

MY472 - Data for Data Scientists

Week 5: HTML, CSS, and Scraping

Static Websites

Daniel de Kadt
23 October 2023

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

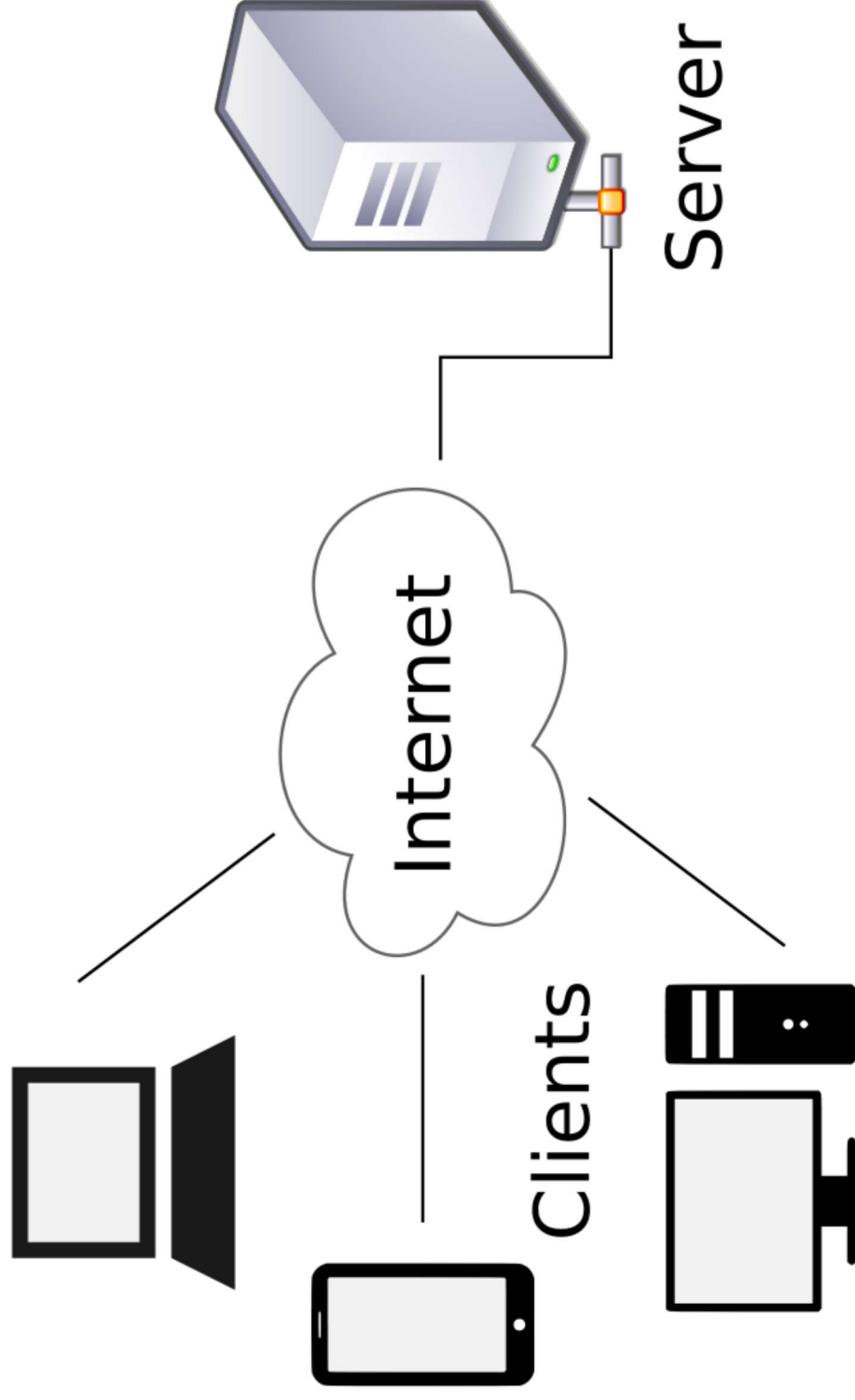
Obtaining data from the web: Two approaches

Two different approaches

1. Screen scraping Extract data from source code of website, with html parser and/or regular expressions
 - `rvest` (this week) and `R Selenium` 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.

