MAZ-CAS-DDJ / kurs_18_19

Branch: master ▾ kurs_18_19 / 06 APIs, Scraping I / 03 BeautifulSoup .ipynb Find file Copy path

 barjacks 07 013a690 on 7 Aug

1 contributor

1697 lines (1696 sloc) 55.7 KB

Einführung in BeautifulSoup, einlesen von Files

Und als aller Erstes wollen wir die Developer Tools kennenlernen:

- Gehen wir auf: www.srf.ch.
- Öffnen den Chrome Browser. Die Developer Tools findest DU im Menü unter More Tools -> Developer Tools. Shortcuts: F12, Str-Ctr-I oder Cmd + Opt + I (alle Browser die Dev Tools, wir fokussieren hier auf den Chrome Browser)
- Der Reiter, der uns interessiert, ist: “Elements”. Steuere ihn an.
- Wir können hier zwischen Mobilansicht und Desktop wechseln
- Oder Elemente auf der Website suchen. Wähle den Pfeil an. Und suche nun etwas auf der Website.
- Hier können wir auch Texte manipulieren, teste es auf www.srf.ch.
- Aber das interessiert uns nicht. Wir wollen die Struktur der Website kennenlernen. Vertiefen wir uns eine Weile in diese Struktur. Je besser wir sie kennen, desto einfacher ist es, Daten auszulesen.
- Das sollte euch nun etwas bekannter vorkommen. Ihr habt eben selber solche Zeilen geschrieben.

Gehen wir zurück zu dem File, den ihr eben selber bearbeitet habt. Zuerst müssen wir das File einlesen.

In [83]:
```
file = open('02 HTML Code.htm', 'r')
```

`open` ist eine eingebaute Funktion von Python. Ihr findet hier eine Zusammenstellung (https://docs.python.org/2/library/functions.html) all dieser Funktionen. Es gibt neben `r`und `w` noch die Binary modi. `rb`und `wb`. Binary heisst, dass der Computer sich die Buchstaben im Format abspeichert, der ihm am besten liegt. 0 und 1. E = 01000101, F = 01000110, ihr findet mehr hier (http://sticksandstones.kstrom.com/appen.html) (allerdings Python 2) und mehr hier (https://docs.python.org/3/tutorial/inputoutput.html).

In [84]:
```
file2 = open('testingopen.txt', 'w')
```

Oben seht ihr das Ergebnis, wenn ihr ein File schreibt. Im selben Ordner wie dieses Jupyter Notebook erscheint nun eben dieses file2. Das einfach der vollständigkeitshalber.

In [85]:
```
file.read()
```

Out[85]:
```
'<h1> Anatomie einer Website</h1>\n\n<!doctype html>\n<html lang="de">\n\n<head>\n    <title>Hier
steht mein Titel</title>\n</head>\n\n<body style="color:blue;">\n    <h2>Das ist die erste Übersch
rift</h2>\n    <p>Hier steht der Einleitungstext meiner Website. </p>\n    <ul>\n    <li>Erster Li
stenpunkt</li>\n    <li>Zweiter Listenpunkt</li>\n    <li>Dritter Listenpunkt</li>\n    <li>Vierte
r Listenpunkt</li>\n</body style="color:blue;">\n</ul>\n<body>\n<h2>Das ist die zweite Überschrift
</h2>\n<p style="color:red;">Hier steht der Einleitungstext meiner Website. </p>\n<ul>\n<li>Erster
Listenpunkt</li>\n<li>Zweiter Listenpunkt</li>\n<li>Dritter Listenpunkt</li>\n<li>Vierter Listenpu
nkt</li>\n</body>\n<body style="color:red;">\n</ul>\n<h2>Das ist die dritte Überschrift</h2>\n<p>H
ier steht der Einleitungstext meiner Website. </p>\n<ul>\n<li>Erster Listenpunkt</li>\n<li>Zweiter
Listenpunkt</li>\n<li>Dritter Listenpunkt</li>\n<li>Vierter Listenpunkt</li>\n</body>\n'
```

In [86]:
```
text = file.read()
```

In [87]:
```
file = open('02 HTML Code.htm', 'r')
```

In [88]:
```
file.readline()
```

Out[88]:
```
'<h1> Anatomie einer Website</h1>\n'
```

In [89]:
```
#Das sieht aber nicht sehr schön aus
file = open('02 HTML Code.htm', 'r')
text = file.read()
text
```

