

# DATA2001

Data Science, Big Data  
and Data Variety

**Week 6**  
*Web Scraping*



THE UNIVERSITY OF  
**SYDNEY**

# Outline

- Introduction – 10 mins
- Scraping Data Exploration – 50 mins
- Web Traversal, Storage and Querying – 40 mins

*Fair warning: today's tutorial is quite Python heavy, and may get quite detailed.  
Don't stress about remembering all the syntactic semantics – our focus today is  
**exploring a semi-structured data source and exercising problem solving.**  
(even if the example can be a little tricky!)*



# Dataset

Library Current students Staff intranet Find an event Give



Study Research Engage with us About us News & opinion Q

What will you start here?

Ranked 1st in Australia for graduate employability

Explore your study options →

Covid-19 updates: University of Sydney response

More information



## Current students

[Units](#) / DATA2001

Unit of study\_

### **DATA2001: Data Science, Big Data and Data Variety**

This course focuses on methods and techniques to efficiently explore and analyse large data collections. Where are hot spots of pedestrian accidents across a city? What are the most popular travel locations according to user postings on a travel website? The ability to combine and analyse data from various sources and from databases is essential for informed decision making in both research and industry. Students will learn how to ingest, combine and summarise data from a variety of data models which are typically encountered in data science projects, such as relational, semi-structured, time series, geospatial, image, text. As well as reinforcing their programming skills through experience with relevant Python libraries.

Not all data is presented as neatly as a structured dataframe of rows and columns, like the datasets we've used so far. Often, meaningful information exists in unstructured or semi-structured formats.

Today we'll explore how to extract data from **webpages**, by focussing firstly on a familiar example – the online **UoS outline** for this subject – and then investigating how this can be scaled up in application.

# Webpages

- Documents intended for web browsers are written in **HTML** (HyperText Markup Language)
  - These are tree-like structures, comprising multiple elements that start and end with tags

- e.g.

```
<div class="firstSection">
  <h1>This is a heading!</h1>
  <p>This is a paragraph containing text</p>
  <a href="https://bit.ly/3JX9nLM">This is a link</a>
</div>
```

→ this would have a heading element ('h1' tag), a paragraph ('p' tag) and a hyperlink ('a' tag), all contained within a 'div'

→ elements can have classes (not unique), or a single id (unique), which is particularly useful for setting styles with **CSS**

# HTML



# Inspect Element

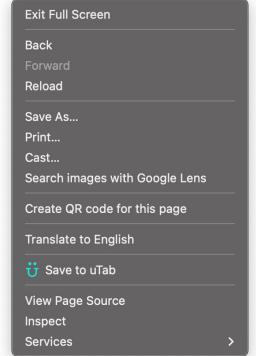
Webpage

The screenshot shows a university website page for a course. At the top, there's a navigation bar with a menu icon, the university logo, and a search icon. Below the header, the title "Current students" is underlined. A breadcrumb trail "Units / DATA2001" is visible. The main content area has a blue header "Unit of study\_" followed by the course title "DATA2001: Data Science, Big Data and Data Variety". The main text describes the course content, mentioning methods and techniques for exploring large data collections, and how it relates to real-world applications like pedestrian accidents and travel locations. Below the text is a table with course details: Code (DATA2001), Academic unit (Computer Science), and Credit points (6). At the bottom, there's a section titled "Unit outlines".

Content

The screenshot shows the browser's developer tools with the "Elements" tab selected. The left pane displays the HTML structure of the page, including the DOCTYPE, head, and body elements. The right pane shows the CSS styles applied to the selected element, which is the "pageTitleModule" div containing the course title. The styles include a margin-top of 14px and other properties for webkit and moz box-sizing. The bottom pane shows the "user agent stylesheet".

Right clicking anywhere on a webpage should give an "**Inspect**" or "Inspect Element" option, which reveals, and allows interaction with, the underlying source code.



Hovering over an element in the HTML code will reveal where it exists on the webpage.

The **CSS styling** of each element can also be viewed when a HTML element is clicked. Here for example, we see an element with an upper margin of 14 pixels.

# Webpage Parsing

Jump to the Jupyter Notebook for this week and begin exploring the webpage.

(return and continue with the slides after completing Section 1)



The screenshot shows the University of Sydney homepage with the title "Current students". The browser's developer tools are open, specifically the "Elements" tab under the "Inspector" section. The "primaryNavigation" header is selected. The "Styles" panel on the right displays the CSS for the ".primaryNavigation" class, which includes styles like "border-box", "border-box", and "padding: 0 16px;". The "Computed" and "Layout" tabs are also visible.