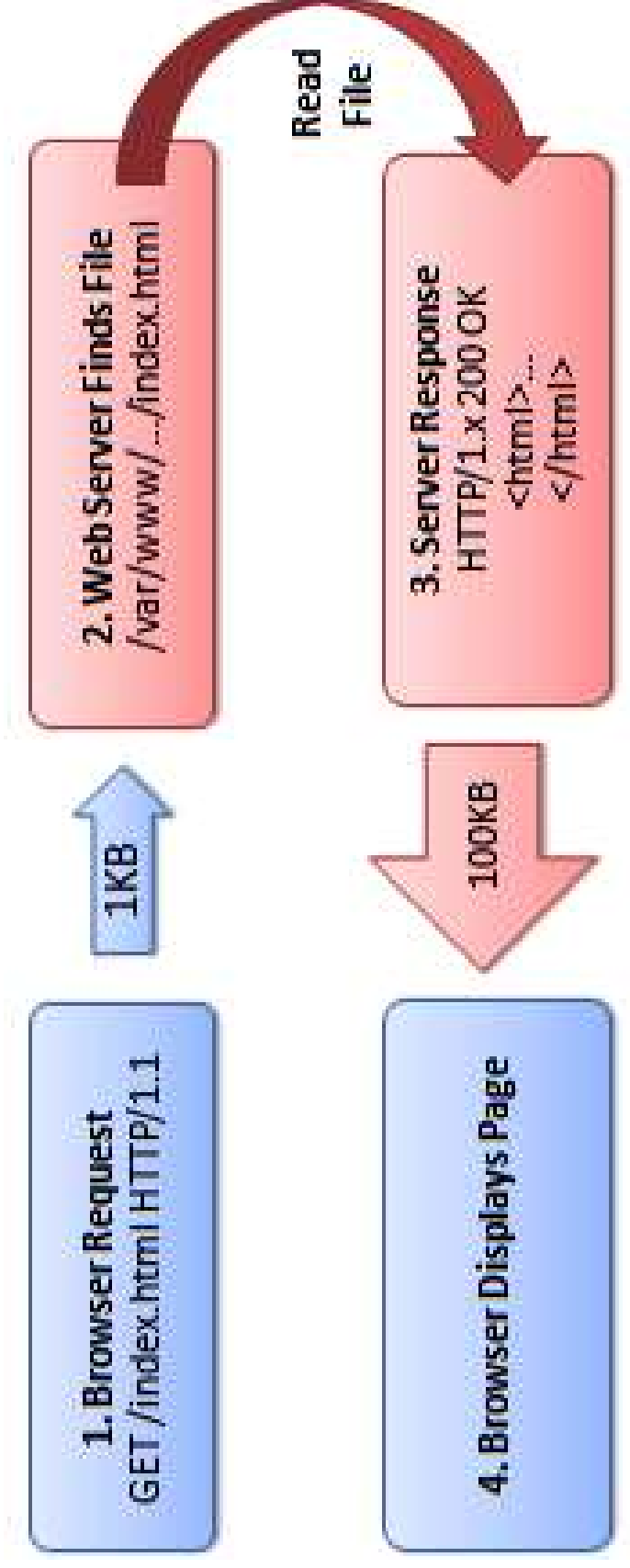
1. Client makes request to the server

- 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

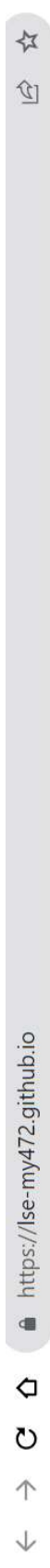
Request and response in the case of HTTP

From [StackOverflow](#)