Out[89]:
```
'<h1> Anatomie einer Website</h1>\n\n<!doctype html>\n<html lang="de">\n\n<head>\n    <title>Hier
steht mein Titel</title>\n</head>\n\n<body style="color:blue;">\n    <h2>Das ist die erste Übersch
rift</h2>\n    <p>Hier steht der Einleitungstext meiner Website. </p>\n    <ul>\n    <li>Erster Li
stenpunkt</li>\n    <li>Zweiter Listenpunkt</li>\n    <li>Dritter Listenpunkt</li>\n    <li>Vierte
r Listenpunkt</li>\n</body style="color:blue;">\n</ul>\n<body>\n<h2>Das ist die zweite Überschrift
</h2>\n<p style="color:red;">Hier steht der Einleitungstext meiner Website. </p>\n<ul>\n<li>Erster
Listenpunkt</li>\n<li>Zweiter Listenpunkt</li>\n<li>Dritter Listenpunkt</li>\n<li>Vierter Listenpu
nkt</li>\n</body>\n<body style="color:red;">\n</ul>\n<h2>Das ist die dritte Überschrift</h2>\n<p>H
ier steht der Einleitungstext meiner Website. </p>\n<ul>\n<li>Erster Listenpunkt</li>\n<li>Zweiter
Listenpunkt</li>\n<li>Dritter Listenpunkt</li>\n<li>Vierter Listenpunkt</li>\n</body>\n'
```

Mit BeautifulSoup kann man HTML-Sites in eine Form bringen, die es uns Menschen wieder einfacher macht, die Inhalte zu lesen. Wir beginnne wir immer, indem wir die Library installieren. Wie mit `!pip import BS4`falls ihr das nicht schon getan habt. Und dann importieren wir die Library.

In [90]:
```
from bs4 import BeautifulSoup
```

In [91]:
```
soup = BeautifulSoup(text, 'lxml')
```

In [92]:
```
soup.find('h2')
```

Out[92]:
```
<h2>Das ist die erste Überschrift</h2>
```

In [93]:
```
soup.find_all('h2')
```

Out[93]:
```
[<h2>Das ist die erste Überschrift</h2>,
 <h2>Das ist die zweite Überschrift</h2>,
 <h2>Das ist die dritte Überschrift</h2>]
```

In [94]:
```
#Bauen wir eine neue Liste
new_lst = []
for elem in soup.find_all('h2'):
    new_lst.append(elem.text)
```

In [95]:
```
new_lst
```

Out[95]:
```
['Das ist die erste Überschrift',
 'Das ist die zweite Überschrift',
 'Das ist die dritte Überschrift']
```

In [96]:
```
#Bauen wir eine Liste von Dictionaries
new_lst = []
for elem in soup.find_all('h2'):
    mini_dict = {'Überschrift':elem.text}
    new_lst.append(mini_dict)
```

In [97]:
```
import pandas as pd
```

In [98]:
```
pd.DataFrame(new_lst)
```

Out[98]:

	Überschrift
0	Das ist die erste Überschrift
1	Das ist die zweite Überschrift
2	Das ist die dritte Überschrift

Gehen wir zu etwas komplexerem Inhalt: Ein RSS-Feed

Was wir hier haben, ist die aktuelle Ausbuchung der Parkplätze in der Stadt Zürich. Sie werden von der Stadt in einem RSS Feed abgeboten. Schauen wir uns den Link einmal an (http://www.plszh.ch/plsFeed/rss). Nicht sehr lesbar, aber äusserst strukturiert. Lesen wir die Plätze aus.

In [99]:
```
import requests
```

In [100]:
```
r = requests.get('http://www.plszh.ch/plsFeed/rss') #Besuchen wir die URL
```

In [101]:
```
contents = r.text #Wandeln wir den Text in ein Format um, mit dem BeautifulSoup umgehen kann.
soup = BeautifulSoup(contents,'xml') #Geben wir das an BeautifulSoup weiter
titles = soup.find_all('title') #Nun lesen wir Titel aus.
len(titles) #Schauen wir, wie lange die Titel sind.
```

Out[101]:
```
37
```

In [102]:
```
titles[0]
```

Out[102]:
```
<title>FEED Parkleitsystem Stadt Zürich</title>
```

In [103]:
```
titles[1]
```

Out[103]:
```
<title>Parkgarage am Central / Seilergraben</title>
```

In [104]:
```
titles[1].text
```

Out[104]:
```
'Parkgarage am Central / Seilergraben'
```

In [105]:
```
descr = soup.find_all('description')
descr[1]
```

Out[105]:
```
<description>open /   15</description>
```

In [106]:
```
#Bauen wir die eigene Liste
lst = []
for garage,b in zip(titles,descr):

    mini_dict = {'Parkgarage':garage.text,
                 'Descr':b.text}

    lst.append(mini_dict)
```

