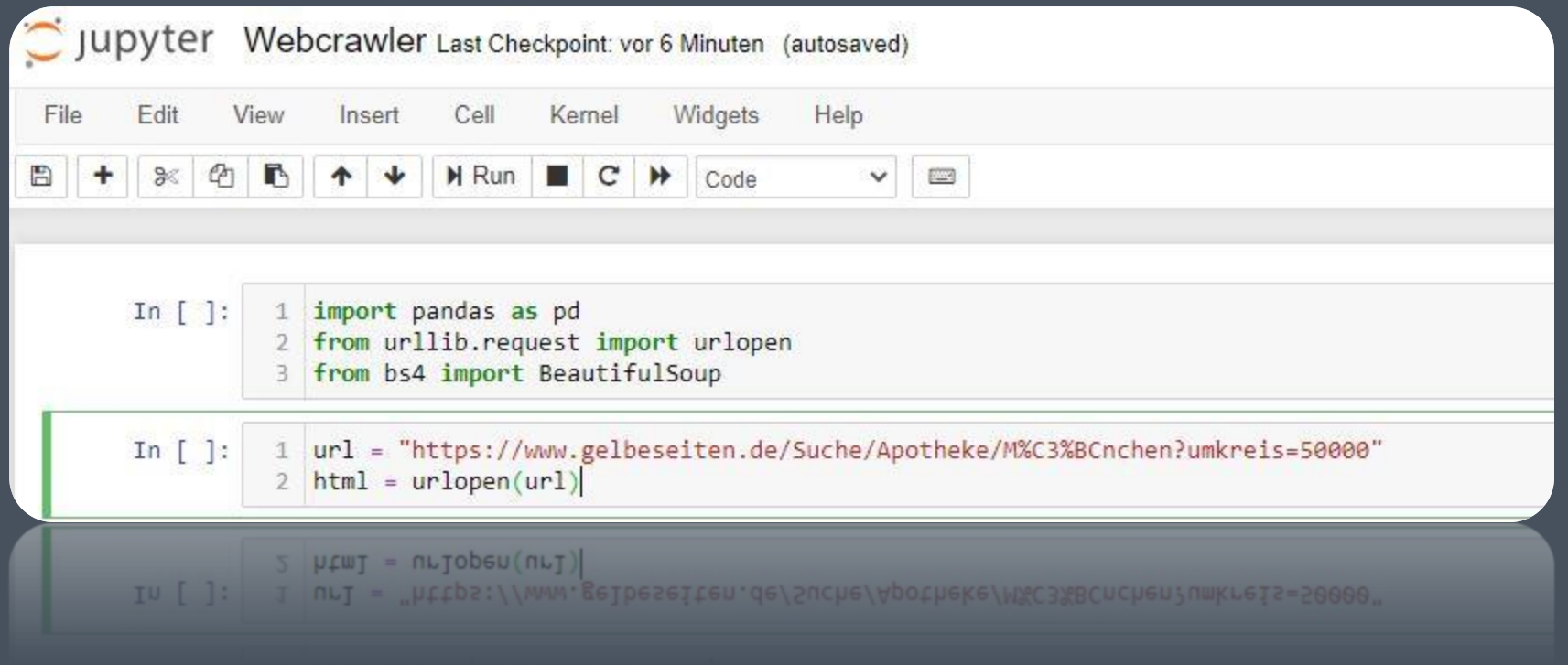


Öffnen Sie ein neues Jupyter Notebook (Python 3)
Benennen Sie es nach Ihrem Belieben
Wird der „Run “ Button betätigt, wird der ausgewählte Codeabschnitt ausgeführt



Sagen Sie dem Computer welche Python-Pakete Sie für Ihren Crawler benötigen
Wir benötigen:

- Pandas (Datenverarbeitung)
- Requests (HTML abfragen)
- BeautifulSoup (Crawler Python-Paket)