Simple example: MY472 website

Let's see a very simple example of <https://lse-my472.github.io>



MY472 Data for Data Scientists

Michaelmas Term 2023

[Main course repo](#)

[Moodle page](#)

Simple example: MY472 website

Name	X	Headers	Preview	Response	Initiator	Timing
 lse-my472.git...		▼ General				
 css?family=Op...		Request URL:			https://lse-my472.github.io/	
 style.css?v=78...		Request Method:			GET	
 css?family=Op...		Status Code:			 304 Not Modified	
 memvYaGs126...		Remote Address:			185.199.108.153:443	
		Referrer Policy:			strict-origin-when-cross-origin	

Simple example: Request headers

Name	X	Headers	Preview	Response	Initiator	Timing
 lse-my472.git...		▼ Request Headers				
 css?family=Op...		:authority:			lse-my472.github.io	
 style.css?v=78...		:method:			GET	
 css?family=Op...		:path:			/	
 memvYaGs126...		:scheme:			https	
		Accept:			text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7	
		Accept-Encoding:			gzip, deflate, br	
		Accept-Language:			en-GB,en-US;q=0.9,en;q=0.8	
		Cache-Control:			max-age=0	
		If-Modified-Since:			Mon, 23 Oct 2023 18:57:27 GMT	
		If-None-Match:			W/"6536c217-6b3e"	
		Sec-Ch-Ua:			"Chromium";v="118", "Google Chrome";v="118", "Not=A?Brand";v="99"	
		Sec-Ch-Ua-Mobile:			?0	
		Sec-Ch-Ua-Platform:			"Windows"	
		Sec-Fetch-Dest:			document	
		Sec-Fetch-Mode:			navigate	
		Sec-Fetch-Site:			none	
		Sec-Fetch-User:			?1	
		Upgrade-Insecure-Requests:			1	
		User-Agent:			Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36	
5 requests						307 E

Simple example: Response headers

Name	X	Headers	Preview	Response	Initiator	Timing
 lse-my472.git...		▼ Response Headers				
 css?family=Op...		Age:		0		
 style.css?v=78...		Cache-Control:		max-age=600		
 css?family=Op...		Date:		Mon, 23 Oct 2023 19:25:46 GMT		
 memvYaGs126...		Etag:		W/"6536c217-6b3e"		
		Expires:		Mon, 23 Oct 2023 19:34:43 GMT		
		Vary:		Accept-Encoding		
		Via:		1.1 varnish		
		X-Cache:		HIT		
		X-Cache-Hits:		1 		
		X-Fastly-Request-Id:		fce072886916615cb28cf6d34f4bc76ede90a004		
		X-Served-By:		cache-lin2290031-LIN		
		X-Timer:		S1698089147.771362,VSO,VE116		

Simple example: Reponse content

←

→

🔍

📄

🌐

🌟

view-source:https://lse-my472.github.io

Line wrap

```
1 <!DOCTYPE html>
2 <html lang="en-US">
3   <head>
4     <meta charset="UTF-8">
5
6     <!-- Begin Jekyll SEO tag v2.8.0 -->
7     <title>LSE MY472 Data for Data Scientists | Course Website for Michaelmas Term 2023</title>
8     <meta name="generator" content="Jekyll v3.9.3" />
9     <meta property="og:title" content="LSE MY472 Data for Data Scientists" />
10    <meta property="og:locale" content="en_US" />
11    <meta name="description" content="Course Website for Michaelmas Term 2023" />
12    <meta property="og:description" content="Course Website for Michaelmas Term 2023" />
13    <link rel="canonical" href="https://lse-my472.github.io/" />
14    <meta property="og:url" content="https://lse-my472.github.io/" />
15    <meta property="og:site_name" content="LSE MY472 Data for Data Scientists" />
16    <meta property="og:type" content="website" />
17    <meta name="twitter:card" content="summary" />
18    <meta property="twitter:title" content="LSE MY472 Data for Data Scientists" />
19    <script type="application/ld+json">
20      {"@context": "https://schema.org", "@type": "Website", "description": "Course Website for Michaelmas Term 2023", "headline": "LSE MY472
      MY472 Data for Data Scientists", "url": "https://lse-my472.github.io/"}</script>
21    <!-- End Jekyll SEO tag -->
```

