

Big Data and Automated Content Analysis

I+II

Week 6 – Wednesday

»Web scraping«

Anne Kroon

a.c.kroon@uva.nl
@annekroon

Afdeling Communicatiewetenschap
Universiteit van Amsterdam

11 March 2020

Today

- ① A first intro to parsing webpages
- ② Scraping
- ③ Scaling up
- ④ Next steps

Let's have a look of a webpage with comments and try to understand the underlying structure.



HEADLINES

OS-OS
Daar is pathologische loutenaar Reinhinkel weer!

Kijk. Zo leren knijps in Tsjelche over de EU

GeenStijl
Samen op iPhone
• **GeenStijl**
• **GeenStijl**
[www.geenstijl.nl](#)

Gezocht. Dief met tatoeage en motorfiets

Wierdjes: 'Hey paupers, de EU is fantastisch hoor'

Arandiel
Wildplassen is ook vrijheid

Vreemde vogel vogelvrij bij dodenherdenking Dam

Aangifte tegen GeenStijl - De Q en A Kloosier

Generaal van Uhm: 'Polderhiedieten zijn



Fimfje uit het rauwrealistische archief van Amsterdamse stadsneest Frank Buis. En met 'rauwrealistisch' bedoelen wij: de beelden die u niet ziet op de stream-met-nietszeggend-commentaar-in-stemmige-fluisterdon van de Staatsomroep. Vlak voor de Dodenherdenking dook bovenstaande dodo op de Dam op. Plastic Hannibal Lecter maskertje voor de mond. Rare oranje klomp aan de voet. Geoguhulzen in de kroon. Daar wilde de politie wel even mee babbelen, zo vlak voor de aankomst van de koninklijke nazi-nazet en zijn Junta-meisje. Want doden herdenken op je eigen manier is 1 ding. Maar jezelf daarbij koedig kloeden valt buiten de voorschriften van de vrijheid. Wat moesten die 69 knijps in het buitenlandse gasten-vak wel niet denken? Daarom. Goed dat de politie deze malle meneer en zijn aluminium lunchpakketje evenwies naar een veilige afstand van de vrijheidsviering voerde. (Video: Frank Buis/Royal Press Amsterdam)

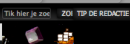
Van Rossem | 05-05-14 | 11:11 | Link | 100 reacties | [...](#)

REACTIES

Tja, ik zou m ook niet in mn buurt willen hebben, die vreemde vogel.

Lepo | 05-05-14 | 11:13

Volgend jaar stropdascontrole voor de heren en hoofdjescheck voor de dames. De volledige lijst van goedgekeurde kleding kunt u vinden op Postbus51.nl.



POWNEWS

- POWNEWS - ELKE WERKDAG, 22:45, NETO**
Gasten Delft strijdt tegen Volker
- 16:40 Jongens melden zich na mishandeling
- 16:27 Polio gaat wereld slopen
- 16:24 Verkeersgeposten plakken niet
- 15:40 Stormfe - celstraf in Zuid-Afrika
- 15:36 Update: ontvoering bewijst
- 15:26 Timmermans apt op rijn over Rude
- 15:16 Tientallen migranten verdronken
- 15:13 Nette vrouwen op superglijbaan

GEENSTIJL TV



DASKAPITAL

16:58 Pensoerpolder jaagt fondsen naar België

REAGUURSELS

Tsja, ik zou m ook niet in mn buurt willen hebben, die vreemde vogel.

Lepo | 05-05-14 | 11:13



Volgend jaar stropdascontrole voor de heren en hoedjescheck voor de dames. De volledige lijst van goedgekeurde kleding kunt u vinden op Postbus51.nl.

Uiteraard bent u extra welkom als u Abercrombie & Fitch draagt.

rara | 05-05-14 | 11:15



Gewoon even de afdeling psychiatrie bellen, wie ze missen: Klaar!

Mazzeltov | 05-05-14 | 11:16



netjes opgelost toch?

--sql error-- | 05-05-14 | 11:16





view-source:https://www.geenstijl.nl/3945571/das_toch_niet_normaal/

Aan de slag Press This Atlassian Cloud

```
n class="divider">|</span>
n class="datetime">05-05-14 | 23:34</span>
```

```
class="reportcomment" title="Deze reactie is in overtreding met de huisregels."></a>
er>
```

```
e>
```

```
article class="comment col-xs-12 no-y-padding"
commentid="192631041"
192631041">
ss="anchor-pos" id="cid_192631041"></div>
```

```
class="cmt-content"><p>Een VZ800 Marauder, wat een giga kutmotor is dat. Na 10 km. heb je al
```

```
er>
```

```
span class="username">tiswat</span>
```

Let's make a plan!

