Index

A

Amazon reviews, 198
  browser’s developer tools, 241–242
  dataset library, 245–248
  “json” module, 243, 245
  Learning Python by Mark Lutz, 241
  matplotlib library, 250
  nltk, 249
  POST requests, 241, 243
  Python shell, 249
  sentiment plots, rating level, 251
  text editor, 243
  vaderSentiment library, 248–249
Application Programming Interface (APIs), 4–5, 99, 180, 193, 197, 201
Associated Press (AP) vs. Meltwater, 182

B

Barclays’ mortgage simulator, 214–215
Beautiful Soup, 67, 187–188, 190, 193
  attrs argument, 64
  code snippets, 71
  Comment objects, 76
  complex selectors, 75
  CSS selectors, 75
  documentation, 77
  elements, 74
  episode tables, 70
find and find_all, 65–66, 72
Game of Thrones, 68, 75
HTML pages, 61
HTML tree, 73–74
installing, 61
**keywords, 65
limit argument, 64
lists, and regular expressions, 73
methods, 63
name argument, 64
NavigableString objects, 76
parsers, 62–63
recursive argument, 64
string argument, 62, 64
tree-based representation, 63
warning, 62
Board members, 199, 278–281
Books to scrape, 197, 206–208

C

CacheControl, 188
Caching, 188–189
Cascading Style Sheets (CSS)
  Chrome’s Developer Tools, 60
  formatting, 57
  HTML attributes, 56–57
  selectors, 58–61
  style declarations, 57–58
  Wikipedia page, 60

© Seppe vanden Broucke and Bart Baesens 2018
S. vanden Broucke and B. Baesens, Practical Web Scraping for Data Science,
https://doi.org/10.1007/978-1-4842-3582-9
INDEX

Cloud scraping, 195
Clustering, fashion products, 198, 237–238, 240
Command-line tools, 191
Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA), 194, 199
contours, separate images, 285–291
drawContours method, 285
generated_images, 282–283
Keras configuration file, 292–293
Keras library, 291
kernel sizes, 285
letters directory, 291
neural architecture, 296–298
OpenCV, 281, 284–285
Python scripts, 282
soil erosion, 285
thresholding, opening, and contour detection, 284
train.py, 294–295
Computer Fraud and Abuse Act (CFAA), 181–182, 184–185
Cookie hijacking, 112
Craigslist vs. Naturemarket, 182
Cross-Industry Standard Process for Data Mining (CRISP-DM)
“Business Understanding”, 177
clean data, 178
data analysis, 178
data sources, 178
description, 176
interpret, evaluate, and deploy, 178
KDD process, 177
select data, 178
transformations, 178
cURL, 191

D, E

Data science
analysts and modelers, 175
big data platform experts, 176
business integrators, sponsors, and champions, 176
business intelligence experts, 176
CRISP-DM (see Cross-Industry Standard Process for Data Mining (CRISP-DM))
database administrators, 175
data wranglers, 175
fundamental theorists, mathematicians and statisticians, 175
hacker profiles, 176
IT architects, 176
management, 176
stakeholders, 176
web scraping
APIs, 180
data sources, 179
managerial warnings, 180
model train vs. model run gap, 179
production setting, 179
products, 180
scraped data, 180
websites, 179

Digital Millennium Copyright Act (DMCA), 185

F

Facebook vs. Power Ventures, 184
Form Data, 85
G
Gephi’s visualization, 281
GitHub, 197, 209–214
Global interpreter lock (GIL), 274
Graphical scraping tools, 191–192

