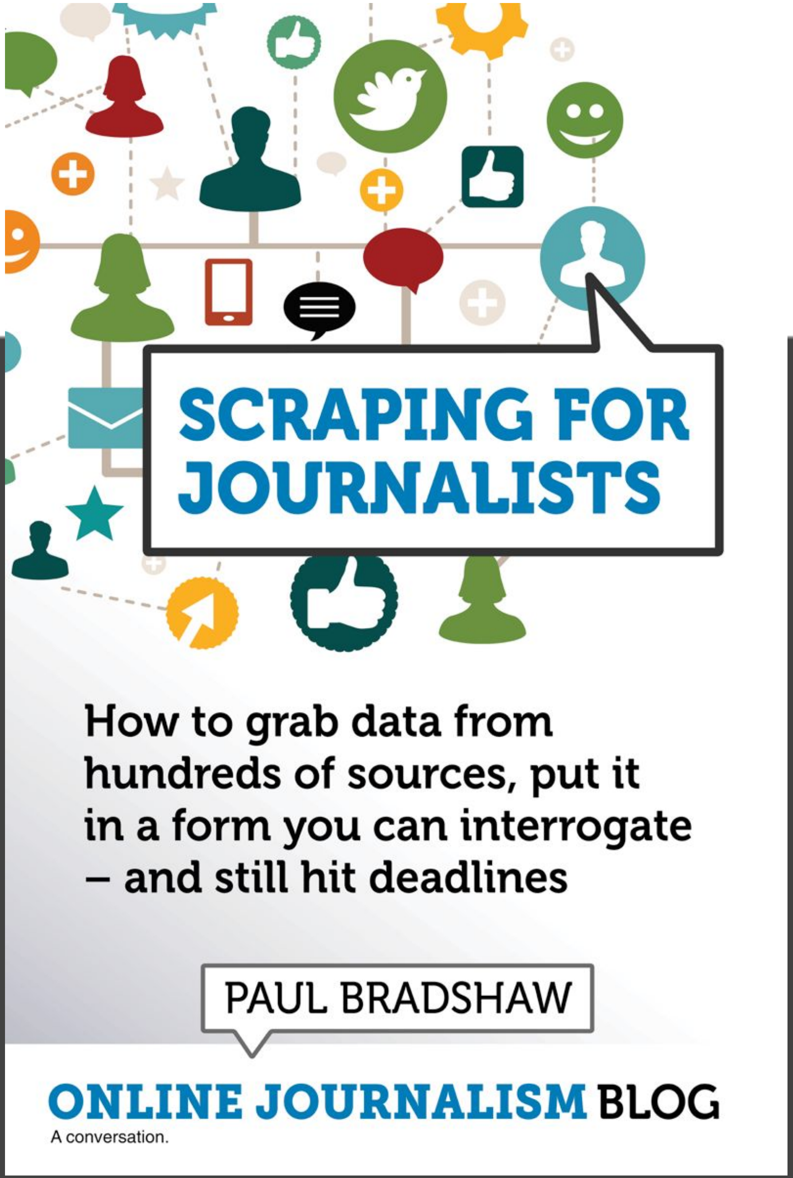
Using CSS selectors in a scraper



SCRAPING FOR JOURNALISTS

How to grab data from hundreds of sources, put it in a form you can interrogate – and still hit deadlines

PAUL BRADSHAW

ONLINE JOURNALISM BLOG

A conversation.

Paul Bradshaw

Leanpub.com/scrapingforjournalists

What we’ll cover

- What are **CSS selectors** - and why they are useful in scraping
- How to **use** CSS selectors in a Colab Python notebook, with **BeautifulSoup**

Selectors

CSS selectors

- Created so web designers could style elements, e.g. ‘make links red’
- They **‘select’ elements within HTML tags** (e.g. links, headings, images) so they can be styled
- E.g. to select anything inside a <img> tag, the selector would be simply `img`
- Also used by scraping tools like Web Scraper, Octoparse, Browse AI etc. to select content to scrape

Detour: HTML

- HTML webpages are created using HTML tags
- Most tags are like buttons, with an ‘opening’ tag turning something on (e.g. bold), and a ‘closing’ tag turning it off, e.g. **`<p> </p>`**
- Tags can have attributes and values, e.g. **`<p class=”firstpar”>`**
- ‘class’ and ‘id’ are common attributes. The *value* comes after, normally in quotes
- Tags can be **nested** within each other, e.g. a tag to make a word bold will be nested within a paragraph, nested within an article and so on

CSS selectors

- You can specify combinations of HTML tags, e.g. to select any bold text within a paragraph within a div tag:
  `div p strong`

- You can also specify attributes of those tags, such as their class or id
  `div[@class=”article”]`

- Or a combination of those
  `div[@class=”article”] p strong`

.select( )

- The Beautiful Soup library has a function that uses CSS selectors to extract information from a webpage that's been converted using the BeautifulSoup( ) function (often stored in a variable called ‘soup’) - e.g.

