

Week 5: HTML, CSS, and Scraping Static Websites

LSE MY472: Data for Data Scientists

<https://lse-my472.github.io/>

Autumn Term 2024

Ryan Hübert

Plan for today

- Introduction
- Some key features of the internet
- HTML and CSS
- Fundamentals of web scraping
- Coding

Introduction

Examples

An increasing amount of data is available on the web

- Speeches, biographical information ...
- Social media data, articles, press releases ...
- Geographic information, conflict data ...

These datasets are often provided in an **unstructured format**

Web scraping is the process of extracting this information automatically and transforming it into a **structured dataset**

Why automate?

Copy & pasting is time-consuming, boring, prone to errors, and impractical or infeasible

In contrast, automated web scraping

1. Scales well for large datasets
2. Allows for dynamic data collection
3. Is (mostly) reproducible
4. Involves adaptable techniques
5. Facilitates detecting and fixing errors

When to scrape?

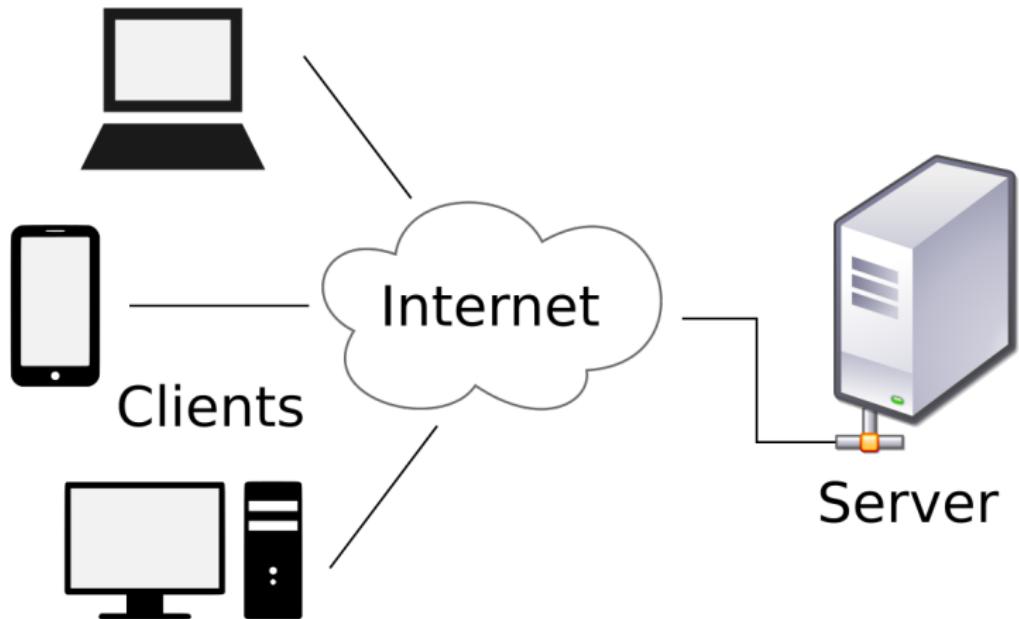
1. Trade-off between your time today and your time in the future.
Invest in your future self!
2. Computer time is often cheap; human time more expensive

Obtaining data from the web: Two approaches

1. **Screen scraping**: Extract data from source code of website, with an html parser and/or regular expressions
 - `rvest` (this week) and `RSelenium` packages (week 7) in R
2. **Web APIs** (week 8): A set of structured http requests that return JSON or XML data
 - `httr` package to construct API requests
 - Packages specific to each API: For example `WDI`, `Rfacebook`
 - Check CRAN Task View on [Web Technologies and Services](#) for examples