In [107]:
```
lst
```

Out[107]:

```
[{'Descr': 'http://www.plszh.ch/plsFeed/rss?type=rss_0.92 | rss_0.93 | rss_0.94 | rss_1.0 | rss_
2.0 (=default) | atom_0.3 | atom_1.0',
  'Parkgarage': 'FEED Parkleitsystem Stadt Zürich'},
 {'Descr': 'open /   15',
  'Parkgarage': 'Parkgarage am Central / Seilergraben'},
 {'Descr': 'open /  163', 'Parkgarage': 'Parkhaus Accu / Otto-Schütz-Weg'},
 {'Descr': 'open /   56',
  'Parkgarage': 'Parkhaus Albisriederplatz / Badenerstrasse 380'},
 {'Descr': 'open /   12',
  'Parkgarage': 'Parkhaus Bleicherweg / Beethovenstrasse 35'},
 {'Descr': 'open /  220',
  'Parkgarage': 'Parkhaus Center Eleven / Sophie-Täuber-Strasse 4'},
 {'Descr': 'open /   89',
  'Parkgarage': 'Parkhaus City Parking / Gessnerallee 14'},
 {'Descr': 'open /   54',
  'Parkgarage': 'Parkhaus Cityport / Affolternstrasse 56'},
 {'Descr': 'open /  277',
  'Parkgarage': 'Parkhaus Crowne Plaza / Badenerstrasse 420'},
 {'Descr': 'open /   36',
  'Parkgarage': 'Parkhaus Dorflinde / Schwamendingenstrasse 31'},
 {'Descr': 'open /   72',
  'Parkgarage': 'Parkhaus Feldegg / Riesbachstrasse 7'},
 {'Descr': 'open /    0', 'Parkgarage': 'Parkhaus Globus / Löwenstrasse 50'},
 {'Descr': 'open /  103',
  'Parkgarage': 'Parkhaus Hardau II / Bullingerstrasse 73'},
 {'Descr': 'open /   72', 'Parkgarage': 'Parkhaus Hauptbahnhof / Sihlquai 41'},
 {'Descr': 'open /  270',
  'Parkgarage': 'Parkhaus Hohe Promenade / Rämistrasse 22a'},
 {'Descr': 'open /    0',
  'Parkgarage': 'Parkhaus Jelmoli / Steinmühleplatz 1'},
 {'Descr': 'open /  109',
  'Parkgarage': 'Parkhaus Jungholz / Jungholzstrasse 19'},
 {'Descr': 'open /   56',
  'Parkgarage': 'Parkhaus Max-Bill-Platz / Armin-Bollinger-Weg'},
 {'Descr': 'open /  999',
  'Parkgarage': 'Parkhaus Messe Zürich AG / Andreasstrasse 65'},
 {'Descr': 'open /   42',
  'Parkgarage': 'Parkhaus Nordhaus / Siewerdtstrasse 8'},
 {'Descr': 'open /    1',
  'Parkgarage': 'Parkhaus Octavo / Brown-Boveri-Strasse 2'},
 {'Descr': 'open /   92', 'Parkgarage': 'Parkhaus Opéra / Schillerstrasse 5'},
 {'Descr': 'open /  334',
  'Parkgarage': 'Parkhaus P West / Förrlibuckstrasse 151'},
 {'Descr': 'open /  107',
  'Parkgarage': 'Parkhaus Park Hyatt / Beethovenstrasse 21'},
 {'Descr': 'open /    0',
  'Parkgarage': 'Parkhaus Parkside / Sophie-Täuber-Strasse 10'},
 {'Descr': 'open /  125',
  'Parkgarage': 'Parkhaus Pfingstweid / Pfingstweidstrasse 1'},
 {'Descr': 'open /  128',
  'Parkgarage': 'Parkhaus Stampfenbach / Niklausstrasse 1'},
 {'Descr': 'open /    5',
  'Parkgarage': 'Parkhaus Talgarten / Nüschelerstrasse 31'},
 {'Descr': 'open /   17',
  'Parkgarage': 'Parkhaus USZ Nord / Frauenklinikstrasse'},
 {'Descr': '???? / ???',
  'Parkgarage': 'Parkhaus Uni Irchel / Winterthurerstrasse 181'},
 {'Descr': 'open /   24', 'Parkgarage': 'Parkhaus Urania / Uraniastrasse 3'},
 {'Descr': 'open /   16', 'Parkgarage': 'Parkhaus Utoquai / Färberstrasse 6'},
 {'Descr': 'open /   26',
  'Parkgarage': 'Parkhaus Züri 11 Shopping / Nansenstrasse 5/7'},
 {'Descr': 'open /   38',
  'Parkgarage': 'Parkhaus Zürichhorn / Dufourstrasse 142'},
 {'Descr': 'open /    0',
  'Parkgarage': 'Parkplatz Eisfeld / Thurgauerstrasse 54'},
 {'Descr': 'open /  144',
  'Parkgarage': 'Parkplatz Theater 11 / Dörfli-/Thurgauerstrasse'},
 {'Descr': 'open /    1', 'Parkgarage': 'Parkplatz USZ Süd / Gloriastrasse'}]
```