Bestimmen sie die Website URL, die Sie crawlen wollen

TRANSFORM8

jupyter Webcrawler Last Checkpoint: vor 7 Minuten (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run Code

In [ ]:
```
1 import pandas as pd
2 from urllib.request import urlopen
3 from bs4 import BeautifulSoup
```

In [ ]:
```
1 url = "https://www.gelbeseiten.de/Suche/Apotheke/M%C3%BCnchen?umkreis=50000"
2 html = urlopen(url)
```

In [ ]:
```
1 soup = BeautifulSoup(html, 'lxml')
2 x=soup.find_all("address")
3 print(x)
```

HTML Quellcode:



Aus dem HTML Quellcode der Gelben Seiten kennen wir, die Abschnitte in denen der Adresstext liegt. Wird im Code mit <address class=„ …. “ > – gekennzeichnet.

Der Computer soll alles auf der Website crawlen, was mit „ address “ gekennzeichnet ist

Output

TRANSFORM8

```
In [3]:
1 soup = BeautifulSoup(html, 'lxml')
2 x=soup.find_all("address")
3 print(x)
```

```
[<address class="mod mod-AdresseKompakt">
<p data-wipe-name="Adresse">
                        Frauenstr. 17,
                        <span class="nobr">
                                80469
                                München
                        </span>
                        (Altstadt)
                        <span class="mod-AdresseKompakt__entfernung" title="Entfernung ab Suchmittelpunkt">385 m</span>
</p>
<p class="mod-AdresseKompakt__phoneNumber" data-hochgestellt-position="end" data-wipe-name="Kontaktdaten">(089) 22 56 30</p>
</address>, <address class="mod mod-AdresseKompakt">
<p data-wipe-name="Adresse">
                        Orleansplatz 10,
                        <span class="nobr">
                                81667
                                München
                        </span>
                        (Haidhausen)
```

HTML Quellcode:

```
In [ ]:
1 Alle_Adressen=[]
2 for a in soup.find_all("address"):
3     htmladresse=a.find_all('p')[0]
4     address = htmladresse.get_text()
5     address = " ".join(address.split())
6     Alle_Adressen.append(address)
7 print(Alle_Adressen)
```



Im vorherigen Output haben wir festgestellt, dass wir zu viele Informationen crawlen
Deshalb soll der PC jetzt in jeder einzelnen „address “ Sektion nur den Textinhalt abspeichern, der von „p “ eingegrenzt wird.