H
Hacker News API, 197, 201
HTTPie, 191
Hypertext Markup Language (HTML)
  Google Chrome
    Developer Tools, 52–53
    Elements tab, 55–56
    Network tab, 53–54
    Preserve log, 54
    view source, 51–52
  Wikipedia, 51
programming language, 50
Python library, 50
tags, 50–51
web browser, 50
Wikipedia, 49
HyperText Transfer Protocol (HTTP)
  “Accept-Encoding”, 32
  binary, 121–126
  blank <CR><LF> line, 34
  CONNECT and OPTIONS method, 98
  “Content-Type”, 34
cookies
  “action” attribute, 116
  browser’s developer tools, 114
drawbacks, 109
ever cookies, 113
fingerprinting, 113
functionality, 109
GET request, 97, 117–118
multi-valued headers, 112
networking protocol, 108
PHPSESSID, 115
request-reply-based approach, 109
secret code, 118–119
“Set-Cookie” headers, 110–112, 116
DELETE method, 98
forms and POST requests
  “action” attribute, 87
  Chrome Developer Tools, 86
  “comments” field, 90
  “Content-Type” request, 95
data argument, 88, 96
  “<form>” tag, 93, 94
  “<input>” tag, 82
methods, 85, 88
protection field, 91, 92
Python dictionary, 89
request data, 95–96
URL parameters, 81, 84
web browser, 83
web form, 82–83
web server, 81, 87
websites, 81
forms of content, 121–126
“GET / HTTP/1.1”, 30–31
headers
  allow_redirects argument, 106
auth argument, 108
authentication mechanism, 107–108
browser’s developer tools, 102
categories, 103–104
destination.php, 105
“Preserve log” option, 105
“Referer”, 102
request library, 100
“User-Agent”, 100–101
HyperText Transfer Protocol (HTTP) (cont.)
  HEAD method, 98
  JSON, 121–126
  POST request, 98, 99
  PUT method, 98
  Python (see Python, requests library)
  request headers, 31
  request message, 28–30, 33
  request-reply messages, 29
  sessions mechanism, 119–120
  status codes, 33
  TRACE and PATCH method, 98
  “User-Agent” header, 32, 120–121

I
  IATA airline information, 198, 222–228
  Internet Movie Database (IMDB), 198, 220–222
  Internet Service Provider (ISP), 26

J, K
  JavaScript
    client-side web browsers, 127
    cookies, 131
    definition, 127
    fragment, 128–129
    “$()” function, 129
    “getJSON” function, 129
    “jsenabled” cookie, 129
    network requests, 131
    nonce and PHPSESSID cookies, 131–133
    “quotes.php”, 130
    requests and Beautiful Soup, 129, 133
    reverse engineering, 130
    “<script>” tags, 128, 130
    source code, 128

L
  Latent Dirichlet Allocation (LDA)
    model, 266, 268–270
  Legal concerns
    Associated Press (AP) vs. Meltwater, 182
    CFAA, 181–182
copyright protection, Google’s book, 183
  Craigslist vs. Naturemarket, 182
  European Union (EU)
    Computer Misuse Act and trespass to chattels, 186
    EU Database Directive of 1996, 185
  Facebook, 182
  Facebook vs. Power Ventures, 184
  Google News site, 183
  hiQ Labs, 181
  LinkedIn and Microsoft executives, 181
  LinkedIn vs. Doe Defendants, 184
  potential customers, 181
  projects, principles, 186
  QVC, 182
  Ryanair Ltd vs. PR Aviation BV, 183
  Ticketmaster vs. Riedel Marketing Group (RMG), 182
  United States
    breach of terms and conditions, 184
    CFAA, 185
copyright/trademark infringement, 184
    DMCA, CAN-SPAM Act, 185
robots exclusion protocol, 185
trespass to chattels, 185
  LinkedIn vs. Doe Defendants, 184
M
Mortgage rates, 198, 214–220

N
Networking
  communications, 28
  OSI model, layers, 28
  set up connections, 25
www.google.com
  DNS servers, 26
    establish connection to
      172.217.17.68, 26
  HTTP request, 27
  IP address, 25–26
  operating system, 26
  TCP, 27
  textual content, HTML, 27
News articles, 198
  browser’s console, 253, 261
  Chrome, 254–255, 262
  “Content-Security-Policy”, 262
  dataset library, 264–265
  DB Browser for SQLite, 266
  execute_script method, 256, 258–259
  extracting, article’s information, 256
  for loops, 261
  GitHub, 255
  LDA model, 266, 268–270
  links, 253
  loading contents, local file, 262
  Mozilla’s documentation, 255
  Mozilla’s Readability implementation,
    JavaScript, 253
  nltk, stop-words, and gensim libraries,
    266
  readability-script, 259–261
  “<script>” tag, 254
  “Top Stories” from Google News,
    252–253

O
Object Relational Mapping (ORM), 163
Open Systems Interconnection (OSI)
  model, 28

P
Parsehub, 191
PhantomJS, 190, 195
Portia, 191
Programming languages, 190
Proxy servers, 189
Puppeteer, 190
Python
  angled brackets (“> > >”), 10
  arguments, 21–22
  Boolean logic values, 11
  control flow, 18
  dictionaries, 17
  equality and inequality operators
    (“==” and “!=”), 12
  for-loops, 19
  “if”-blocks, 18
  interpreter REPL, 9
  iterables, 20
  lists, 14
  modulo operator, 10
  number types, 11
  print function, 13
  range function, 19–20
  resources, 23
  running, source file, 23
  sets, 16–17
INDEX