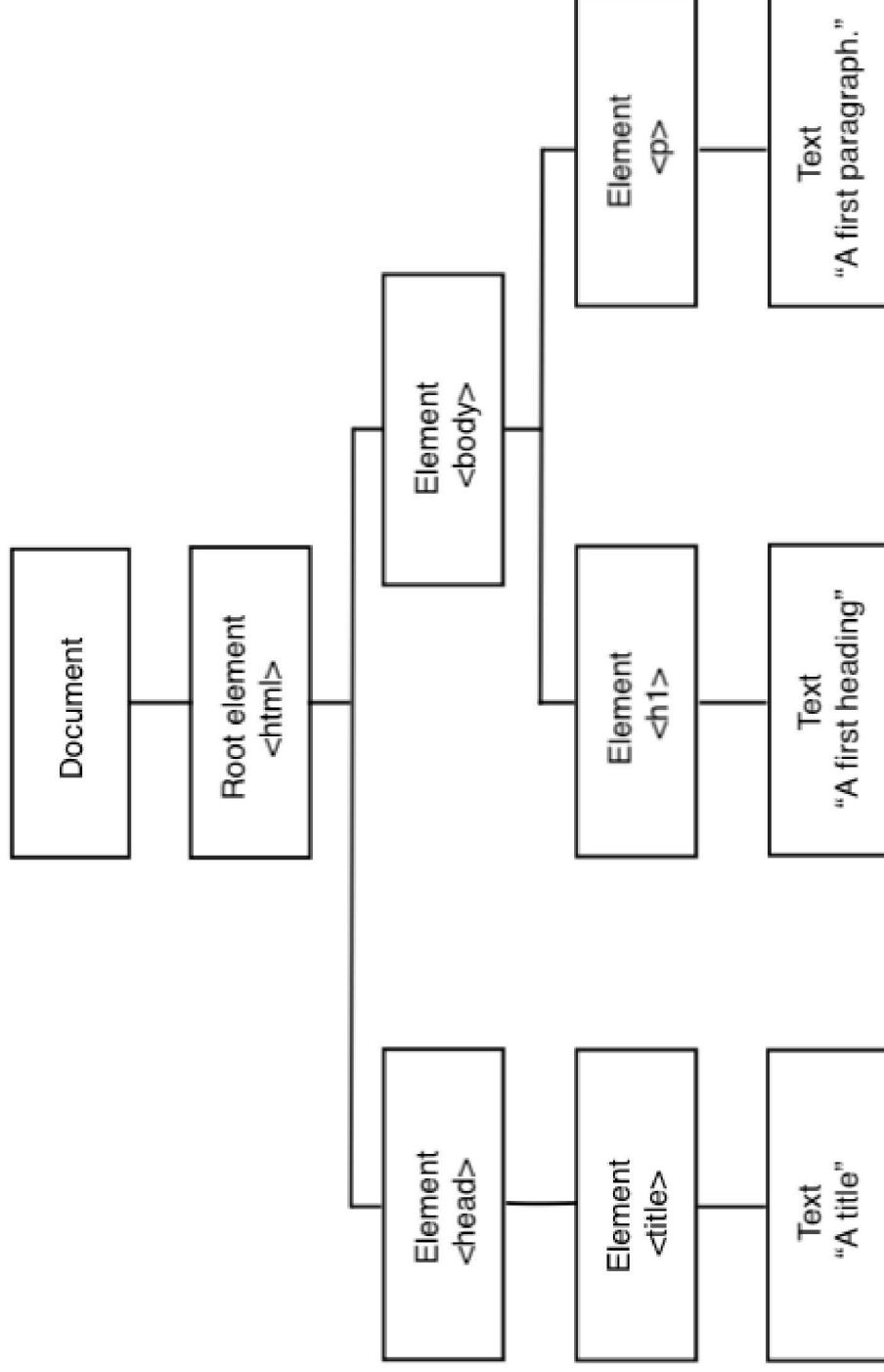
HTML and CSS

HTML

HTML: Hypertext Markup Language

- HTML displays mostly **static** content
- Many contents of dynamic webpages cannot be found in HTML
 - Example: 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