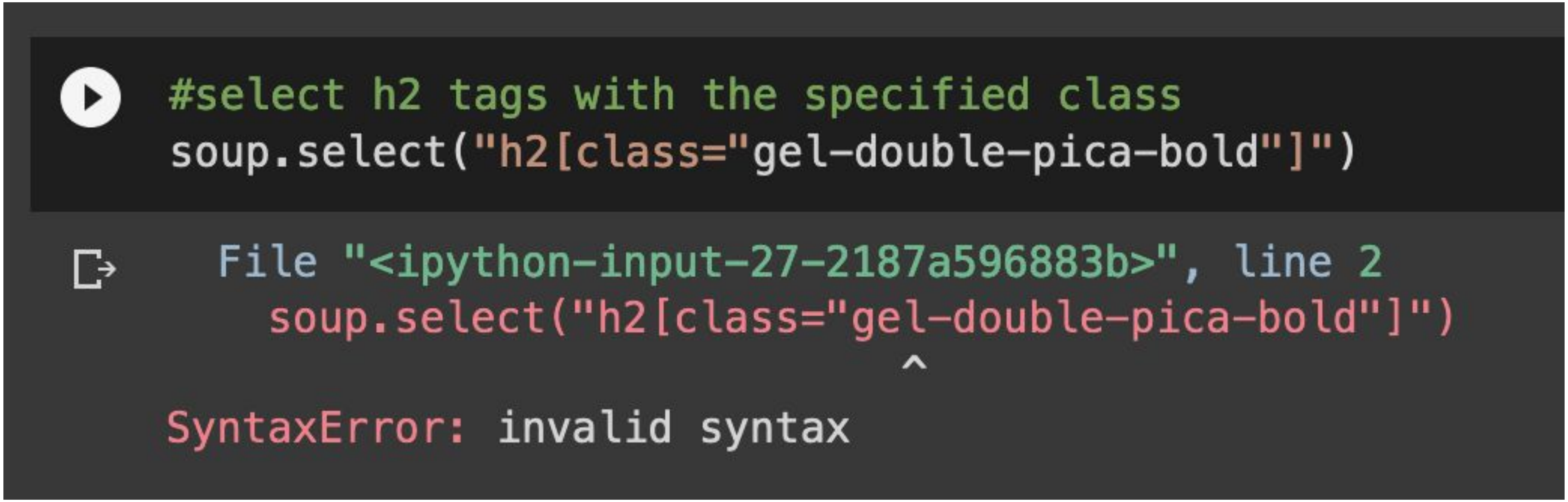
  ```
  pars = soup.select(‘p’)
  ```
- The **select( )** function is attached to **`soup`** with a period, and the selector put in parentheses
- The result is always a **list** - even if it's a list of one, or zero, results

Some examples

```
h2tags = soup.select(‘h2’)

links = soup.select(‘a’)

boldtext = soup.select(‘p b’)

nums =
soup.select(‘li[class=”number”]’)
```

⚠️ Quotes inside quotes warning!

- Note that there are two sets of quotation marks in **`(‘li[class=”number”]’)`** so they need to use different quotation marks
- The whole string is inside single quotes; so “number” needs to be inside double quotes



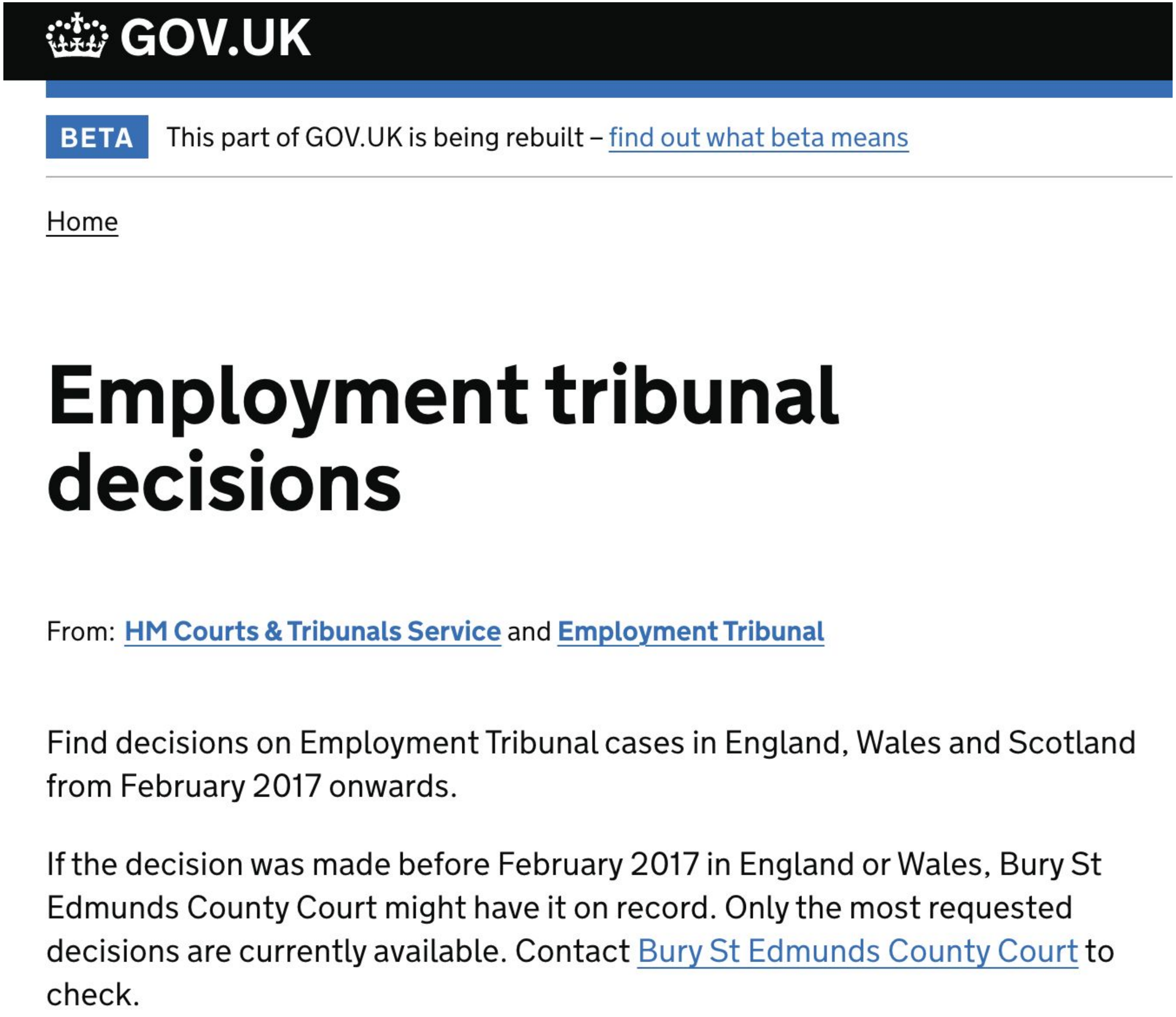
```
#select h2 tags with the specified class
soup.select("h2[class="gel-double-pica-bold"]")
```

```
  File "<ipython-input-27-2187a596883b>", line 2
    soup.select("h2[class="gel-double-pica-bold"]")
                             ^
SyntaxError: invalid syntax
```

result looks like this

```
[<h2>Accessibility links</h2>,
 <h2 class="gs-u-vh">News
Navigation</h2>]
```

