block5-hsozkult

September 24, 2018

1 Web Scraping von Hsozkult

Das Ziel besteht darin, eine Suchanfrage an Hsozkult zu schicken und die erste Rezension, die wir erhalten, in einer Text-Datei abzuspeichern.

Als erstes binden wir die benötigten Bibliotheken ein:

```
In [1]: import re
        from urllib.request import urlopen
        from bs4 import BeautifulSoup
```

`re` stellt Funktionen für sogenannte 'Reguläre Ausdrücke' zur Verfügung. `urllib` ist eine Standardbibliothek für den Umgang mit dem Internetprotokoll `http`. `bs4` steht für *BeautifulSoup* – die beste Bibliothek, um HTML zu parsen.

AnschlieSSend bauen wir uns eine URL zusammen: Sie besteht aus der eigentlichen Domain, aus dem Query-String, über den sich Suchanfragen ausführen lassen, und aus den Suchbegriffen.

```
In [2]: base_url = "http://www.hsozkult.de"
        query_string = "/searching/page?q="
        search_string = "rezension grundwissenschaft"
        search_string = search_string.replace(" ", "+")
        url = base_url + query_string + search_string
```

Da keine Leerzeichen in einer URL vorhanden sein dürfen, müssen wir im Such-String alle Leerzeichen durch '+' ersetzen. Danach wird die URL aus den einzelnen Bestandteilen zusammengesetzt.

Als nächstes senden wir mithilfe von `urlopen` eine Anfrage an Hsozkult. Die Antwort des Servers übergeben wir an *BeautifulSoup*, lassen es durch einen HTML-Parser verarbeiten und speichern das Ergebnis in der Variable `bs` ab.

```
In [3]: search = urlopen(url)
        bs = BeautifulSoup(search, 'html.parser')
```

Daraufhin müssen wir einen Blick auf eine entsprechende Seite von Hsozkult werfen. Dabei stellen wir fest, dass die Tabelle mit den einzelnen Suchergebnissen sich in einem `div`-Kontainer der `class` "hfn-list-itemtitle" befindet. Aus der Ergebnis-Liste nehmen wir uns den ersten Treffern vor.

```
In [4]: results = bs.find_all('div', {'class': 'hfn-list-itemtitle'})

        first_hit = results[0].find('a')['href']
        first_hit

Out[4]: '/searching/id/rezbuecher-28479?title=t-kahlert-unternehmungen-grossen-stils&q=rezensi
```

Wir suchen in diesem ersten Treffer nach einem Link <a> mit dem Attribut `href`, lesen das aus und speichern es in der Variable `first_hit` ab. Damit wissen wir, hinter welchem Link sich die erste gefundene Rezension verbirgt.

Wir bauen anschlieSSend erneut eine URL zusammen, diesmal aus der `base_url` und unserem `first_hit`. Da sich darin aber ein Leerzeichen befindet, müssen wir es wieder durch den entsprechenden Escape-Code '%20' ersetzen.

Haben wir das erledigt, schicken wir erneute eine Anfrage an Hsozkult, diesmal für die Rezension, und parsen das Ganze mit *BeautifulSoup*.

```
In [5]: link_url = base_url + first_hit
        link_url = link_url.replace(" ", "%20")

        review_html = urlopen(link_url)
        bs_review = BeautifulSoup(review_html, 'html.parser')
```

Unser Ziel ist es, den eigentlichen Text der Rezension zusammen mit den dazugehörigen bibliographischen Metadaten in einer Datei zu speichern. Um mit *BeautifulSoup* auf die einschlägigen HTML-Tags zugreifen zu können, werfen wir einen Blick auf den Quelltext der Seite. Auf diese Weise stellen wir fest, dass sich alle bibliographischen Metadaten in einer Tabelle jeweils in einem `div` mit dem Klassen-Attribut `class="hfn-item-metarow"` befinden. Eine Spalte gibt Auskunft darüber, um welchen Datentyp es sich handelt, also bspw. 'Autor(en)', 'Titel', 'Erschienen' etc. Die andere Spalte enthält die entsprechenden Angaben. Es bietet sich daher an, anhand dieser Angaben ein *Dictionary* zu erstellen.