Speichern Sie es in einer Liste namens „Alle_Adressen “ ab

Output

In [4]:

```
Alle_Adressen=[]
for a in soup.find_all("address"):
    htmladresse=a.find_all('p')[0]
    address = htmladresse.get_text()
    address = " ".join(address.split())
    Alle_Adressen.append(address)
print(Alle_Adressen)
```

```
['Frauenstr. 17, 80469 München (Altstadt) 385 m', 'Orleansplatz 10, 81667 München (Haidhausen) 2,4 km', 'Giesinger Bahnhofpl.
7, 81539 München (Obergiesing) 3,3 km', 'St.-Magnus-Str. 30, 81545 München (Harlaching) 4,9 km', 'Giesinger Bahnhofplatz 7, 815
39 München (Obergiesing) 3,3 km', 'Bad-Schachener-Str. 69, 81671 München (Berg am Laim) 4 km', 'Orleansplatz 10, 81667 München
(Haidhausen) 2,4 km', 'Tannenfleckstr. 2, 82194 Gröbenzell 15,8 km', 'Ludwigstr. 25, 86316 Friedberg 50,3 km', 'Stadtplatz 21,
86551 Aichach 48,7 km', 'Stadtplatz 29, 86551 Aichach 48,7 km', 'Rosenstr. 6, 80331 München (Altstadt) 134 m', 'Färbergraben 1
2, 80331 München 303 m', 'Isartorplatz 6, 80331 München (Lehel) 684 m', 'Liebigstr. 5, 80538 München (Lehel) 954 m', 'Sendlinge
r-Tor-Platz 7, 80336 München (Ludwigsvorstadt) 868 m', 'Sendlinger-Tor-Platz 7, 80336 München (Ludwigsvorstadt) 869 m', 'Trifts
tr. 6, 80538 München (Lehel) 1,1 km', 'Brienner Str. 10, 80333 München (Maxvorstadt) 669 m', 'Schillerstr. 9, 80336 München (Lu
dwigsvorstadt) 1,1 km', 'Dultstr. 1, 80331 München (Altstadt) 335 m', 'Ohlmüllerstr. 16, 81541 München (Au) 1,4 km', 'Oettingen
str. 26, 80538 München (Lehel) 1,6 km', 'Türkenstr. 80, 80799 München (Maxvorstadt) 1,6 km', 'Lindwurmstr. 97, 80337 München (I
sarvorstadt) 1,7 km', 'Auenstr. 84, 80469 München (Isarvorstadt) 1,6 km', 'Max-Planck-Str. 12, 81675 München (Haidhausen) 1,7 k
m', 'Pacellistr. 5, 80333 München (Altstadt) 579 m', 'Entenbachstr. 50, 81541 München (Au) 1,8 km', 'Ledererstr. 4, 80331 Münch
en (Altstadt) 246 m', 'Hans-Sachs-Str. 1, 80469 München (Isarvorstadt) 832 m', 'Implerstr. 26, 81371 München (Sendling) 2,7 k
m', 'Augustenstr. 116, 80798 München (Maxvorstadt) 2,1 km', 'Augustenstr. 116, 80798 München (Maxvorstadt) 2,1 km', 'Theresienh
öhe 12, 80339 München (Schwanthalerhöhe) 2,3 km', 'Goethestr. 20, 80336 München (Ludwigsvorstadt) 1,3 km', 'Färbergraben 12Rgb,
80331 München (Altstadt) 303 m', 'Keuslinstr. 17, 80798 München (Schwabing-West) 2,5 km', 'Leopoldstr. 58, 80802 München (Schwa
bing) 2,6 km', 'Nymphenburger Str. 92, 80636 München (Neuhausen) 2,7 km', 'Lindwurmstr. 103, 80337 München (Isarvorstadt) 1,7 k
m', 'Marienplatz 2, 80331 München (Altstadt) 56 m', 'Tal 13, 80331 München (Altstadt) 284 m', 'Orleansplatz 5, 81667 München (H
aidhausen) 2,3 km', 'Theatinerstr. 45, 80333 München (Altstadt) 266 m', 'Falkenturmstr. 14, 80331 München (Altstadt) 392 m', 'F
allmerayerstr. 1, 80796 München (Schwabing-West) 2,6 km', 'Deisenhofener Str. 46, 81539 München (Obergiesing) 2,8 km', 'Reichen
bachstr. 9, 80469 München (Altstadt) 413 m', 'Schönstr. 2 A, 81543 München (Untergiesing) 2,9 km']
```

TRANSFORM8

```
In [ ]:
1 df=pd.DataFrame(Alle_Adressen)
2 df1 = df[0].str.split(',', expand=True)
3 df2=df1[1].str.split(' ', expand=True)
4 df=pd.concat([df1[0], df2], axis=1, ignore_index=True)
5 df.head(10)
```

Splitten Sie das Datenset in verschiedene Spalten nach jedem Komma und Leerzeichen

Fügen Sie es nach dem splitten wieder zusammen

Zeigen Sie die ersten 10 Zeilen an

Output

TRANSFORM8

In [5]:

```
df=pd.DataFrame(Alle_Adressen)
df1 = df[0].str.split(',', expand=True)
df2=df1[1].str.split(' ', expand=True)
df=pd.concat([df1[0], df2], axis=1, ignore_index=True)
df.head(10)
```

Out[5]:

	0	1	2	3	4	5	6	7	8
0	Frauenstr. 17		80469	München	(Altstadt)	385	m	None	None
1	Orleansplatz 10		81667	München	(Haidhausen)	2	None	None	None
2	Giesinger Bahnhofpl. 7		81539	München	(Obergiesing)	3	None	None	None
3	St.-Magnus-Str. 30		81545	München	(Harlaching)	4	None	None	None
4	Giesinger Bahnhofplatz 7		81539	München	(Obergiesing)	3	None	None	None
5	Bad-Schachener-Str. 69		81671	München	(Berg	am	Laim)	4	km
6	Orleansplatz 10		81667	München	(Haidhausen)	2	None	None	None
7	Tannenfleckstr. 2		82194	Gröbenzell	15	None	None	None	None
8	Ludwigstr. 25		86316	Friedberg	50	None	None	None	None
9	Stadtplatz 21		86551	Aichach	48	None	None	None	None

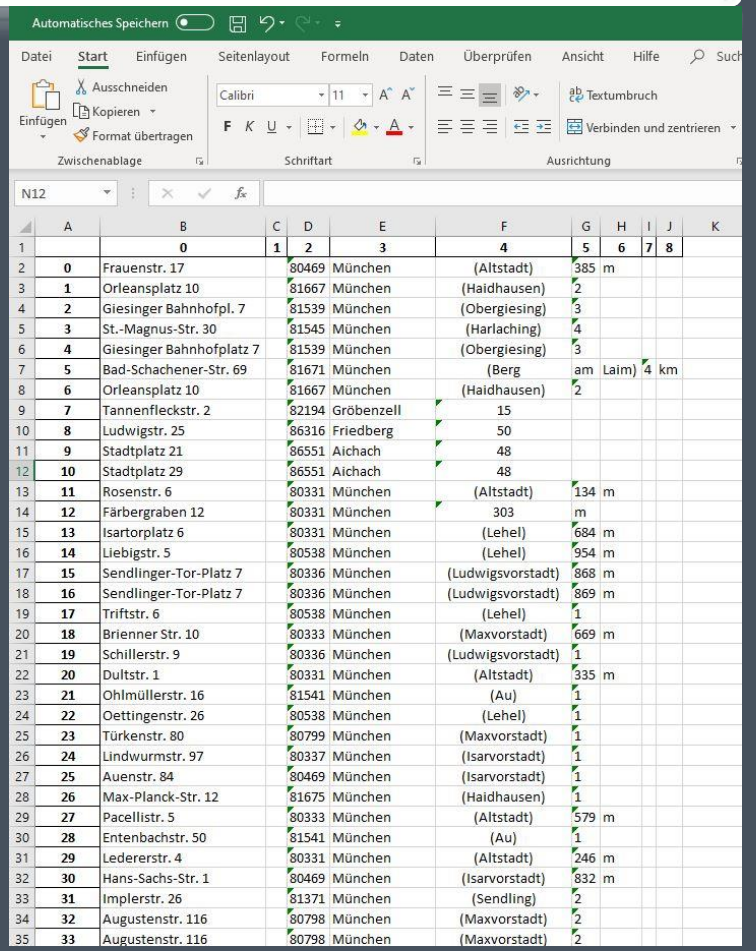
```
In [ ]:  1  df.to_excel("C:\\Users\\JulianKoller\\Webcrawler.xlsx")
```

LETZTER SCHRITT:
Speichern Sie die Daten in einer Excel-Datei „ Webcrawler “ ab

Nun können Sie den Code beliebig an Ihre Bedürfnisse anpassen und den Crawler erweitern.