Units / DATA2001 / Semester 1 2025 [Normal day]

Current students

Units / DATA2001 / Semester 1 2025 [Normal day]

Elements Console Sources Network Performance Memory >

... <div class="primaryNavigation" == \$0

<header class="mobile hidden-md hidden-lg">

<div class="tabletContainer">

<a href="javascript:void(0)" class="hamburgerIcon" tabindex="2" title="Toggle the side menu">...</a>

<a href="/content/corporate/home.html" tabindex="1">...</a>

<a href="javascript:void(0)" class="searchIcon" tabindex="1">

<span class="glyphicon glyphicon-search">...</span>

<span class="sr-only" aria-hidden="true">Search for Courses</span>

</a>

</div>

Styles Computed Layout >

Filter :hover .cls +, □, □

element.style {

.globalHeaderModule primaryNavigation {

webkit\_box\_sizing: border-box;

box-sizing: border-box;

background: □ #fff;

width: 100%;

max-width: 100%;

position: relative;

padding: □ 0 16px;

Guiding screenshot for the demonstration in Section 1.3  
Can we extract some attributes of the header links?

The screenshot shows the University of Sydney homepage with the title "Leadership for good starts here". The footer is visible, containing links to various university services like Media, Student links, About us, and Connect. Logos for APRU and Athena SWAN are present. The footer also includes links for Disclaimer and copyright, Privacy statement, Accessibility, and Website feedback, along with institutional details like ABN and CRICOS numbers.

Leadership for good starts here

THE UNIVERSITY OF SYDNEY

Media Student links About us Connect

News Find an expert Media contacts How to log in to University systems Our world university rankings Contact us

Find an expert Class timetables Faculty and schools Faculties and schools

Media contacts Policies Research centres Research centres

Policies Campus locations Campus locations

Member of GROUP OF EIGHT AUSTRALIA

APRU Athena SWAN

Disclaimer and copyright Privacy statement Accessibility Website feedback

ABN: 15 211 513 464 CRICOS Number: 00026A TEOGA: PRV12057

Guiding screenshot for the task in Section 1.3  
Can we extract the text and links for the social media platforms?

# Thought Questions



*What applications/benefits could web scraping have?*



*What challenges are faced when attempting to web scrape?*

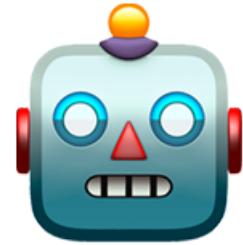


*How do we determine what is legal, and remain a good internet citizen?*

# Robots.txt

The location for websites to specify what can and can't be scraped is **robots.txt**.

For any given web domain, simply put '/robots.txt' on the end. Below is Sydney University's:



```
User-agent: FunnelBack
Disallow: /education_social_work/bulletin/

User-agent: *
Allow: /
Sitemap:https://www.sydney.edu.au/sitemap.xml
Allow: /muni-content/
Allow: /medicine-health/schools/sydney-school-of-health-sciences/academic-staff/
Allow: /science/about/our-people/academic-staff/
```

**User-agent:** the group the rules apply to (e.g. 'AdsBot-Google' may be subject to different terms of use, '\*' indicates all other users)

It will generally then detail what is allowed/disallowed. Here, the root directory ('/') is listed as "**allow**", so our use here is permissible.

Some may even specify a "**crawl-delay**", which specifies the minimum time that must be left in between requests. Even if this is not specified, **always be sure to add delays** between requests!

```
Disallow: /library/images/
Disallow: /library/scripts/
Disallow: /library/styles/
Disallow: /library/test/
Disallow: /library/templates/
Disallow: /library/stream/
Disallow: /library/screens/
Disallow: /library/cgi-bin/
Disallow: /library/unified-search/
Disallow: /library/contacts/email-campaigns/

Disallow: /styleguide/
Disallow: /agents/

Disallow: /errors/
Disallow: /architecture/about/our-people/academic-staff/staff-profile.html
Disallow: /law/about/our-people/academic-staff/staff-profile.html
Disallow: /music/about/our-people/academic-staff/staff-profile.html
Disallow: /engineering/about/our-people/academic-staff/staff-profile.html
Disallow: /medicine-health/about/our-people/academic-staff/staff-profile.html
Disallow: /medicine-health/schools/faculty-of-health-sciences/academic-staff/staff-profile.html
Disallow: /arts/about/our-people/academic-staff/staff-profile.html
```

There will also often be sites listed as "**disallow**" that should not be visited programmatically. For USYD, this entails the pages of academic staff profiles, for example.

# Tasks

*Summarising the tasks remaining, to be completed in Jupyter Notebook*



- 1.4) Extract the details of all **assessments** in the webpage.
- 2.2) Create a **function** that receives a URL + returns assessment data.
- 3.3) After ingestion, develop an SQL **query** that reports the first session, last session, and avg weight of each assessment type.
- 4) BONUS OPTIONAL TASK\*: Extract the list of **all OLEs**, and for each, extract the assessments for their most recent UoS outline. Load this into a database and determine, according to your own metrics, the 'ideal' OLE.

\*Pursue this final task with caution, and follow the prompts provided in the notebook.

# Done! Enjoy your week :)



Seth Rosen (@sethrosen) · 20 Apr 2020

Them: Can you just quickly pull this data for me?

Me: Sure, let me just:

```
SELECT * FROM  
some_ideal_clean_and_pristine.table  
_that_you_think_exists
```

1:42 PM · 20 Apr 20 · Twitter Web App

3,161 Retweets 20.5K Likes

Reply Retweet Like Share