This section does not constitute legal advice

Please note that the information provided on this page is for information purposes only and does not constitute professional legal advice on the practice of web scraping.
If you are concerned about the legal implications of using web scraping on a project you are working on, it is probably a good idea to seek advice from a professional, preferably someone who has knowledge of the intellectual property (copyright) legislation in effect in your country.
Now that we have seen several different ways to scrape data from websites and are ready to start working on potentially larger projects, we may ask ourselves whether there are any legal implications of writing a piece of computer code that downloads information from the Internet.
In this section, we will be discussing some of the issues to be aware of when scraping websites, and we will establish a code of conduct (below) to guide our web scraping projects.

Don't break the web: Denial of Service attacks

The first and most important thing to be careful about when writing a web scraper is that it typically involves querying a website repeatedly and accessing a potentially large number of pages. For each of these pages, a request will be sent to the web server that is hosting the site, and the server will have to process the request and send a response back to the computer that is running our code. Each of these requests will consume resources on the server, during which it will not be doing something else, like for example responding to someone else trying to access the same site.
If we send too many such requests over a short span of time, we can prevent other “normal” users from accessing the site during that time, or even cause the server to run out of resources and crash.
In fact, this is such an efficient way to disrupt a web site that hackers are often doing it on purpose. This is called a Denial of Service (DoS) attack.
Since DoS attacks are unfortunately a common occurence on the Internet, modern web servers include measures to ward off such illegitimate use of their resources. They are watchful for large amounts of requests appearing to come from a single computer or IP address, and their first line of defense often involves refusing any further requests coming from this IP address.
A web scraper, even one with legitimate purposes and no intent to bring a website down, can exhibit similar behaviour and, if we are not careful, result in our computer being banned from accessing a website.

Copyright: respecting other's intellectual property

It is important to recognize that in certain circumstances web scraping can be illegal, and this differs from country to country.
If the terms and conditions of the web site we are scraping specifically prohibit downloading and copying its content, then we could be in trouble for scraping it. In practice, however, web scraping is a tolerated practice, provided reasonable care is taken not to disrupt the “regular” use of a web site, as we have seen above. However you must be aware that without permisson from the copyright owner you may be in breach of copyright law.

In a sense, web scraping is no different than using a web browser to visit a web page, in that it amounts to using computer software (a browser vs a scraper) to acccess data that is publicly available on the web. However, researchers should be aware of the risk since the law views web browsing differently to automated web scraping.
In general, if data is publicly available (the content that is being scraped is not behind a password-protected authentication system), then it may be OK to scrape it, provided we don't break the web site doing so. What is potentially problematic is if the scraped data will be shared further. For example, downloading content off one website and posting it on another website (as our own), unless explicitly permitted, may constitute a violation of copyright law.
Copyright law in some countries recognises "fair use" (USA, Israel, Singapore and South Korea) or "fair dealing" (Australia) which may, under very specific circumstances, allow reusing some copyrighted material. However the scope of these exceptions is narrow, and moreso for "fair dealing" under Australian law; you should not assume they apply to your case.
For an interesting (Australian) copyright case involving web scraping, see IceTV vs Channel Nine.

Better be safe than sorry

Be aware that copyright and data privacy legislation typically differs from country to country. Be sure to check the laws that apply in your context. For example, in Australia, it can be illegal to scrape and store personal information such as names, phone numbers and email addresses, even if they are publicly available.
If you are looking to scrape data for your own personal use, then the above guidelines should probably be all that you need to worry about. However, if you plan to start harvesting a large amount of data for research or commercial purposes, you should probably seek legal advice first.
If you work in a university, chances are it has a copyright office that will help you sort out the legal aspects of your project. The university library is often the best place to start looking for help on copyright related queries.

Challenge

- What are the contact details for the copyright office (or similar) at your organisation ?

Be nice: ask and share

Depending on the scope of your project, it might be worthwhile to consider asking the owners or curators of the data you are planning to scrape if they have it already available in a structured format that could suit your project. If your aim is do use their data for research, or to use it in a way that could potentially interest them, not only it could save you the trouble of writing a web scraper, but it could also help clarify straight away what you can and cannot do with the data.
On the other hand, when you are publishing your own data, as part of a research project, documentation or a public website, you might want to think about whether someone might be interested in getting your data for their own project. If you can, try to provide others with a way to download your raw data in a structured format, and thus save them the trouble to try and scrape your own pages!

Web scraping code of conduct

This all being said, if you adhere to the following simple rules, you will probably be fine.

1. Ask nicely. If your project requires data from a particular organisation, for example, you can try asking them directly if they could provide you what you are looking for. With some luck, they will have the primary data that they used on their website in a structured format, saving you the trouble.
2. Don't download copies of documents that are clearly not public. For example, academic journal publishers often have very strict rules about what you can and what you cannot do with their databases. Mass downloading article PDFs is probably prohibited and can put you (or at the very least your friendly university librarian) in trouble. If your project requires local copies of documents (e.g. for text mining projects), special agreements can be reached with the publisher. The library is a good place to start investigating something like that.
3. Check your local legislation. For example, certain countries have laws protecting personal information such as email addresses and phone numbers. Scraping such information, even from publicly available web sites, can be illegal (e.g. in Australia).
4. Don't share downloaded content illegally. Scraping for personal purposes is usually OK, even if it is copyrighted information, as it could fall under the fair use provision of the intellectual property legislation. However, sharing data for which you don’t hold the right to share is illegal.
5. Share what you can. If the data you scraped is in the public domain or you got permission to share it, then put it out there for other people to reuse it (e.g. on datahub.io). If you wrote a web scraper to access it, share its code (e.g. on GitHub) so that others can benefit from it.
6. Don't break the Internet. Not all web sites are designed to withstand thousands of requests per second. If you are writing a recursive scraper (i.e. that follows hyperlinks), test it on a smaller dataset first to make sure it does what it is supposed to do. Adjust the settings of your scraper to allow for a delay between requests. By default, Scrapy uses conservative settings that should minimize this risk.
7. Publish your own data in a reusable way. Don’t force others to write their own scrapers to get at your data. Use open and software-agnostic formats (e.g. JSON, XML), provide metadata (data about your data: where it came from, what it represents, how to use it, etc.) and make sure it can be indexed by search engines so that people can find it.
8. View robots.txt file. Robots.txt is a file used by websites to let 'bots' know if or how the site should be crawled and indexed. When you are trying to extract data from the web, it is critical to understand what robots.txt is and how to respect it to avoid legal ramifications. This file can be accessed for any domain by accessing <domain_url>/robots.txt. For eg: monash.edu/robots.txt, facebook.com/robots.txt, linkedin.com/robots.txt.

Happy scraping!

References

- The Web scraping Wikipedia page has a concise definition of many concepts discussed here.
- This case study is a great example of what can be done using web scraping and a stepping stone to a more advanced python library scrapy.
- This recent case about Linkedin data is a good read.
- A crisp and simple explanation to robots.txt can be found here.
- Commencing 25 May 2018, Monash University will also become subject to the European Union's General Data Protection Regulation (GDPR).
- Software Carpentry is a non-profit organisation that runs learn-to-code workshops worldwide. All lessons are publicly available and can be followed indepentently. This lesson is heavily inspired by Software Carpentry.
- Data Carpentry is a sister organisation of Software Carpentry focused on the fundamental data management skills required to conduct research.
- Library Carpentry is another Software Carpentry spinoff focused on software skills for librarians.