Let’s apply this to a webpage…



This part of GOV.UK is being rebuilt – find out what beta means

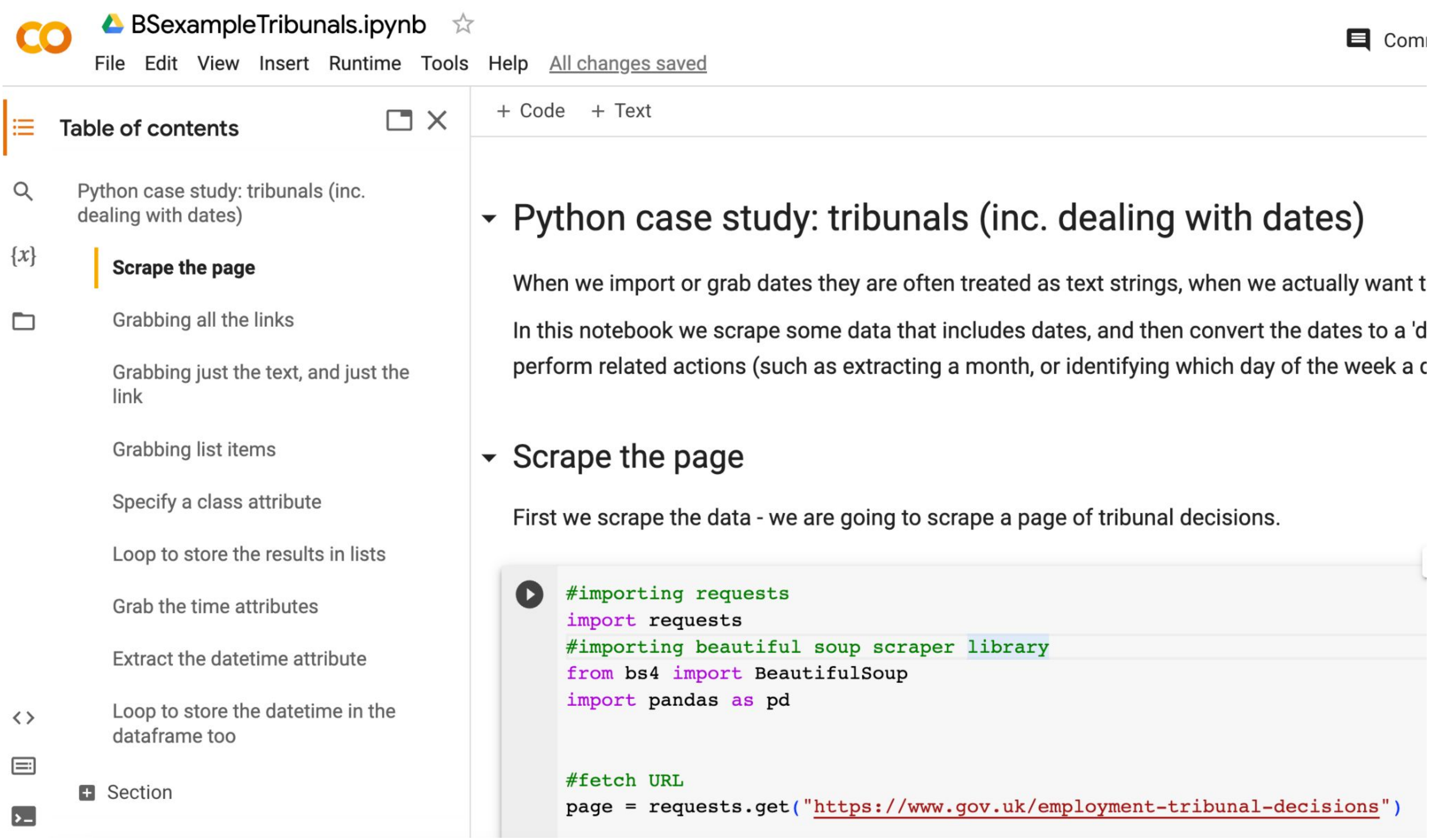
Employment tribunal decisions

From: HM Courts & Tribunals Service and Employment Tribunal

Find decisions on Employment Tribunal cases in England, Wales and Scotland from February 2017 onwards.

If the decision was made before February 2017 in England or Wales, Bury St Edmunds County Court might have it on record. Only the most requested decisions are currently available. Contact Bury St Edmunds County Court to check.

https://www.gov.uk/employment-tribunal-decisions



Python case study: tribunals (inc. dealing with dates)

When we import or grab dates they are often treated as text strings, when we actually want t

In this notebook we scrape some data that includes dates, and then convert the dates to a 'd perform related actions (such as extracting a month, or identifying which day of the week a c

Scrape the page

First we scrape the data - we are going to scrape a page of tribunal decisions.

```
#importing requests
import requests
#importing beautiful soup scraper library
from bs4 import BeautifulSoup
import pandas as pd


#fetch URL
page = requests.get("https://www.gov.uk/employment-tribunal-decisions")
```

https://colab.research.google.com/drive/1xEyJFgpRmqzgmVHiEMxmMo9jtm3cQ7vT?usp=sharing

```
import requests

from bs4 import BeautifulSoup

import pandas as pd
```

```
page =
requests.get("https://www.gov.uk/employment
-tribunal-decisions")


#turn the page content into a ‘soup’ object
soup = BeautifulSoup(page.content)
```

Reuse the code.