Some key features of the internet

Client-server model

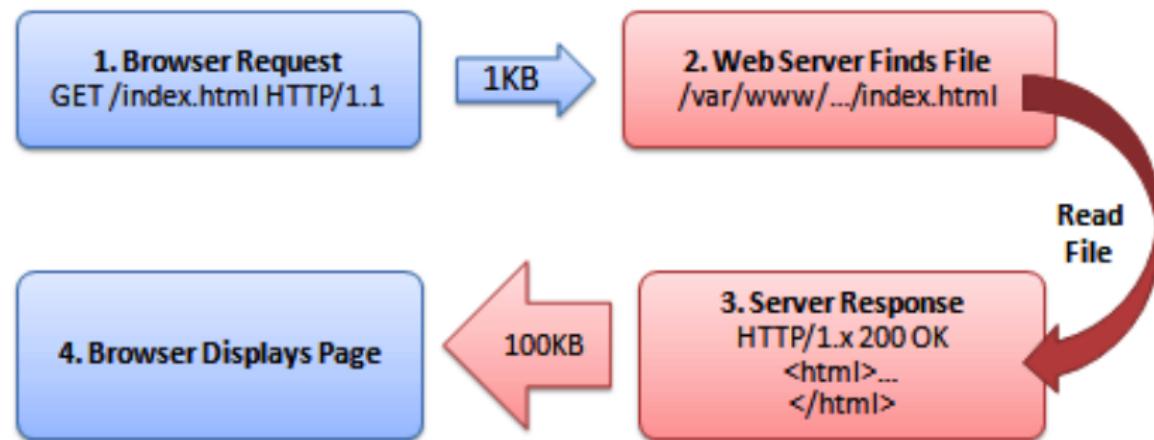


Client-server model

- Client: User computer, tablet, phone, software application, etc.
 - Server: Web server, mail server, file server, Jupyter server, etc.
1. Client makes request to the server
 - Depending on what you want to get, the request might be
 - HTTP: Hypertext Transfer Protocol
 - HTTPS: Hypertext Transfer Protocol Secure
 - SMTP: Simple Mail Transfer Protocol
 - FTP: File Transfer Protocol
 2. Server returns response

Request and response in the case of HTTP

From StackOverflow



Source code: HTML and CSS

Source code

- A webserver returns a combination of text and multimedia files (images, videos, etc.) that are used to display a website
- Each “webpage” is typically a plain text file coded in a combination of languages: HTML, CSS and JavaScript
- These plain text files contain instructions about how the webpage should look
- The purpose of a web browser (Chrome, Safari, Firefox, etc.) is to take these plain text files and render them according to the instructions so that a user sees something “nice”
 - Different browsers have different bells and whistles, but they all basically do the same thing
- We often refer to the underlying plain text files as the **source code** for the webpage

Browser “developer tools”

In Chrome:

- Look at plain text version of page in source code:
View > Developer > View Source
- Look at suite of developer tools:
View > Developer > Developer Tools

In Safari:

- Enable developer settings: Safari > Settings > Advanced > Show features for web developers
- Look at plain text version of page in source code:
Develop > Show Page Source
- Look at suite of developer tools:
Develop > Show Web Inspector

Simple example: <https://lse-my472.github.io/>

The screenshot shows a web browser window with the title bar "LSE MY472 Data for Data Sci". The address bar displays the URL "lse-my472.github.io". The main content area has a dark blue header with the text "LSE MY472 Data for Data Scientists" in white. Below the header, a green section contains the text "Course Website for Autumn 2024". The main body of the page is white and features a "Welcome!" message, followed by sections on "Course format and scheduling", "Lectures", "Seminars", and "Important course links".

Welcome!

Course format and scheduling

Lectures: There is a two-hour lecture each week during the term on **Wednesdays from 13:00 to 15:00 in CLM.2.02.**

Seminars: There is a one-hour “lab-style” seminar each week during the term. See the [LSE Timetable](#) for the schedule and locations for the seminars.

There are no lectures or seminars during week 6, which is LSE’s reading week.

Important course links

- [Course Moodle page](#)