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

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

Slightly more features

```
<!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

```
<!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

Adding some simple CSS (1/2)

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

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

    <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>
```

Adding some simple CSS (2/2)


```
<!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>
    <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 class="text-about-web-scraping">A third paragraph now containing some text about web scraping ...</p>
    </div>
    <div class="division-two">
      <h1>Heading of the second division</h1>
      <p>Another paragraph with some text.</p>
      <p class="text-about-web-scraping">A last paragraph discussing some web scraping ...</p>
    </div>
  </body>
```


Fundamentals of web scraping

Scenario 1: Data in table format



WIKIPEDIA

The Free Encyclopedia

ArticleTalk

International court

From Wikipedia, the free encyclopedia

Not logged inTalkContributionsCreate accountLog in

ReadEditView history

Search

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

India

English

Search

Register for updates

11,072,800 Visitors

ALL REPORTS

NEWS

BRIBE HOTLINE

I DID NOT PAY A BRIBE

I MET AN HONEST OFFICER

REPORT A BRIBE

ALL / IPaid A BRIBE / BRIBE FIGHTER / HONEST OFFICER / BRIBE HOTLINE

IPaid A BRIBE

1 day ago

76 views

POLICE NILO GHUSS (bribe)

Passport | Police Verification for Passport | Paid INR 5,000

Reported on January 17, 2016 from Bankura, West Bengal | Report #89544

What will happen to this country.. police mamu's govt income: 30,000 per month. Per day GHUSS income 5000 (per passport verification). Imagine they t...[Read more](#)

How to Get a Passport Verified in Ghaziabad

ALL / IPaid A BRIBE / 1 day ago

104 views

Corruption due to vague rules

Police | Traffic Violations | Paid INR 500

Reported on January 16, 2016 from Mumbai, Maharashtra | Report #89509

At Chembur near Eastern Expressway traffic cop stopped me and started checking docs..all was fine buy puc expired..then he pointed out film.. He took....[Read more](#)

Things to Know on Traffic Offences and Respective Penalties

ALL / IPaid A BRIBE / 2 days ago

105 views

Bribe collected by Staff of Enrollment agency

Municipal Services | Aadhaar or UID Related | Paid INR 120

Reported on January 16, 2016 from Mysore, Karnataka | Report #89467

UIDAI has to take a stand on fees to be paid to enrolment agencies for processing Adhaar

FILTER REPORTS

Which city?

All cities

Department

All departments

Bribe Amount

All Amount

SUBMIT

INSPIRE OTHERS WITH YOUR STORY

Manik Taneja, a sports enthusiast, wrote against a custom official on [Ipaidabribe.com](#), for cough up a hefty bribe by a Customs official at Bengaluru airport.


SEE HIS STORY

Ever Paid A Bribe?


Report your Bribe Story!


See action taken.


Scenario 3: Hidden behind web forms





MONITOR
LEGISLATIVO


 INICIO


 PERFIL IDEAL


 NOTICIAS

 CANDIDATOS

 ASAMBLEA NACIONAL

 ABUSOS

 CONTACTENOS




RESULTADOS DE LA CONSULTA

Seleccione


Partido

BUSCAR


DIPUTADOS ENCONTRADOS




Julio Ygarza
Estado: Amazonas




Mauligmer Baloa
Estado: Amazonas




Nirma Guarulla
Estado: Amazonas



José Brito
Estado: Anzoátegui



Chaím Bucarán
Estado: Anzoátegui



Richard Arteaga
Estado: Anzoátegui

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 e.g. find text boxes, buttons, and results
 - 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