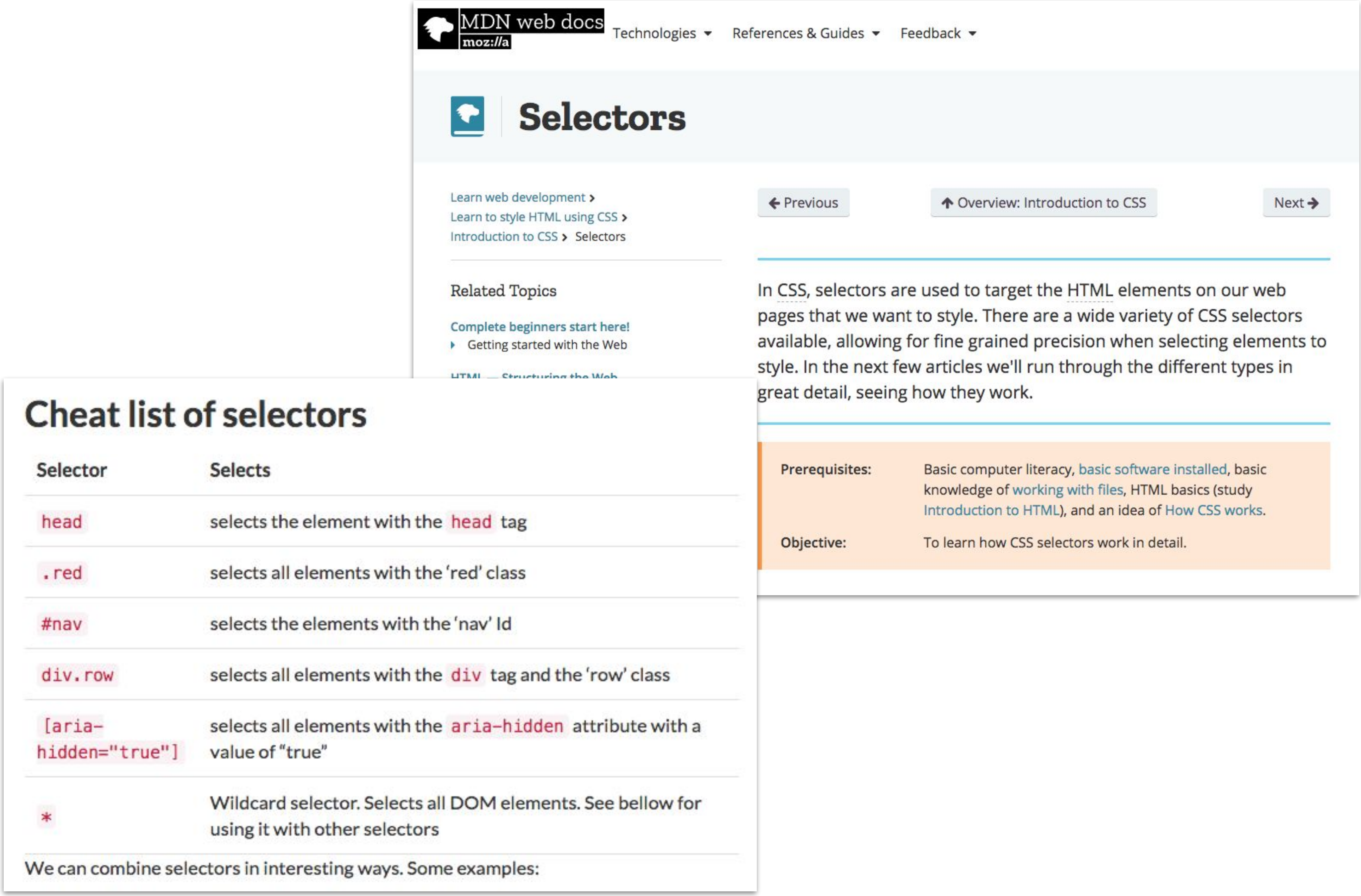
All you have to change is the URL.

```
#identify the lines in the page that I want by link and class
cases = soup.select('a')


#print the first match
print(cases[0])
```

Reuse the code.

All you have to change is the selector:

```
('a')
```



MDN web docs
moz://a
Technologies
References & Guides
Feedback

Selectors

Learn web development >
Learn to style HTML using CSS >
Introduction to CSS > Selectors
Previous
Overview: Introduction to CSS
Next
Related Topics
Complete beginners start here!
Getting started with the Web
HTML — Structuring the Web

In CSS, selectors are used to target the HTML elements on our web pages that we want to style. There are a wide variety of CSS selectors available, allowing for fine grained precision when selecting elements to style. In the next few articles we'll run through the different types in great detail, seeing how they work.

Prerequisites: Basic computer literacy, basic software installed, basic knowledge of working with files, HTML basics (study Introduction to HTML), and an idea of How CSS works.

Objective: To learn how CSS selectors work in detail.