Simple example: Viewing source code

```
 25 <meta name="viewport" content="width=device-width, initial-scale=1">
 26 <meta name="theme-color" content="#157878">
 27 <meta name="apple-mobile-web-app-status-bar-style" content="black-translucent">
 28 <link rel="stylesheet" href="/assets/css/style.css?v=0a429e13ebc1a52ebcd4f7fc8e785e841129eed">
 29 <!-- start custom head snippets, customize with your own _includes/head-custom.html file -->
30
31 <!-- Setup Google Analytics -->
32
33
34
35 <!-- You can set your favicon here -->
36 <!-- link rel="shortcut icon" type="image/x-icon" href="/favicon.ico" -->
37
38 <!-- end custom head snippets -->
39
40 </head>
41 <body>
42   <a id="skip-to-content" href="#content">Skip to the content.</a>
43
44   <header class="page-header" role="banner">
45     <h1 class="project-name">LSE MY472 Data for Data Scientists</h1>
46     <h2 class="project-tagline">Course Website for Autumn 2024</h2>
47
48
49   </header>
50
51   <main id="content" class="main-content" role="main">
52     <p>Welcome!</p>
53
54   <h2 id="course-format-and-scheduling">Course format and scheduling</h2>
55
56   <p><strong>Lectures:</strong> There is a two-hour lecture each week during the term on <strong>Wednesdays from 13:00 to 15:00 in CLM.2.02</strong></p>
57
58   <p><strong>Seminars:</strong> There is a one-hour "lab-style" seminar each week during the term. See the <a href="https://info.lse.ac.uk/</a></p>
59
60   <p><em>There are no lectures or seminars during week 6, which is LSE's reading week.</em></p>
61
62   <h2 id="important-course-links">Important course links</h2>
63
64   <ul>
65     <li><a href="https://moodle.lse.ac.uk/course/view.php?id=10954">Course Moodle page</a></li>
66     <li><a href="https://github.com/lse-my472/lse-my472.github.io">Course GitHub repository</a></li>
67   </ul>
68
69   <h2 id="instructors">Instructors</h2>
70
71   <p>Office hour slots with all instructors should be booked via LSE's <a href="https://studenthub.lse.ac.uk/">StudentHub</a>.</p>
72
73   <ul>
74     <li><a href="mailto:r.hubert@lse.ac.uk">Ryan Hubert</a>, Department of Methodology. <em>Course convenor</em>.</li>
```

Simple example: Using developer tools

The screenshot shows a web browser window with the URL lse-my472.github.io. The main content area displays the "LSE MY472 Data for Data Scientists" page, featuring a green header with the title and a white body with course details. A developer tools panel is open on the right side of the browser.

Elements Tab: This tab shows the DOM structure of the page. The selected element is the `<h1>` tag with the class `project-name`, which contains the text "LSE MY472 Data for Data Scientists". Other visible elements include the `<head>` and `<body>` tags.

```
<!DOCTYPE html>
<html lang="en-US"> scroll
  ><head>::</head>
    ... <body> == $0
      <a id="skip-to-content" href="#content">Skip to the content.</a>
      <header class="page-header" role="banner">
        <h1 class="project-name">LSE MY472 Data for Data Scientists</h1>
        <h2 class="project-tagline">Course Website for Autumn 2024</h2>
      </header>
      <main id="content" class="main-content" role="main">::</main>
    </body>
</html>
```

Styles Tab: This tab shows the CSS styles applied to the selected element. It includes rules from the main style sheet (`style.css?v_841129ee4d1`) and the user agent stylesheet (`user agent stylesheet`). The main rule is `element.style { }`.

```
element.style {
}
body {
  padding: 0;
  margin: 0;
  font-family: "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-size: 16px;
  line-height: 1.5;
  color: #606c71;
}
body {
  margin: 0;
}
* {
  box-sizing: border-box;
}
```

Console Tab: This tab is currently empty, showing no output.