In [108]:
```
pd.DataFrame(lst).head()
```

Out[108]:

	Descr	Parkgarage
0	http://www.plszh.ch/plsFeed/rss?type=rss_0.92 ...	FEED Parkleitsystem Stadt Zürich
1	open / 15	Parkgarage am Central / Seilergraben
2	open / 163	Parkhaus Accu / Otto-Schütz-Weg
3	open / 56	Parkhaus Albisriederplatz / Badenerstrasse 380
4	open / 12	Parkhaus Bleicherweg / Beethovenstrasse 35

In [109]:
```
pd.DataFrame(lst)[1:]
```

Out[109]:

	Descr	Parkgarage
1	open / 15	Parkgarage am Central / Seilergraben

1	open / 15	Parkgarage am Central / Sellergraben
2	open / 163	Parkhaus Accu / Otto-Schütz-Weg
3	open / 56	Parkhaus Albisriederplatz / Badenerstrasse 380
4	open / 12	Parkhaus Bleicherweg / Beethovenstrasse 35
5	open / 220	Parkhaus Center Eleven / Sophie-Täuber-Strasse 4
6	open / 89	Parkhaus City Parking / Gessnerallee 14
7	open / 54	Parkhaus Cityport / Affolternstrasse 56
8	open / 277	Parkhaus Crowne Plaza / Badenerstrasse 420
9	open / 36	Parkhaus Dorflinde / Schwamendingenstrasse 31
10	open / 72	Parkhaus Feldegg / Riesbachstrasse 7
11	open / 0	Parkhaus Globus / Löwenstrasse 50
12	open / 103	Parkhaus Hardau II / Bullingerstrasse 73
13	open / 72	Parkhaus Hauptbahnhof / Sihlquai 41
14	open / 270	Parkhaus Hohe Promenade / Rämistrasse 22a
15	open / 0	Parkhaus Jelmoli / Steinmühleplatz 1
16	open / 109	Parkhaus Jungholz / Jungholzstrasse 19
17	open / 56	Parkhaus Max-Bill-Platz / Armin-Bollinger-Weg
18	open / 999	Parkhaus Messe Zürich AG / Andreasstrasse 65
19	open / 42	Parkhaus Nordhaus / Siewerdtstrasse 8
20	open / 1	Parkhaus Octavo / Brown-Boveri-Strasse 2
21	open / 92	Parkhaus Opéra / Schillerstrasse 5
22	open / 334	Parkhaus P West / Förrlibuckstrasse 151
23	open / 107	Parkhaus Park Hyatt / Beethovenstrasse 21
24	open / 0	Parkhaus Parkside / Sophie-Täuber-Strasse 10
25	open / 125	Parkhaus Pfingstweid / Pfingstweidstrasse 1
26	open / 128	Parkhaus Stampfenbach / Niklausstrasse 1
27	open / 5	Parkhaus Talgarten / Nüschelerstrasse 31
28	open / 17	Parkhaus USZ Nord / Frauenklinikstrasse
29	???? / ???	Parkhaus Uni Irchel / Winterthurerstrasse 181
30	open / 24	Parkhaus Urania / Uraniastrasse 3
31	open / 16	Parkhaus Utoquai / Färberstrasse 6
32	open / 26	Parkhaus Züri 11 Shopping / Nansenstrasse 5/7
33	open / 38	Parkhaus Zürichhorn / Dufourstrasse 142
34	open / 0	Parkplatz Eisfeld / Thurgauerstrasse 54
35	open / 144	Parkplatz Theater 11 / Dörfli-/Thurgauerstrasse
36	open / 1	Parkplatz USZ Süd / Gloriastrasse