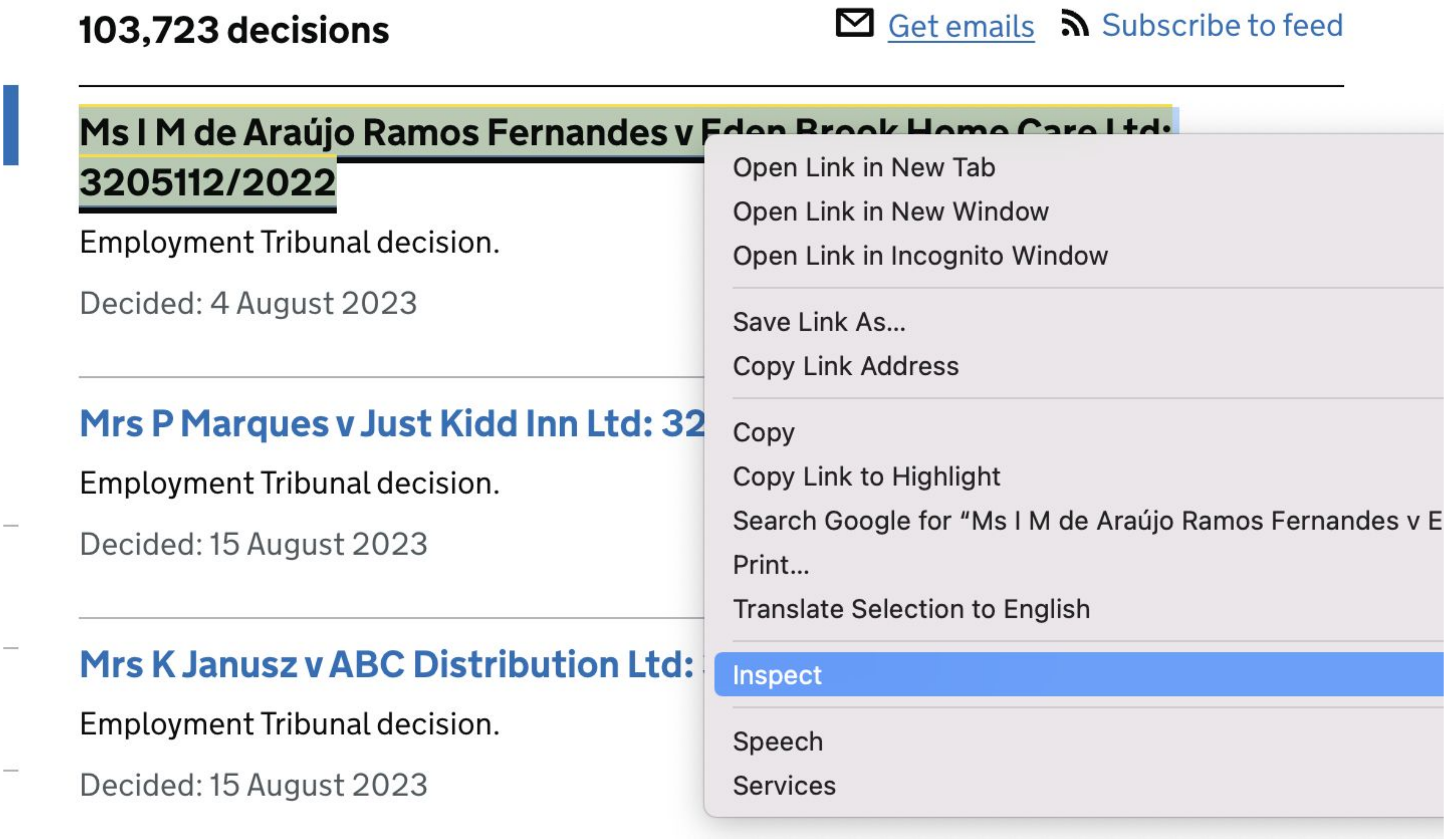
Cheat list of selectors

Selector	Selects
`head`	selects the element with the `head` tag
`.red`	selects all elements with the 'red' class
`#nav`	selects the elements with the 'nav' Id
`div.row`	selects all elements with the `div` tag and the 'row' class
`[aria-hidden="true"]`	selects all elements with the `aria-hidden` attribute with a value of "true"
`*`	Wildcard selector. Selects all DOM elements. See bellow for using it with other selectors

We can combine selectors in interesting ways. Some examples:

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Selectors
https://guide.freecodecamp.org/css/tutorials/css-selectors-cheat-sheet/

Use the inspector



103,723 decisions

Get emails Subscribe to feed

Ms I M de Araújo Ramos Fernandes v Eden Brook Home Care Ltd: 3205112/2022

Employment Tribunal decision.

Decided: 4 August 2023

Mrs P Marques v Just Kidd Inn Ltd: 32

Employment Tribunal decision.

Decided: 15 August 2023

Mrs K Janusz v ABC Distribution Ltd:

Employment Tribunal decision.

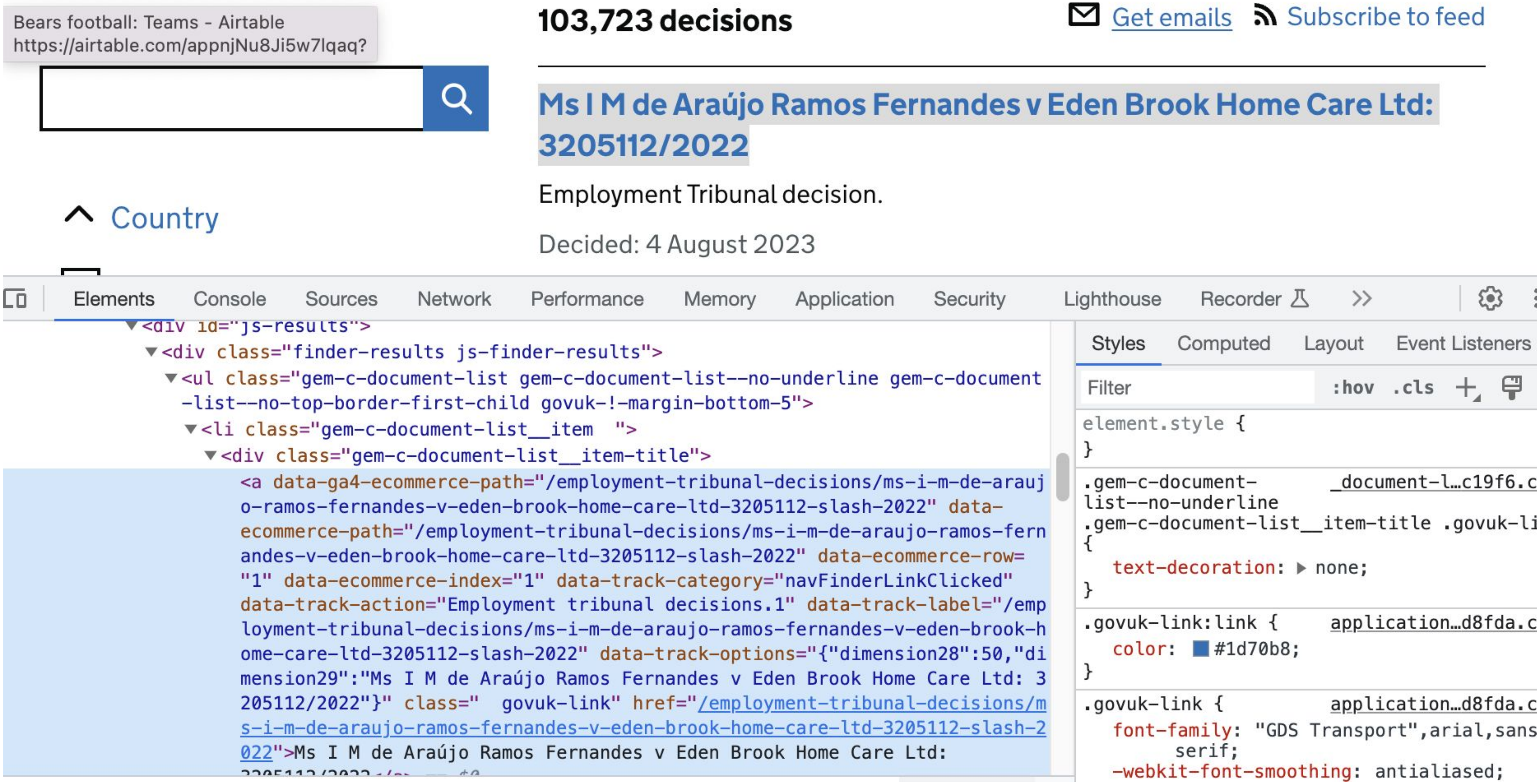
Decided: 15 August 2023

https://www.gov.uk/employment-tribunal-decisions

Using the inspector

- Use Chrome or Firefox
- Right-click on the part of the page you want to grab
- Select ‘**Inspect**’ from the menu that appears
- The **Inspector** should open across the bottom or side of the page, on the **Elements** tab
- The section of HTML that you right-clicked on should be highlighted

The inspector: Elements



Bears football: Teams - Airtable
https://airtable.com/appnjNu8Ji5w7lqaq?