HTML

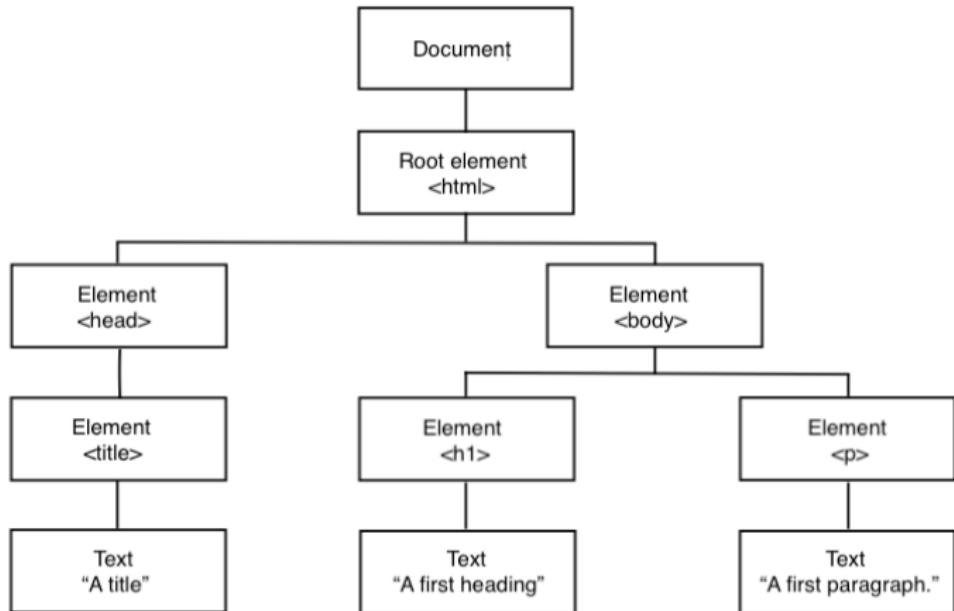
HTML: Hypertext Markup Language

- HTML displays mostly **static** content
 - Contents of many dynamic webpages cannot be found in HTML, e.g. **Google Maps**



- Understanding what is static and dynamic in a webpage is a crucial first step for web scraping

HTML tree structure



A very simple HTML file

<https://lse-my472.github.io/week05/data/html1.html>

```
<!DOCTYPE html>
<html>
  <head>
    <title>A title</title>
  </head>
  <body>
    <h1>A first heading</h1>
    <p>A first paragraph.</p>
  </body>
</html>
```

Inspiration: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro

Slightly more features

<https://lse-my472.github.io/week05/data/html2.html>

```
<!DOCTYPE html>
<html>
  <head>
    <title>A title</title>
  </head>
  <body>
    <h1>A first heading</h1>
    <p>A first paragraph.</p>
    <p>A second paragraph with some
      <b>formatted</b> text.</p>
    <p>A third paragraph with a
      <a href="http://www.lse.ac.uk">hyperlink</a>.</p>
  </body>
</html>
```

With some content divisions

<https://lse-my472.github.io/week05/data/html3.html>

```
<!DOCTYPE html>
<html>
  <head>
    <title>A title</title>
  </head>
  <body>
    <div>
      <h1>Heading of the first division</h1>
      <p>A first paragraph.</p>
      <p>A second paragraph with some
          <b>formatted</b> text.</p>
      <p>A third paragraph with a
          <a href="http://www.lse.ac.uk">hyperlink</a>.</p>
    </div>
    <div>
      <h1>Heading of the second division</h1>
      <p>Another paragraph with some text.</p>
    </div>
  </body>
</html>
```

Beyond plain HTML

1. **Cascading Style Sheets (CSS)**: “Style sheet” language which describes formatting of HTML components, useful for us because of **selectors**
2. **Javascript**: Adds functionalities to the websites, e.g. change content/structure after website has been loaded
 - This usually makes webpages interactive

Adding some simple CSS to the last example (1/2)

<https://lse-my472.github.io/week05/data/css1.html>

```
<!DOCTYPE html>
<html>
  <head>
    <!-- CSS start -->
    <style>
      p {
        color: green;
      }
    </style>
    <!-- CSS end -->

    <title>A title</title>
  </head>
  <body>
    ...
  </body>
</html>
```

Adding some simple CSS to the last example (2/2)

<https://lse-my472.github.io/week05/data/css2.html>

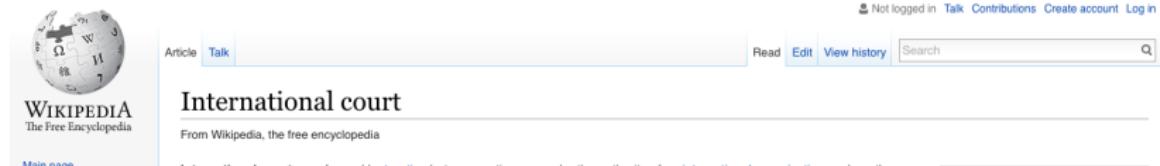
```
<!DOCTYPE html>
<html>
  <head>

    <!-- CSS start -->
    <style>
      .text-about-web-scraping {
        color: orange;
      }
      .division-two h1 {
        color: green;
      }
    </style>
    <!-- CSS end -->

    <title>A title</title>
  </head>
  <body>
    ...
  </body>
</html>
```

Fundamentals of web scraping

Scenario 1: Data in table format



The screenshot shows a Wikipedia page for "International court". The top navigation bar includes links for "Article", "Talk", "Read", "Edit", "View history", and "Search". The main content area features the title "International court" and a summary from Wikipedia. Below the summary is a section titled "List of international courts" with a link to edit it. A table is displayed below this section, listing various international courts along with their names, scopes, years active, and subject matters.