Which elements from the page do we need?

- What do they mean?
- How are they represented in the source code?

How should our output look like?

- What *lists* do we want?
- ...

And how can we achieve this?

A first plan

A first plan

- 1 Download the page

A first plan

- 1 Download the page
 - Possibly taking measures to deal with cookie walls, being blocked, etc.

A first plan

- 1 Download the page
 - Possibly taking measures to deal with cookie walls, being blocked, etc.
- 2 Remove all line breaks (`\n`, but maybe also `\n\r` or `\r`) and TABs (`\t`): We want one long string

A first plan

- ❶ Download the page
 - Possibly taking measures to deal with cookie walls, being blocked, etc.
- ❷ Remove all line breaks (`\n`, but maybe also `\n\r` or `\r`) and TABs (`\t`): We want one long string
- ❸ Try to isolate the comments
 - Do we see any pattern in the source code? \Rightarrow *last week: if we can see a pattern, we can describe it with a regular expression*

```
1 import requests
2 import re
3
4 URL = 'http://www.geenstijl.nl/mt/archieven/2014/05/
    das_toch_niet_normaal.html'
5
6 # ugly workaround to circumvent cookie wall, not of interest for today
7 client = requests.session()
8 r = client.get(URL)
9 cookies = client.cookies.items()
10 cookies.append(('cpc','10'))
11 response = client.get(URL,cookies=dict(cookies))
12 # end workaround
13
14 tekst=response.text.replace("\n"," ").replace("\t"," ")
15
16 comments=re.findall(r'<div class="cmt-content">(.*?)</div>',tekst)
17 print("There are",len(comments),"comments")
18 print("These are the first two:")
19 print(comments[:2])
```

Some remarks

The regexp

- `. *?` instead of `. *` means *lazy* matching. As `. *` matches everything, the part where the regexp should stop would not be analyzed (*greedy* matching) – we would get the whole rest of the document (or the line, but we removed all line breaks).

Some remarks

The regexp

- `. * ?` instead of `. *` means *lazy* matching. As `. *` matches everything, the part where the regexp should stop would not be analyzed (*greedy* matching) – we would get the whole rest of the document (or the line, but we removed all line breaks).
- The parentheses in `(. * ?)` make sure that the function only returns what's between them and not the surrounding stuff (like `<div>` and `</div>`)

Some remarks

The regexp

- `.*?` instead of `.*` means *lazy* matching. As `.*` matches everything, the part where the regexp should stop would not be analyzed (*greedy* matching) – we would get the whole rest of the document (or the line, but we removed all line breaks).
- The parentheses in `(.*?)` make sure that the function only returns what's between them and not the surrounding stuff (like `<div>` and `</div>`)

Optimization

- Parse usernames, date, time, ...
- Replace `<p>` tags

Further reading

Doing this with other sites?

- It's basically puzzling with regular expressions.
- Look at the source code of the website to see how well-structured it is.

OK, but this surely can be doe more elegantly? Yes!

Scraping

Geenstijl-example

- Worked well (and we could do it with the knowledge we already had)

Scraping

Geenstijl-example

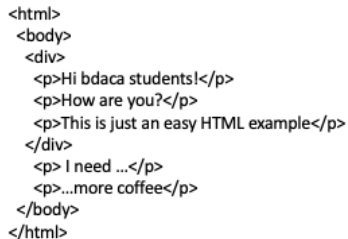
- Worked well (and we could do it with the knowledge we already had)
- But we can also use existing parsers (that can interpret the structure of the html page)