103,723 decisions

Get emails

Subscribe to feed

Ms I M de Araújo Ramos Fernandes v Eden Brook Home Care Ltd: 3205112/2022

Employment Tribunal decision.

Decided: 4 August 2023

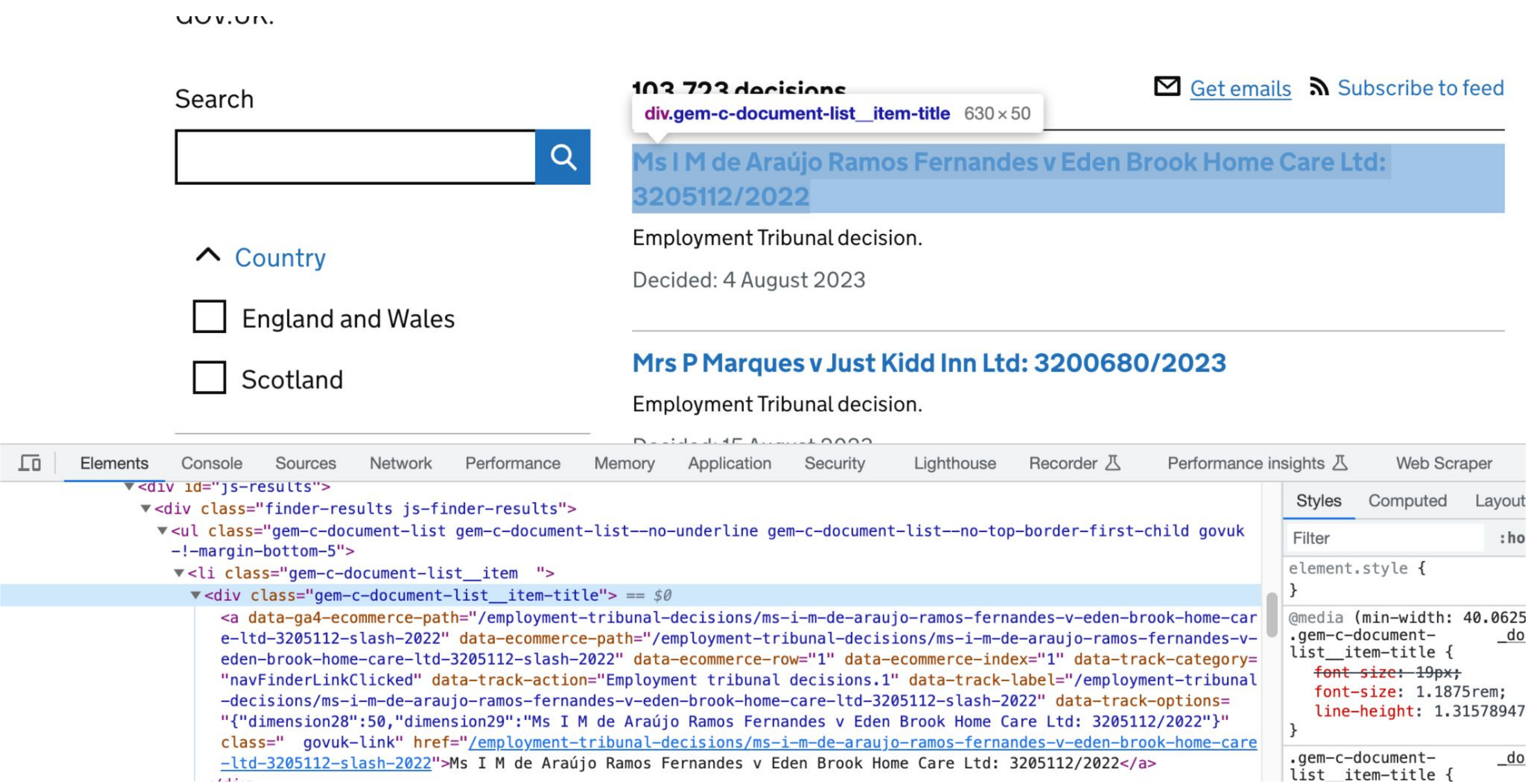
Country

```
▼<div id="js-results">
  ▼<div class="finder-results js-finder-results">
    ▼<ul class="gem-c-document-list gem-c-document-list--no-underline gem-c-document
      -list--no-top-border-first-child govuk-!-margin-bottom-5">
      ▼<li class="gem-c-document-list__item  ">
        ▼<div class="gem-c-document-list__item-title">
            <a data-ga4-ecommerce-path="/employment-tribunal-decisions/ms-i-m-de-arauj
            o-ramos-fernandes-v-eden-brook-home-care-ltd-3205112-slash-2022" data-
            ecommerce-path="/employment-tribunal-decisions/ms-i-m-de-araujo-ramos-fern
            andes-v-eden-brook-home-care-ltd-3205112-slash-2022" data-ecommerce-row=
            "1" data-ecommerce-index="1" data-track-category="navFinderLinkClicked"
            data-track-action="Employment tribunal decisions.1" data-track-label="/emp
            loyment-tribunal-decisions/ms-i-m-de-araujo-ramos-fernandes-v-eden-brook-h
            ome-care-ltd-3205112-slash-2022" data-track-options="{"dimension28":50,"di
            mension29":"Ms I M de Araújo Ramos Fernandes v Eden Brook Home Care Ltd: 3
            205112/2022"}" class="  govuk-link" href="/employment-tribunal-decisions/m
            s-i-m-de-araujo-ramos-fernandes-v-eden-brook-home-care-ltd-3205112-slash-2
            022">Ms I M de Araújo Ramos Fernandes v Eden Brook Home Care Ltd:
            3205112/2022</a> == $0
```

```
element.style {
}
.gem-c-document-                _document-l…c19f6.c
list--no-underline
.gem-c-document-list__item-title .govuk-li
{
  text-decoration: ▶ none;
}
.govuk-link:link {        application…d8fda.c
  color: ■#1d70b8;
}
.govuk-link {             application…d8fda.c
  font-family: "GDS Transport",arial,sans
      serif;
  -webkit-font-smoothing: antialiased;
```