List of international courts [edit]

Name	Scope	Years active	Subject matter
International Court of Justice	Global	1945–present	General disputes
International Criminal Court	Global	2002–present	Criminal prosecutions
Permanent Court of International Justice	Global	1922–1946	General disputes
Appellate Body	Global	1995–present	Trade disputes within the WTO
International Tribunal for the Law of the Sea	Global	1994–present	Maritime disputes
African Court of Justice	Africa	2009–present	Interpretation of AU treaties
African Court on Human and Peoples' Rights	Africa	2006–present	Human rights
COMESA Court of Justice	Africa	1998–present	Trade disputes within COMESA
ECOWAS Community Court of Justice	Africa	1996–present	Interpretation of ECOWAS treaties
East African Court of Justice	Africa	2001–present	Interpretation of EAC treaties
SADC Tribunal	Africa	2005–2012	Interpretation of SADC treaties

Scenario 2: Data in “unstructured” format

LSE People Ise.ac.uk/Methodology/People Incognito (2) Search Menu

Department of Methodology 30 Year Anniversary About us Alumni Methods surgery Methods Short Courses People Research Seminar series Software tutorials Study

Department of Methodology > People

People

Who's who in the Department of Methodology

 30TH ANNIVERSARY

In Conversation... Our faculty share career highlights, future collaborations and more in this special insight

Senior Leadership Team

Share

Professor Patrick Sturgis
Professor of Quantitative Social Science
Head of Department of Methodology
[More information](#)

Professor Flora Cornish
Professor in Research Methodology
Deputy Head of Department (Research)
Doctoral Programmes Director
[More information](#)