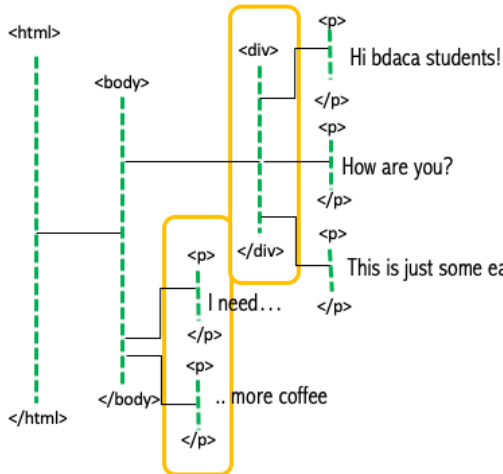
Scraping

Geenstijl-example

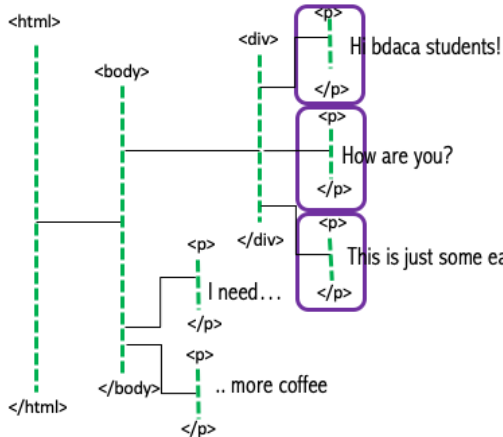
- Worked well (and we could do it with the knowledge we already had)
- But we can also use existing parsers (that can interpret the structure of the html page)
- especially when the structure of the site is more complex

```
<html>
<body>
  <div>
    <p>Hi bdaca students!</p>
    <p>How are you?</p>
    <p>This is just an easy HTML example</p>
  </div>
  <p>I need ...</p>
  <p>...more coffee</p>
</body>
</html>
```

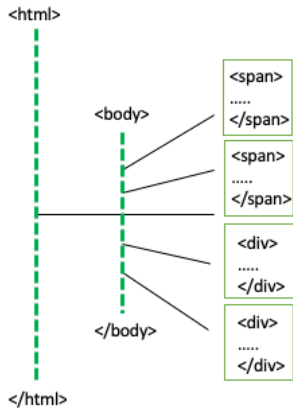




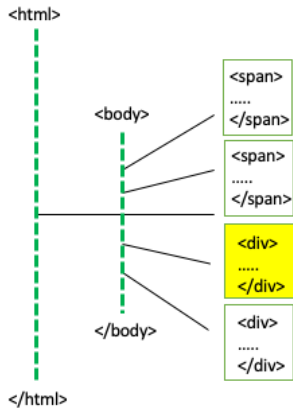
```
<html>
<body>
  <div>
    <p>Hi bdaca students!</p>
    <p>How are you?</p>
    <p>This is just an easy HTML example</p>
  </div>
  <p>I need ...</p>
  <p>...more coffee</p>
</body>
</html>
```

```
<html>
<body>
  <div>
    <p>Hi bdaca students!</p>
    <p>How are you?</p>
    <p>This is just an easy HTML example</p>
  </div>
  <p> I need ...</p>
  <p>...more coffee</p>
</body>
</html>
```



XPATH = `"/html/body/div[1]"` refers to which element?



XPATH = `"/html/body/div[1]"` refers to which element?

What do we need?

- the URL (of course)

What do we need?

- the URL (of course)
- the XPATH of the element we want to scrape (you'll see in a minute what this is)

What do we need?

- the URL (of course)
- the XPATH of the element we want to scrape (you'll see in a minute what this is)

The following example is based on <https://www.kieskeurig.nl/smartphone/product/3518001-samsung-galaxy-a5-2017-goud/reviews>.
It uses the module `lxml`

Samsung Galaxy A5 (2017) zwart / 16 GB - Reviews

Reviewscore | 8,7 [Lees 1049 reviews](#) [Schrijf een review](#)[Favorieten](#) [Vergelijk](#) [Prijscalculator](#)

6 Afbeeldingen

Testpanel score 8,2 - april 2017

Laagste prijzen

-1% prijsdaling

> Mediamarkt	€ 267,00
> Bol.com	€ 275,00
> Mobiel.nl	€ 279,00

Specificaties

4,7 inch scherm

16 GB opslag

Android

[Bekijk 3 andere uitvoeringen](#)

Samsung Galaxy A5 (2017) zwart / 16 GB

[Prijzen \(9\)](#)[Overzicht](#)[Abonnementen](#)[Specificaties](#)[Reviews \(1049\)](#)Review type: [Consumenten reviews \(1049\)](#)

Gemiddelde beoordeling

8,7

De gemiddelde beoordeling is gebaseerd op de 1049 reviews die zijn achtergelaten op Kieskeurig.nl.

Help anderen en geef jouw mening over dit product. We verloten iedere maand onder alle reviewers 2 x € 250,-.