Dafür lesen wir jede Zeile der Tabelle aus und übergeben die Werte einem *Dictionary* namens `meta_data`. Wir interessieren uns dabei nur für die ausgegebenen Texte innerhalb der *Tags*. Und wir entfernen den unnötigen *White space*. Wenn eine Spalte keine Angaben enthält, übernehmen wir sie auch nicht ins *Dictionary*.

```
In [6]: meta_data = {}

        for key, val in bs_review.find_all('div', {'class': 'hfn-item-metarow'}):
            k = key.get_text()
            k = k.strip()
            v = val.get_text()
            v = v.strip()
            v = re.sub(r"[\n\t]+", " ", v)
            if k != '':
                meta_data[k] = v

        meta_data
```

```
Out[6]: {'Autor(en)': 'Kahlert, Torsten',
         'Erschienen': 'Berlin  2017: be.bra Verlag',
         'ISBN': '978-3-95410-089-7',
         'Preis': ' 40,00',
         'Titel': 'Unternehmungen groen Stils.  Wissenschaftsorganisation, Objektivität und His
         'Umfang': '384 S.'}
```

Nun kümmern wir uns um den Rezensenten. Die Angaben dazu verstecken sich im `div` mit der Klasse `hfn-item-creator`. Auch hier interessieren wir uns nur für den Text. Das Institut etc. spielt für uns aber keine Rolle. Wir zerlegen also den String `review_author` an den Kommas und greifen nur auf den ersten Abschnitt der geschaffenen Liste zurück.

```
In [7]: review_author = bs_review.find('div', {'class': 'hfn-item-creator'})
        review_author = review_author.get_text()
        review_author = review_author.strip().split(',')[0]

        review_author

Out[7]: 'Jan Ruhkopf'
```

Jetzt lesen wir den eigentlichen Text der Besprechung aus. In dem `div` mit der Klasse `hfn-item-fulltext` befinden sich gleich mehrere Paragraphen. Diese suchen wir mit zwei `find`- bzw. `find_all`-Methoden des *BeautifulSoup*-Objekts. Das Ergebnis ist eine Liste mit den einzelnen Paragraphen, die wir in der Variable `review` abspeichern. Danach nutzen wir eine besondere Python-Konstruktion, die *list comprehension*. Mit ihrer Hilfe erstellen wir eine neue Liste `review_content`, in der der Text der einzelnen Paragraphen-*tags* hinterlegt wird.

```
In [8]: review = bs_review.find('div', {'class': 'hfn-item-fulltext'}).find_all('p')

        review_content = [paragraph.get_text() for paragraph in review]
```

Zu guter letzt speichern wir das alles in der Datei `rezension.txt` ab. Wir öffen es mit der Funktion `with open(...) as x:`, die sicherstellt, dass das *File*-Objekt am Ende auch wieder geschlossen wird.

In dem Block unterscheiden wir, ob es sich um einen Sammelband handelt oder um eine Monographie. Dementsprechend formatieren wir den String, den wir durch Interpolation aus den Metadaten und `review_author` zusammensetzen. Danach kommen zwei *newlines* und der Rezensionstext.

```
In [9]: with open('rezension.txt', 'w', encoding='utf-8') as f:
            if 'Hrsg. v.' in meta_data.keys():
                f.write("{}: Rezension von: {} (Hg.): {}, {}.".format(
                    review_author,
                    meta_data['Hrsg. v.'],
                    meta_data['Titel'],
                    meta_data['Erschienen']))
            else:
                f.write("{}: Rezension von: {}: {}, {}.".format(
                    review_author,
```

```
                meta_data['Autor(en)'],
                meta_data['Titel'],
                meta_data['Erschienen']))
        f.write("\n\n")
        f.write("\n".join(review_content))
```

Fertig ist unser Hsozkult-Scraper!