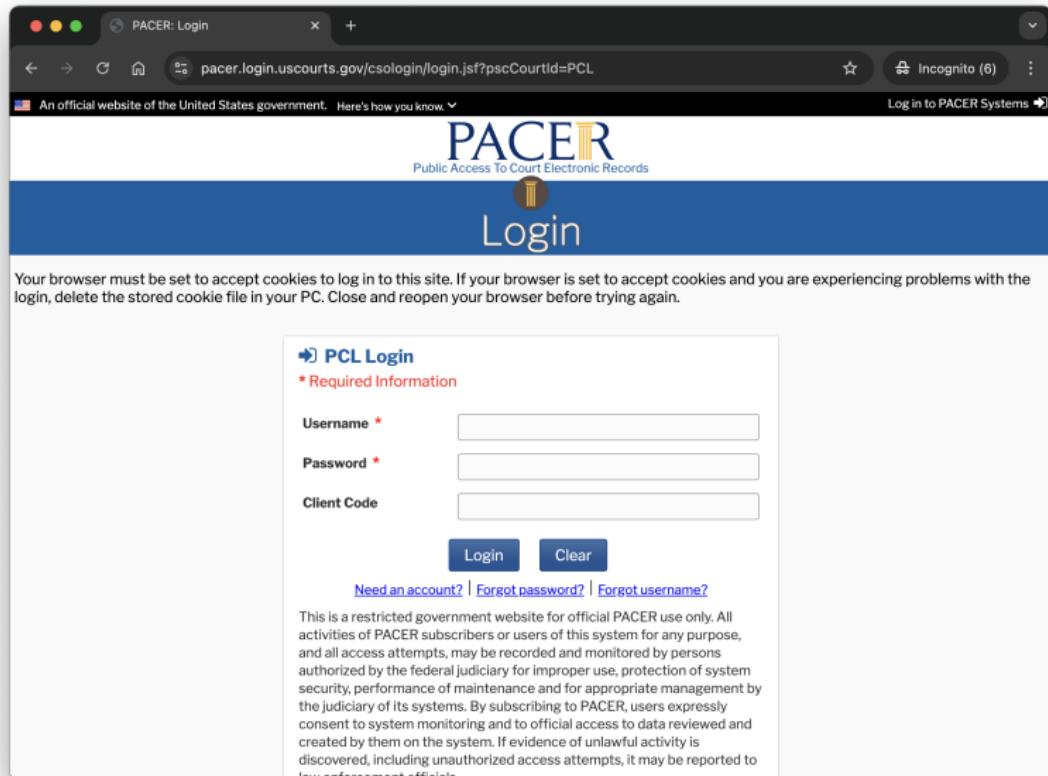
Professor Jonathan Jackson

Office Hours Book using Student Hub





Scenario 3: “Hidden” behind web forms



Your browser must be set to accept cookies to log in to this site. If your browser is set to accept cookies and you are experiencing problems with the login, delete the stored cookie file in your PC. Close and reopen your browser before trying again.

PCL Login
* Required Information

Username *

Password *

Client Code

Login **Clear**

[Need an account?](#) | [Forgot password?](#) | [Forgot username?](#)

This is a restricted government website for official PACER use only. All activities of PACER subscribers or users of this system for any purpose, and all access attempts, may be recorded and monitored by persons authorized by the federal judiciary for improper use, protection of system security, performance of maintenance and for appropriate management by the judiciary of its systems. By subscribing to PACER, users expressly consent to system monitoring and to official access to data reviewed and created by them on the system. If evidence of unlawful activity is discovered, including unauthorized access attempts, it may be reported to law enforcement officials.

Scenario 3: “Hidden” behind web forms

The screenshot shows a web browser window for the PACER Case Locator. The URL is pcl.uscourts.gov/pcl/pages/search/findCase.jsf. The page title is "Case Search | PACER: PCL". The header includes the PACER logo and "Public Access To Court Electronic Records". A red circular icon with a question mark is overlaid on the logo. The main content area is titled "PACER Case Locator". It features a navigation bar with "New Search", "Saved Items", "Court Information", "Settings", and a user profile for "Ryan Hubert". Below the navigation is a breadcrumb trail: "Case Search". The "Case Information" section contains fields for "Court Type" (set to "All"), "Case Number" (1:24-cr-00556), "Title", and "Type". There is also a "Case Status" dropdown set to "All". A note at the bottom states: "NOTE: Newly filed cases will typically appear on this system within 24 hours. Check the [Court Information](#) page for data that is currently available on the PCL. The most recent data is available directly from the court." At the bottom are "Search" and "Clear" buttons, and a checkbox to "Make this my PCL home page".

Three main scenarios

1. Data in *table* format
 - Automatic extraction with `rvest` or select specific table with *inspect element* in browser
2. Data in *unstructured* format
 - Element identification key in this case
 - *Inspect element* in browser
 - Identify the target e.g. with *CSS* (this week) or *XPath* selector (week 7)
 - Automatic extraction with `rvest`
3. Data hidden *behind web forms* (week 7)
 - Element identification to find text boxes, buttons, results, etc.
 - Automation of web browser with `RSelenium`

Identifying elements via CSS selector (1/2)

- Selecting by tag-name
 - Example html code: <h3>This is the main item</h3>
 - Selector: h3
- Selecting by class
 - Example html code: <div class = 'itemdisplay'>This is the main item</div>
 - Selector: .itemdisplay
- Selecting by id
 - Example html code: <div id = 'maintitle'>my main title</div>
 - Selector: #maintitle

Identifying elements via CSS selector (2/2)

- Selecting by tag structure

- Example html code (hyperlink tag a inside div tag): <div>Google Link</div>

- Selector: div a

- Selecting by nth child of a parent element

- Example html code: <body><p>First paragraph</p><p>Second paragraph.</p></body>

- Selector of second paragraph: body > p:nth-child(2)

You don't have to figure these out yourself: inspect!

Reference and further examples:

https://www.w3schools.com/cssref/css_selectors.asp

The rules of the game

1. Respect the hosting site's wishes
 - Check if an API exists or if data are available for download
 - Respect copyright and ethics; what are you allowed to do?
 - Keep in mind where data comes from and give credit
 - Some websites disallow scrapers via `robots.txt` file
2. Limit your bandwidth use
 - Wait some time after each hit
 - Scrape only what you need, and just once
3. When using APIs, read documentation
 - Is there a batch download option?
 - Are there any rate limits?
 - Can you share the data?

Coding

Markdown files this week

- 01-selecting-elements.Rmd
- 02-scraping-tables.Rmd