[Schrijf een review](#)

Kieskeurig.nl review criteria:

Geluidskwaliteit

Mogelijkheden

Gebruiksgemak

Vormgeving

Degelijkheid

Goed

Goed

Uitstekend

Uitstekend

Goed

Echte reviews van echte gebruikers, door ons gecontroleerd op betrouwbaarheid.

Sorteer op: [Meest nuttig eerst](#)BartTilburg
21-03-2017

Product paar weken in bezit

Verliefd op de Samsung Galaxy A5 2017

✓ Ja, ik beveel dit product aan

Wat een fenomenale smartphone is de nieuwe Samsung Galaxy A5 2017 geworden, lees hier waarom.

Bij het openen van de verpakking valt direct op dat dit toestel niet onder doet voor een smartphone uit het duurste segment. Ik heb gekozen voor de kleur sand/gold, een mooie champagneachtige gouden kleur die erg opvalt. Het toestel heeft een aluminium behuizing en zowel de voorkant als de achterkant van de Samsung Galaxy A5 2017 zijn geheel van glas en er is zelfs een

Algemene score

Reviewscore | 10,0

Review criteria:

Geluidskwaliteit

Mogelijkheden

Gebruiksgemak

Uitstekend

Uitstekend

Uitstekend

Verliefd op de Samsung Galaxy A5 2017

✓ Ja, ik beveel dit product aan

Wat een fenomenale smartphone is de nieuwe Samsung Galaxy A5 2017 geworden, lees hier waarom.

Bij het openen van de verpakking valt direct op dat dit toestel niet onder doet voor een smartphone uit het duurste segment. Ik heb gekozen voor de kleur sand/gold, een mooie champagneachtige gouden kleur die erg opvalt. Het toestel heeft een aluminium behuizing en zowel de voorkant als de achterkant van de Samsung Galaxy A5 2017 zijn geheel uit aluminium vervaardigd. De vingerafdruksensor aanwezig. Dit alles geeft het toestel een premium look and feel, prachtig! In de verpakking zijn ook accessoires die je nodig hebt zoals een oplaadkabel, een hoofdtelefoon én de bijzondere USB-C kabel kan je dit toestel versnellen. Het toestel heeft echt alles aan boord wat je mag verwachten en zelfs

Algemene score

Reviewscore | 10

Review criteria

Geluidskwaliteit

Mogelijkheden

Gebruiksgemak

Copy

Select All

Search Google for "Wat een fenomen..."

View Selection Source

Inspect Element (Q)



Hebben!

BartTilburg
21-03-2017

Product paar weken in bezit

Verliefd op de Samsung Galaxy A5 2017

✓ Ja, ik beveel dit product aan

div.reviews-single__text | 477.5 × 909

Wat een fenomenale smartphone is de nieuwe Samsung Galaxy A5 2017 geworden, lees hier waarom.

Bij het openen van de verpakking valt direct op dat dit toestel niet onder doet voor een smartphone uit het duurste segment. Ik heb gekozen voor de kleur sand/gold, een mooie champagneachtige gouden kleur die erg opvalt. Het toestel heeft een aluminium behuizing en zowel de voorkant als de achterkant van de Samsung Galaxy A5 2017 zijn geheel van glas.

Algemeen

Reviews

Review

Geluid

Mogelijk

Gebruik

Verpakking

Inspector

Console

Debugger

{ } Style Editor

Performance

Memory

Network

Storage

Search HTML

Rules

```
<div id="6025675" class="reviews-single" data-expand="more-reviews">
  <div class="reviews-single__meta"></div>
  <div class="reviews-single__content-wrap">
    <div class="reviews-single__hero"></div>
    <div class="reviews-single__content">
      <div class="reviews-single__text"></div>
      <div class="reviews-single__individual-rating"></div>
      <div class="reviews-single__comments-votes"></div>
      <div id="review-6025675" class="reviews-single__comments-wrapper" data-context="review-6025675"></div>
    </div>
  </div>
```

div#6025675.reviews-single > div.reviews-single__content-wrap > div.reviews-single__content > div.reviews-single__text

Samsung Galaxy A5 (20... [Software Updater]

Hebben!