https://www.gov.uk/employment-tribunal-decisions

Click HTML to highlight element on page



Ms I M de Araújo Ramos Fernandes v Eden Brook Home Care Ltd: 3205112/2022

Employment Tribunal decision.

Decided: 4 August 2023

Mrs P Marques v Just Kidd Inn Ltd: 3200680/2023

Employment Tribunal decision.

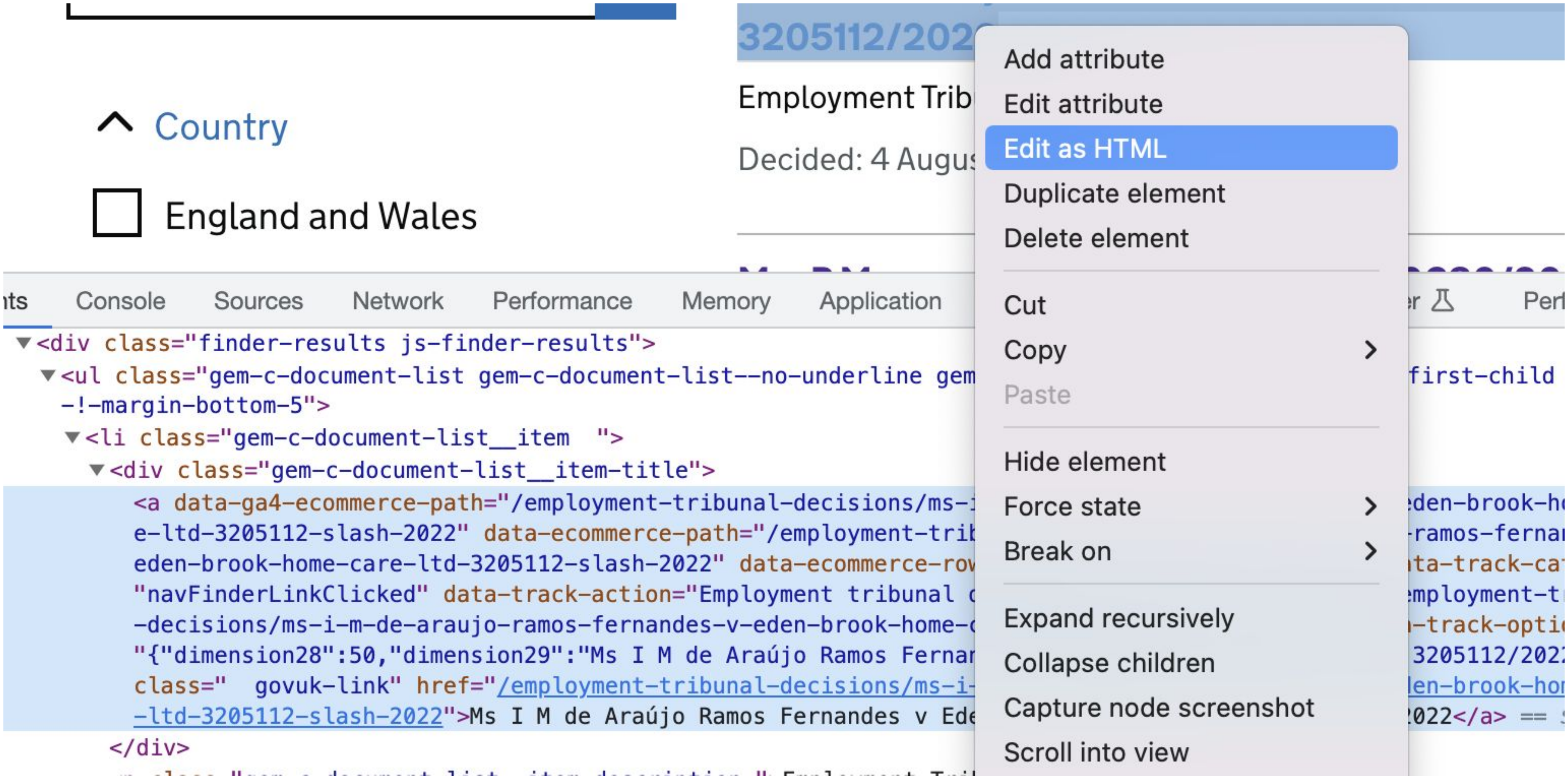
Decided: 15 August 2023

```
▼<div id="js-results">
  ▼<div class="finder-results js-finder-results">
    ▼<ul class="gem-c-document-list gem-c-document-list--no-underline gem-c-document-list--no-top-border-first-child govuk
     -!-margin-bottom-5">
      ▼<li class="gem-c-document-list__item  ">
        ▼<div class="gem-c-document-list__item-title"> == $0
           <a data-ga4-ecommerce-path="/employment-tribunal-decisions/ms-i-m-de-araujo-ramos-fernandes-v-eden-brook-home-car
           e-ltd-3205112-slash-2022" data-ecommerce-path="/employment-tribunal-decisions/ms-i-m-de-araujo-ramos-fernandes-v-
           eden-brook-home-care-ltd-3205112-slash-2022" data-ecommerce-row="1" data-ecommerce-index="1" data-track-category=
           "navFinderLinkClicked" data-track-action="Employment tribunal decisions.1" data-track-label="/employment-tribunal
           -decisions/ms-i-m-de-araujo-ramos-fernandes-v-eden-brook-home-care-ltd-3205112-slash-2022" data-track-options=
           "{"dimension28":50,"dimension29":"Ms I M de Araújo Ramos Fernandes v Eden Brook Home Care Ltd: 3205112/2022"}"
           class="  govuk-link" href="/employment-tribunal-decisions/ms-i-m-de-araujo-ramos-fernandes-v-eden-brook-home-care
           -ltd-3205112-slash-2022">Ms I M de Araújo Ramos Fernandes v Eden Brook Home Care Ltd: 3205112/2022</a>
```

```
element.style {
}
@media (min-width: 40.0625
.gem-c-document-        _do
list__item-title {
  ~~font-size: 19px;~~
  font-size: 1.1875rem;
  line-height: 1.31578947
}
.gem-c-document-        _do
list__item-title {
```