Python (cont.)
  source code in “.py” files, 9
  strings, 11, 14
  tuples, 16
  variables, 13
  while looping, 20
Python, requests library, 187
  “Accept-Encoding” header, 39
  aiohttp, 35
  create, Python file, 36
  errors/warnings, 36
  grequests, 35
  headers attribute, 38
  “HTTP GET” request, 37
  httplib, 35
  HTTP status code, 38
  installing requests, 36
  networking functionality, 34
  request.Response object, 39
  requests, 35
  requests.get method, 37
  r.text, 37
  urllib, 35
  urllib3, 35
  “User-Agent”, 39
requests.get method, 44–45
slash (“/”), 43
“url-lib.parse” functions, 43
web servers, 40
Wikipedia, 46–47
Quotes to scrape, 197, 202–205

R
Reuters, 278–279
Ryanair Ltd vs. PR Aviation BV, 183

S
Scraping Hacker News, 197, 199–200
Scrapy, 188
Screen scraping, 3
Selenium
  ActionChains object, 151–153
  attributes, 150
  Chrome window, 135–136
  conditions, 139–141
  cookies, 148
  custom wait condition, 144
  definition, 134
  documentation of library, 153
  download WebDriver, 135
  execute_script method, 144
  HTML elements, 136–137
  implicit and explicit waits, 138–139
  JavaScript commands and actions, 146
  special Keys helper object, 151
  locator argument, 142
  locator tuples, 143
  matching elements, 142
  methods and attributes, 148–149
  navigation, 148
  Python console, 145

Q
Query strings, URL parameters
  “a,” “b,” and “op”, 46
  characters, 40
  code example, 39
  conventions, 40
  empty and ordered parameters, 45
  fragment identifier, 47
  HTML-formatted text, 47
  quality, requests’ documentation, 48
  requests, 41–42, 44

Python console, 145
quote element, 138
real-life projects, 154
requests and Beautiful Soup, 135
selection of actions, 143
WebDrivers, 134
WebDriverWait object, 142
WebElement objects, 148, 150
web page, actions, 146–147
XPath expressions and rules, 137
siteTheme, 111
SlimerJS, 190
SQLite database, 202, 205

T, U, V
Third-party cookies, 113
Ticketmaster vs. Riedel Marketing Group (RMG), 182
Transmission Control Protocol (TCP), 26

W, X, Y, Z
Web crawling
database, 156
data gathering, 156
description, 155
keywords, 156
legal concerns, 157
parallel programming, 157
product pages, online shop, 155
in Python, 158–161
queue, 157
retry/abort, 157
separate from scraping, 156
single website, 156
stop early, 157
storing progress, SQLite database
crawled images and web pages, 171
crawled results, 164
HTML contents, 167–171
NetworkX library, 167
ORM, 163
queue of links, 161–162
records library, 161, 163
urldefrag function, 167
Wikipedia, 164, 166
Web forum interactions, 198, 228–236
Web scraping
banks and financial institutions, 7
best practices
API, 193
Beautiful Soup, 193
CAPTCHA, 194
cat-and-mouse game, 195
check your browser, 193
cloud scraping, 195
crawling, 194
HTTP requests, 193
JavaScript engine, 194
managerial and legal concerns, 195
user agent and referrer, 193
warnings, 194
web servers, 193
Bitcoin, 8
data science
APIs, 4–5
opportunities, 4
raw material, 4
Web browsers, 4
definition, 3
digital marketeers and digital artists, 6
emmanuel sales, 6
fashion products, 8
Google's products, 5
Greek wines, 7
HR and employee analytics, 6
INDEX

Internet, 3
job sites, 8
legal concerns (see Legal concerns)
Lego brick information, 7
messages, social media, 6
online price information, 6
real-life use cases, 5
setting up
command line, 9
install Python 3, 8
text editors, 9
sociopolitical scientists, 7
Tinder and Instagram, 7
Wikipedia graph, 198

client-server database
   system, 274
crawling setup, 271–273
database, table pages and links, 273
dataset library, 271
get_random_unvisited_pages
   method, 273
get_title_and_links method, 273–274
“if __name__ == '__main__':”, 274
iGraph, 275
joblib library, 274
multithreaded approach, 274
NetworkX, 275–278
should_visit method, 273