Gehen wir zur Übung 3 (https://github.com/MAZ-CAS-DDJ/kurs_18_19/blob/master/06%20APIs%2C%20Scraping%20I/%C3%9Cbun

Komplexere Websites

In [110]:
```
#laden wir die Frontpage von Watson
r = requests.get('https://www.watson.ch')
contents = r.text #Wir lesen den Inhalt aus
soup = BeautifulSoup(contents,'xml')
```

In [111]:
```
#Ziehen wir die Titel raus
titelliste = soup.find_all('a')
```

In [112]:
```
len(titelliste)
```

Out[112]: 174

In [113]:
```
#Ziehen wir alle Kommentare raus
kommentare = soup.find_all('a', {'class':'standard comments'})
```

In [114]:
```
len(kommentare)
```

Out[114]: 17

Hier sehen wir, dass also nicht immer dieselbe Anzahl Kommentare und Artikel herausgelesen werden kann. Nicht jeder Artikel hat einen Kommentar. Wir müssen deshalb unseren Code anpassen.

In [115]:
```
storybox = soup.find_all('div', {'class':'text'})
```

In [116]:
```
len(storybox)
```

Out[116]: 24

In [117]:
```
storybox[0]
```

Out[117]:
```
<div class="text">
    Wir verwenden Cookies und Analysetools, um die Nutzerfreundlichkeit der Internetseite zu verb
essern und passende Werbung von watson und unseren Werbepartnern anzuzeigen.
    Weitere Infos findest Du in unserer <a href="/u/agb">Datenschutzerklärung</a>.
    </div>
```

In [118]:
```
storybox[2]
```

Out[118]:
```
<div class="text">
<h2><a href="/sport/fussball/780028292-die-nati-in-der-verjuengungskur-so-koennte-das-kader-bei-d
er-em-2020-aussehen">Die Nati in der Verjüngungskur – so könnte das Kader bei der EM 2020 aussehe
n</a></h2>
<p class="author">von Adrian Bürgler</p>
<!-- iconrow ==> -->
<div class="iconrow">
<!-- media icons ==> -->
<div class="media_icons">
<!-- comments ==> -->
<a class="standard comments" href="#comments">
<span>26</span>
</a>
<!-- <== comments -->
<!-- social shares ==> -->
<a class="standard shares" href="#">
<span>19</span>
</a>
<!-- <== social shares -->
</div>
<!-- <== media icons -->
</div>
<!-- <== iconrow -->
</div>
```

In [119]:
```
storybox[2].find('h2').text
```

Out[119]: 'Die Nati in der Verjüngungskur – so könnte das Kader bei der EM 2020 aussehen'

In [120]:
```
storybox[2].find('a', {'class':'standard comments'}).text.replace("\n", "")
```

Out[120]: '26'

In [121]:
```
lst = []
for elem in storybox:
    t = elem.find('h2').text
    k = elem.find('a', {'class':'standard comments'}).text.replace("\n", "")

    mini_dict = {'Titel': t,
                 'Kommentar': k}

    lst.append(mini_dict)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-121-6b4fcf56bbb5> in <module>()
      1 lst = []
      2 for elem in storybox:
----> 3     t = elem.find('h2').text
      4     k = elem.find('a', {'class':'standard comments'}).text.replace("\n", "")
      5

AttributeError: 'NoneType' object has no attribute 'text'
```

In [122]:
```
lst = []
for elem in storybox:

    try: #Hier kannn man mit Fehlern umgehen. Man muss allerdings spärlcih damit umgehen.
        t = elem.find('h2').text
    except:
        t = 'Kein Titel'
```

```
    try:
        k = elem.find('a', {'class':'standard comments'}).text.replace("\n", "")
    except:
        k = 'Keine Kommentare'

    mini_dict = {'Titel': t,
                 'Kommentar': k}

    lst.append(mini_dict)
```

In [123]:
```
pd.DataFrame(lst)
```

Out[123]:

	Kommentar	Titel
0	Keine Kommentare	Kein Titel
1	6	Rick Gates ist die Achillesferse von Donald Trump
2	26	Die Nati in der Verjüngungskur – so könnte das...
3	Keine Kommentare	Nach dem Absturz: Ju-Air will noch im August w...
4	19	Gelson Fernandes tritt aus der Nati zurück: «E...
5	19	12 Salate, die du diesen Sommer über essen sol...
6	25	Das Tessin wollte die Burkas verbannen – bestr...
7	Keine Kommentare	A2 gesperrt: So sieht das Lastwagen-Wrack im P...
8	410	United lehnt zwei Spieler plus 50 Millionen fü...
9	46	«In der Nati ist keiner so klug wie du» – offe...
10	10	18 lustige Fails, die dich alles um dich herum...
11	7	\nWegen diesen fiesen Memes? «Pu der Bär» komm...
12	9	\nDiese Handys erhalten (ab sofort) das neue A...
13	10	\nKind vergewaltigt, erniedrigt, verkauft: Zwö...
14	Keine Kommentare	\nDrohne vermiest die Ferien des französischen...
15	Keine Kommentare	\nGewaltiger Erdrutsch am Mont-Blanc fordert z...
16	1	\nDie Favoritin hält dem Druck stand: Jolanda ...
17	11	20 Zitate von wichtigen Menschen, die du fürs ...
18	5	12 Grad kälter in 20 Minuten – nach Gewittern ...
19	62	Leonardo Genoni verlässt den SCB – na und?
20	52	Die Erde ist bei Google Maps keine flache Sche...
21	26	Petkovic provoziert den Neustart – was der ges...
22	Keine Kommentare	Milliardär Kroenke will alleiniger Besitzer vo...
23	Keine Kommentare	Im Bild

Als Übung: Mit Funktion dasselbe tun.

Wenn man sehr ausführlichen Code entwickelt, ist es oft praktisch, mit Funktionen zu arbeiten. Hier ein Beispiel davon, die das aussehen könnte.

In [124]:
```
def kommentarezählen(url):

    r = requests.get(url)
    soup = BeautifulSoup(r.text,'xml')
    storybox = soup.find_all('div', {'class':'text'})

    lst = []
    for elem in storybox:
        try: #Hier kannn man mit Fehlern umgehen. Man muss allerdings spärlcih damit umgehen.
            t = elem.find('h2').text
        except:
            t = 'Kein Titel'
        try:
            k = elem.find('a', {'class':'standard comments'}).text.replace("\n", "")
        except:
            k = 'Keine Kommentare'
        mini_dict = {'Titel': t,
                 'Kommentar': k}
        lst.append(mini_dict)

    return pd.DataFrame(lst)
```

In [125]:
```
kommentarezählen('https://www.watson.ch/')
```

Out[125]:

	Kommentar	Titel
0	Keine Kommentare	Kein Titel
1	6	Rick Gates ist die Achillesferse von Donald Trump
2	26	Die Nati in der Verjüngungskur – so könnte das...
3	Keine Kommentare	Nach dem Absturz: Ju-Air will noch im August w...
4	19	Gelson Fernandes tritt aus der Nati zurück: «E...
5	19	12 Salate, die du diesen Sommer über essen sol...
6	25	Das Tessin wollte die Burkas verbannen – bestr...
7	Keine Kommentare	A2 gesperrt: So sieht das Lastwagen-Wrack im P...
8	410	United lehnt zwei Spieler plus 50 Millionen fü...
9	46	«In der Nati ist keiner so klug wie du» – offe...
10	10	18 lustige Fails, die dich alles um dich herum...
11	7	\nWegen diesen fiesen Memes? «Pu der Bär» komm...
12	9	\nDiese Handys erhalten (ab sofort) das neue A...
13	10	\nKind vergewaltigt, erniedrigt, verkauft: Zwö...
14	Keine Kommentare	\nDrohne vermiest die Ferien des französischen...
15	Keine Kommentare	\nGewaltiger Erdrutsch am Mont-Blanc fordert z...
16	1	\nDie Favoritin hält dem Druck stand: Jolanda ...
17	11	20 Zitate von wichtigen Menschen, die du fürs ...
18	5	12 Grad kälter in 20 Minuten – nach Gewittern ...
19	62	Leonardo Genoni verlässt den SCB – na und?
20	52	Die Erde ist bei Google Maps keine flache Sche...
21	26	Petkovic provoziert den Neustart – was der ges...
22	Keine Kommentare	Milliardär Kroenke will alleiniger Besitzer vo...
23	Keine Kommentare	Im Bild

Das können wir natürlich auf der Kommandozeile erledigen. Speichern wir unseren Skript und wenden wir ihn an. Wir müssen noch dafür sorgen