Beispiele für zusätzliche Features, die einen Crawler stärker und wertvoller machen:

- Automatisches Umblättern auf die nächsten Seiten
- URL-Linkspeicherung
- Automatisches öffnen und schließen von Links die auf Detailinformationen eines Produktes führen und das anschließende Speichern der Informationen und Verlassen der Seite für jedes Produkt
- etc…



	A	B	C	D	E	F	G	H	I	J	K
1		0	1	2	3	4	5	6	7	8	
2	0	Frauenstr. 17		80469	München	(Altstadt)	385	m			
3	1	Orleansplatz 10		81667	München	(Haidhausen)	2				
4	2	Giesinger Bahnhofpl. 7		81539	München	(Obergiesing)	3				
5	3	St.-Magnus-Str. 30		81545	München	(Harlaching)	4				
6	4	Giesinger Bahnhofplatz 7		81539	München	(Obergiesing)	3				
7	5	Bad-Schachener-Str. 69		81671	München	(Berg	am	Laim)	4	km	
8	6	Orleansplatz 10		81667	München	(Haidhausen)	2				
9	7	Tannenfleckstr. 2		82194	Gröbenzell	15					
10	8	Ludwigstr. 25		86316	Friedberg	50					
11	9	Stadtplatz 21		86551	Aichach	48					
12	10	Stadtplatz 29		86551	Aichach	48					
13	11	Rosenstr. 6		80331	München	(Altstadt)	134	m			
14	12	Färbergraben 12		80331	München	303	m				
15	13	Isartorplatz 6		80331	München	(Lehel)	684	m			
16	14	Liebigstr. 5		80538	München	(Lehel)	954	m			
17	15	Sendlinger-Tor-Platz 7		80336	München	(Ludwigsvorstadt)	868	m			
18	16	Sendlinger-Tor-Platz 7		80336	München	(Ludwigsvorstadt)	869	m			
19	17	Triftstr. 6		80538	München	(Lehel)	1				
20	18	Brienner Str. 10		80333	München	(Maxvorstadt)	669	m			
21	19	Schillerstr. 9		80336	München	(Ludwigsvorstadt)	1				
22	20	Dultstr. 1		80331	München	(Altstadt)	335	m			
23	21	Ohlmüllerstr. 16		81541	München	(Au)	1				
24	22	Oettingenstr. 26		80538	München	(Lehel)	1				
25	23	Türkenstr. 80		80799	München	(Maxvorstadt)	1				
26	24	Lindwurmstr. 97		80337	München	(Isarvorstadt)	1				
27	25	Auenstr. 84		80469	München	(Isarvorstadt)	1				
28	26	Max-Planck-Str. 12		81675	München	(Haidhausen)	1				
29	27	Pacellistr. 5		80333	München	(Altstadt)	579	m			
30	28	Entenbachstr. 50		81541	München	(Au)	1				
31	29	Ledererstr. 4		80331	München	(Altstadt)	246	m			
32	30	Hans-Sachs-Str. 1		80469	München	(Isarvorstadt)	832	m			
33	31	Implerstr. 26		81371	München	(Sendling)	2				
34	32	Augustenstr. 116		80798	München	(Maxvorstadt)	2				
35	33	Augustenstr. 116		80798	München	(Maxvorstadt)	2				

Code zum Kopieren

```
import pandas as pd
from urllib.request import urlopen
from bs4 import BeautifulSoup

url = "https://www.gelbeseiten.de/Suche/Apotheke/M%C3%BCnchen?umkreis=50000"
html = urlopen(url)

soup = BeautifulSoup(html, 'lxml')
x=soup.find_all("address")
print(x)

Alle_Adressen=[]
for a in soup.find_all("address"):
    htmladresse=a.find_all('p')[0]
    address = htmladresse.get_text()
    address = " ".join(address.split())
    Alle_Adressen.append(address)
print(Alle_Adressen)

df=pd.DataFrame(Alle_Adressen)
df1 = df[0].str.split(',', expand=True)
df2=df1[1].str.split(' ', expand=True)
df=pd.concat([df1[0], df2], axis=1, ignore_index=True)
df.head(10)

df.to_excel("C:\\         \\         \\Webcrawler.xlsx")
```