1 **Web Scraping with Python: Collecting More Data from the Modern Web**

I prepared a book “Web Scraping with Python: Collecting More Data from the Modern Web” by Ryan Mitchell for home reading. I learned about it when I needed to do one programming practice task . And now I will briefly tell about it. So, let's get started.

2 **About the Author**

Ryan Mitchell is a senior software engineer at HedgeServ in Boston, where she develops APIs and data analytics tools for hedge fund managers. She is a graduate of Olin College of Engineering and Harvard University Extension School with a master’s in software engineering and certificate in data science. Since 2012 she has regularly consulted, lectured, and run workshops around the country on the topics of web scraping, Python automation tools, and data science.

**3 About Web Scraping With Python**

«The tools and examples included in the book allowed me to easily automate several repetitive tasks, freeing that time to solve more interesting problems. It is a results-oriented, quick read that is well rooted in real-world problems and solutions." -Eric VanWyk Electrical Computer Engineer, Olin College of Engineering

To my mind, this book assumes you are familiar with Python and have some basic understanding of network requests and HTML. While Python libraries make the actual demo code fairly concise, complete beginners might have a hard time following. I personally found the material to be well-organized, with good pacing and useful examples.

**4 Table of Contents (Part 1)**

The first part that I read, focuses on web scraping mechanics: using Python to request information from a web server, performing basic handling of the server’s response, and interacting with sites in an automated fashion. This part consists of six chapters:

Your First Web Scraper

Advanced HTML Parsing

Writing Web Crawlers

Web Crawling Models

Scrapy

Storing Data

**5** **Chapter 1. Your First Web Scraper**

This section includes topics such as:

Connecting

An Introduction to BeautifulSoup

Once you start web scraping, you start to appreciate all the little things that browsers do for you. The web, without a layer of HTML formatting, CSS styling, JavaScript execution, and image rendering, can look a little intimidating at first, but in this chapter, you’ll cover how to format and interpret data without the help of a browser.

This chapter starts with the basics of sending a GET request (a request to fetch, or “get,” the content of a web page) to a web server for a specific page, reading the HTML output from that page, and doing some simple data extraction in order to isolate the content that you are looking for.

**6 Chapter 2. Advanced HTML Parsing**

This chapter includes topics such as:

You Don’t Always Need a Hammer

Another Serving of BeautifulSoup

Regular Expressions

Regular Expressions and BeautifulSoup

Accessing Attributes

Lambda Expressions

In this chapter, author use many techniques to chip away the content that doesn’t look like the content that you’re searching for, until you arrive at the information you’re seeking. In this chapter, readers take look at parsing complicated HTML pages in order to extract only the information you are looking for.

**7 Chapter 3. Writing Web Crawlers**

This section includes topics such as:

Traversing a Single Domain

Crawling an Entire Site

Crawling Across the Internet

In this third chapter , readers can start look at real-world problems, with scrapers traversing multiple pages and even multiple sites. Web crawlers are called such because they crawl across the web. At their core is an element of recursion. They must retrieve page contents for a URL, examine that page for another URL, and retrieve that page, ad infinitum. But the author emphasizes beware, however: just because we can crawl the web doesn’t mean that we always should.

**8** **Chapter 4. Web Crawling Models**

This chapter includes topics such as:

Planning and Dealning Objects

Dealing with Different Website Layouts

Structuring Crawlers

Thinking About Web Crawler Models

This chapter focuses primarily on web crawlers that collect a limited number of “types” of data (such as restaurant reviews, news articles, company profiles) from a variety of websites, and that store these data types as Python objects that read and write

**9** **Thank you for listening!**