Inspector



Console



Debugger



```
<div id="6025675" class="review
```



```
<div class="reviews-single_m
```



```
<div class="reviews-single_c
```



```
<div class="reviews-single_
```



```
<div class="reviews-single
```



```
<div class="reviews-single
```



```
<div class="reviews-single__individual-rating"></div>
```



```
<div class="reviews-single__comments-votes"></div>
```



```
<div id="review-6025675" class="reviews-single_comments-wrapper" data-context="review-
```

Delete Node

Attributes

hover

active

focus

Copy

Paste

Expand All

Collapse

Scroll Into View

Screenshot Node

Use in Console

Show DOM Properties

Inner HTML

Outer HTML

CSS Selector

CSS Path

XPath

Image Data-URL

But you can also create your XPATH yourself

There are multiple different XPATHs to address a specific element.
Some things to play around with:

- // means 'arbitrary depth' (=may be nested in many higher levels)

But you can also create your XPATH yourself

There are multiple different XPATHs to address a specific element.
Some things to play around with:

- // means 'arbitrary depth' (=may be nested in many higher levels)
- * means 'anything'. (p[2] is the second paragraph, p[*] are all

But you can also create your XPATH yourself

There are multiple different XPATHs to address a specific element.
Some things to play around with:

- `//` means 'arbitrary depth' (=may be nested in many higher levels)
- `*` means 'anything'. (`p[2]` is the second paragraph, `p[*]` are all
- If you want to refer to a specific attribute of a HTML tag, you can use `@`. For example, every `*[@id="reviews-container"]` would grap a tag like `<div id=reviews-container" class="'user-content'`

But you can also create your XPATH yourself

There are multiple different XPATHs to address a specific element.
Some things to play around with:

- `//` means 'arbitrary depth' (=may be nested in many higher levels)
- `*` means 'anything'. (`p[2]` is the second paragraph, `p[*]` are all)
- If you want to refer to a specific attribute of a HTML tag, you can use `@`. For example, every `*[@id="reviews-container"]` would grap a tag like `<div id=reviews-container" class="'user-content'`
- Let the XPATH end with `/text()` to get all text

But you can also create your XPATH yourself

There are multiple different XPATHs to address a specific element.
Some things to play around with:

- `//` means 'arbitrary depth' (=may be nested in many higher levels)
- `*` means 'anything'. (`p[2]` is the second paragraph, `p[*]` are all
- If you want to refer to a specific attribute of a HTML tag, you can use `@`. For example, every `*[@id="reviews-container"]` would grab a tag like `<div id=reviews-container" class="'user-content'`
- Let the XPATH end with `/text()` to get all text
- Have a look at the source code (via 'inspect elements') of the web page to think of other possible XPATHs!

Let's test it!

`https://www.kieskeurig.nl/wasmachine/product/
2483630-siemens-wmn16t3471/reviews`

Let's scrape them!

```
1 from lxml import html
2 import requests
3
4 response = requests.get('https://www.kieskeurig.nl/wasmachine/product
    /2483630-siemens-wmn16t3471/reviews')
5 tree = html.fromstring(response.text)
6
7 # we extract all relevant elements using their XPATH
8 reviews = tree.xpath('//div[@class="reviews-single__text"]')
9 # alternatively, we can use their CSS selector:
10 # reviews = tree.cssselect("div.reviews-single__text")
11
12 # but we don't want the elements, we want their text
13 review_texts = [e.text_content().strip() for e in reviews]
14
15 print (len(reviews),"reviews scraped. Showing the first 60 characters:")
16 i=0
17 for review in review_texts:
18     print("Review",i,":",review[:60])
19     i+=1
```

If you want to use CSS selectors, you need to `sudo pip3 install cssselect` first



The output – perfect!

```
1 20 reviews scraped. Showing the first 60 characters:
2 Review 0 : Siemens WMN16T3471 nu 4 maanden in gebruik in massagesalon.
3 Review 1 : Na een eerder positief review kort na aankoop nu een bijgest
4 Review 2 : Helaas ben ik teleurgesteld in dit product wegens de navolge
5 Review 3 : Ik ben heel blij met mijn nieuwe wasmachine:
6
7 Wat is hij st
8 Review 4 : Ik heb de wasmachine nu net een paar dagen in huis en heb al
9 Review 5 : Na 25 jaar hebben we afscheid moeten nemen van onze degelijk
```

Scaling up

But this was on *one* page only, right?

Next step: Repeat for each relevant page.

Possibility 1: Based on url schemes

If the url of one review page is

`https://www.hostelworld.com/hosteldetails.php/
ClinkNOORD/Amsterdam/93919/reviews?page=2`
...then the next one is probably?

