

# Webscrapping

Jan Popko

Python Advanced

# Requests

Modul zum einfachen laden von html-Code einer Seite

```
pip install requests
```

```
# HTML-Code laden
```

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

```
# das Argument .text gibt das HTML als text wieder  
print(r.text)
```

```
# die Methode .json() gibt ein JSON zurück  
# JSON = JavaScript Object Notation  
print(r.json())
```

Jan Popko

Python Advanced

# BeautifulSoup

Modul zum extrahieren von Daten aus HTML und XML Dateien

BeautifulSoup macht das Extrahieren von Daten aus dem HTML-Code sehr einfach

```
pip install beautifulsoup4
```

```
pip install beautifulsoup → gibt die veraltete Version!
```

Jan Popko

Python Advanced

# BeautifulSoup

Starten eines Webscraping Programmes:

```
import requests
from bs4 import BeautifulSoup

# requests.get(url) zieht den Code von der Seite
r = requests.get('https://www.webscraper.io/test-sites/tables')

# das BeautifulSoup Objekt übernimmt den Text von r
soup = BeautifulSoup(r.text, 'html.parser')

# prettify() gibt den Quellcode lesbar aus
print(soup.prettify())
```

Jan Popko

Python Advanced

# Nützliche Methoden und Attribute

`soup.tag` – gibt „`<tag> ... </tag>`“

`soup.tag.name` – gibt bezeichnung des Tags wieder

`soup.tag.string` – gibt den Inhalt von `<tag> Inhalt </tag>`

`soup.tag.parent.name` – gibt den Namen des übergeordneten Tags

`soup.tag.contents` – gibt eine Liste mit allen Children des Tags

`soup.find('tag')` – findet den ersten tag im Code

`soup.findall('tag')` – gibt Liste mit allen Tags

`soup.find_all('tag')` – gibt Liste mit allen Tags

`soup.find_next('tag')` – findet das nächste Tag

`soup.get_text()` – extrahiert den ganzen Text der Seite