https://www.gov.uk/employment-tribunal-decisions

Right-click on HTML to edit (copy) it

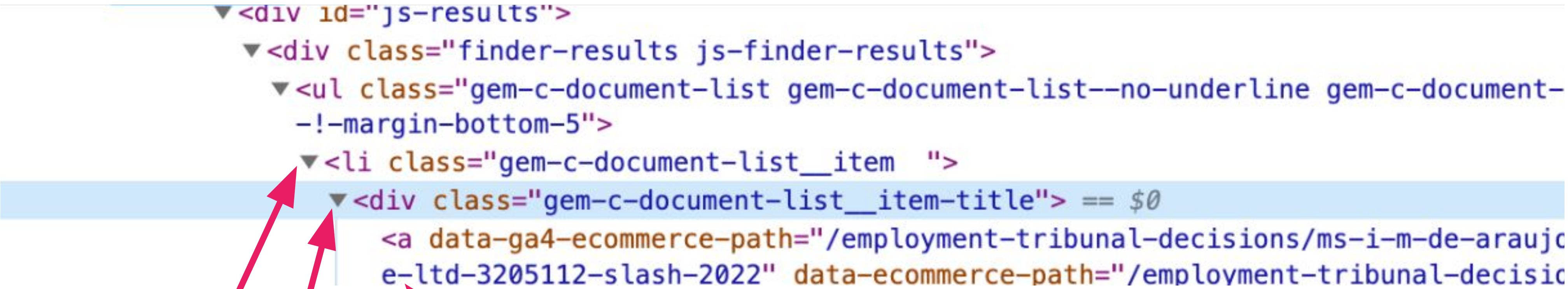


```
▼<div class="finder-results js-finder-results">
  ▼<ul class="gem-c-document-list gem-c-document-list--no-underline gem                    first-child
    -!-margin-bottom-5">
    ▼<li class="gem-c-document-list__item  ">
      ▼<div class="gem-c-document-list__item-title">
          <a data-ga4-ecommerce-path="/employment-tribunal-decisions/ms-i                    eden-brook-h
          e-ltd-3205112-slash-2022" data-ecommerce-path="/employment-trib                    -ramos-ferna
          eden-brook-home-care-ltd-3205112-slash-2022" data-ecommerce-row                    ta-track-ca
          "navFinderLinkClicked" data-track-action="Employment tribunal d                    mployment-t
          -decisions/ms-i-m-de-araujo-ramos-fernandes-v-eden-brook-home-c                    -track-opti
          "{"dimension28":50,"dimension29":"Ms I M de Araújo Ramos Fernar                    3205112/202
          class="  govuk-link" href="/employment-tribunal-decisions/ms-i-                    len-brook-ho
          -ltd-3205112-slash-2022">Ms I M de Araújo Ramos Fernandes v Ede                    022</a> ==
        </div>
```

https://www.gov.uk/employment-tribunal-decisions

```html
<a

data-ga4-ecommerce-path="/employment-tribunal-decisions/mrs-p-marques-v-just-kidd-inn-ltd-3200680-slash-2023"

data-ecommerce-path="/employment-tribunal-decisions/mrs-p-marques-v-just-kidd-inn-ltd-3200680-slash-2023"

data-ecommerce-row="1"

data-ecommerce-index="2"

data-track-category="navFinderLinkClicked"

data-track-action="Employment tribunal decisions.2"

data-track-label="/employment-tribunal-decisions/mrs-p-marques-v-just-kidd-inn-ltd-3200680-slash-2023"

data-track-options="{&quot;dimension28&quot;:50,&quot;dimension29&quot;:&quot;Mrs P Marques v Just Kidd Inn Ltd: 3200680/2023&quot;}"

class="  govuk-link"

href="/employment-tribunal-decisions/mrs-p-marques-v-just-kidd-inn-ltd-3200680-slash-2023">

Mrs P Marques v Just Kidd Inn Ltd: 3200680/2023</a>
```

See parent-child structure



```
▼<div id="js-results">
  ▼<div class="finder-results js-finder-results">
    ▼<ul class="gem-c-document-list gem-c-document-list--no-underline gem-c-document-
     -!-margin-bottom-5">
      ▼<li class="gem-c-document-list__item  ">
        ▼<div class="gem-c-document-list__item-title"> == $0
           <a data-ga4-ecommerce-path="/employment-tribunal-decisions/ms-i-m-de-araujc
           e-ltd-3205112-slash-2022" data-ecommerce-path="/employment-tribunal-decisic
```

<a ...> is indented so is the child of the element above

<div> is a parent of <a ...>

<li> is a parent of <div>
<ul> is a parent of <li>

https://www.gov.uk/employment-tribunal-decisions

```
<div

class="gem-c-document-list__item-title">
```

```
#this grabs the links to each case
cases = soup.select('div[class="gem-c-document-list__item-title"]')

#check the first
print(cases[0])

#show just the text
print(cases[0].get_text())
```

Recap

- Use .select( ) to drill down into the object ‘soup’

```
tels = soup.select('SELECTOR HERE')
```

- Always generates a list - access items using a *for* loop or an index/indices
- Add **.get_text( )** to an item to extract text

```
cases[0].get_text()
for i in cases:
  print(i.get_text())
```

Try it now:

- In your notebook scrape the page and extract the contents of:
  - <div> tags with class="gem-c-document-list__item-title"
  - <time> tags
- Loop through those tags and print the `.get_text()`
- Access the first match and print the `.get_text()`