But this was on *one* page only, right?

Next step: Repeat for each relevant page.

Possibility 1: Based on url schemes

If the url of one review page is

`https://www.hostelworld.com/hosteldetails.php/
ClinkNOORD/Amsterdam/93919/reviews?page=2`
...then the next one is probably?

⇒ you can construct a list of all possible URLs:

```
1 baseurl = 'https://www.hostelworld.com/hosteldetails.php/ClinkNOORD/  
    Amsterdam/93919/reviews?page='  
2  
3 allurls = [baseurl+str(i+1) for i in range(20)]
```

But this was on *one* page only, right?

Next step: Repeat for each relevant page.

Possibility 2: Based on XPATHs or CSS Selectors

Use XPATH to get the url of the next page (i.e., to get the link that you would click to get the next review)

Recap

General idea

- ➊ Identify each element by its XPATH or CSS Selector (look it up in your browser)
- ➋ Read the webpage into a (loooooong) string
- ➌ Use the XPATH or CSS Selectors to extract the relevant text into a list (with a module like lxml)
- ➍ Do something with the list (preprocess, analyze, save)
- ➎ Repeat

Alternatives: scrapy, beautifulsoup, regular expressions, ...

Last remarks

There is often more than one way to specify an XPATH

- ① Sometimes, you might want to use a different suggestion to be able to generalize better (e.g., using the *attributes* rather than the *tags*)
- ② in that case, it makes sense to look deeper into the structure of the HTML code, for example with “Inspect Element” and use that information to play around with different possibilities

BartTilburg
21-03-2017

Product paar weken in bezit

Verliefd op de Samsung Galaxy A5 2017

✓ Ja, ik beveel dit product aan

div.reviews-single__text | 477.5 x 909

Wat een fenomenale smartphone is de nieuwe Samsung Galaxy A5 2017 geworden, lees hier waarom.

Bij het openen van de verpakking valt direct op dat dit toestel onder doet voor een smartphone uit het duurste segment. Ik heb gekozen voor de kleur sand/gold, een mooie champagneachtige gouden kleur die erg opvalt. Het toestel heeft een aluminium behuizing en zowel de voorkant als de achterkant van de Samsung Galaxy A5 2017 zijn geheel van glas en er is zelfs een

Console

Debugger

Style Editor

Performance

Memory

Network

Storage

Search HTML

```
id="6025675" class="reviews-single" data-expand="more-reviews">
  class="reviews-single__meta"></div>
  class="reviews-single__content-wrap">
    div class="reviews-single__hero"></div>
    div class="reviews-single__content">
      div class="reviews-single__text"></div>
    div class="reviews-single__individual-rating"></div>
    div class="reviews-single__comments-votes"></div>
    div id="review-6025675" class="reviews-single__comments-wrapper" data-context="review-6025675"></div>
  div>
v>
```

5.reviews-single

div.reviews-single__content-wrap

div.reviews-single__content

div.reviews-single__text

Samsung Galaxy A5 (20...

[Software Updater]

Next steps

Friday

- Write a scraper for a website of your choice!
- Choose an easy site where you do not have to log on and where there is no dynamically generated content (if you need that, look at the Appendix on Selenium)

Friday

- Write a scraper for a website of your choice!
- Choose an easy site where you do not have to log on and where there is no dynamically generated content (if you need that, look at the Appendix on Selenium)

Scraping can be difficult, but it is also one of the most important data collection methods. Chances are very high you'll need it as a part of your final project. Try writing a scraper, and ask questions next week.



```

In [35]: ratings = tree.xpath("//*[ @class='rating-ratingValue']/text()")
In [36]: restaurants = tree.xpath("//*[ @class='restaurantSelection-name']/text()")
In [37]: len(ratings)
Out[37]: 10
In [38]: len(restaurants)
Out[38]: 10
In [39]: for i in range(10):
    print(restaurants[i].strip(), ratings[i])
    ....:
Risto Enoteca PepeNero 9.3
Cafe Sage 9.2
The Lobby Fizeastraat Restaurant & Bar (Hotel V) 9.1
Brasserie Ambassade 9.1
Tulsi Indian restaurant 9.1
Zaza's 9
Jacobz 9
Bistrot des Alpes 9
Restaurant Jaspers 8.9
En Pluche 8.9
In [40]:
    
```