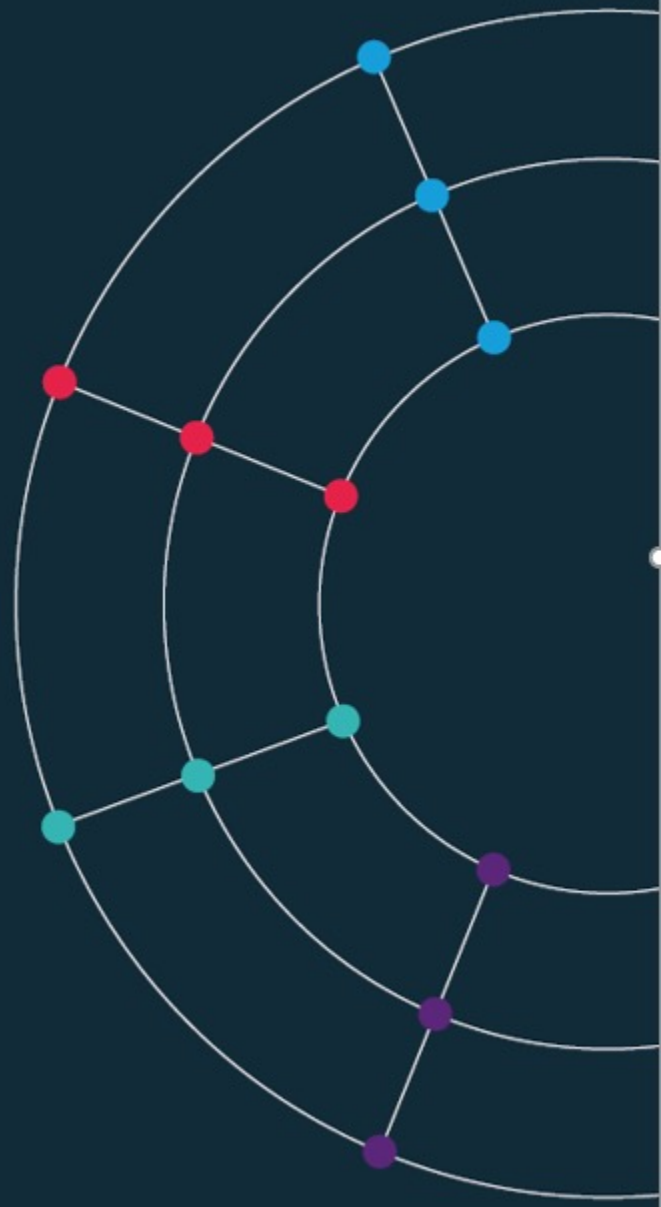
Session 5.

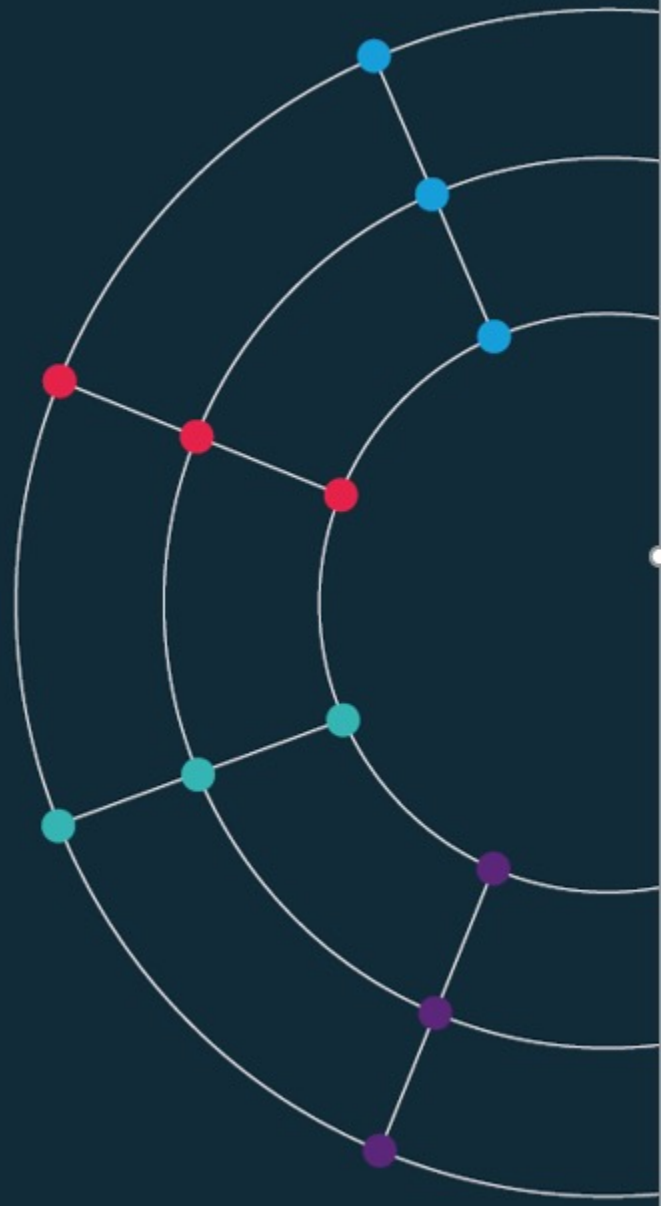
Data scraping



Session 5.

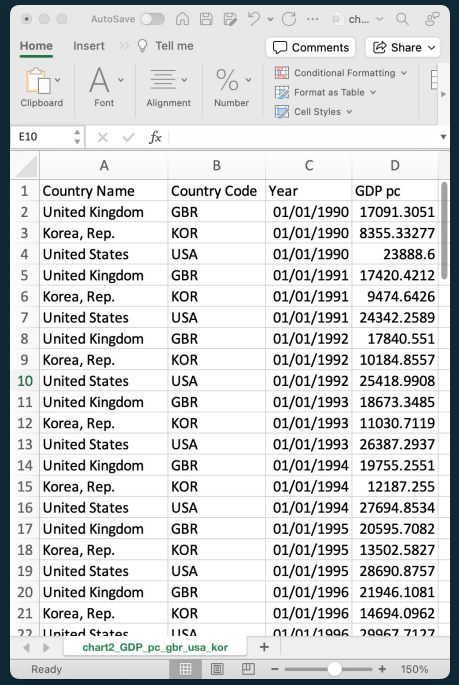
Data scraping

Scraping the HTML source



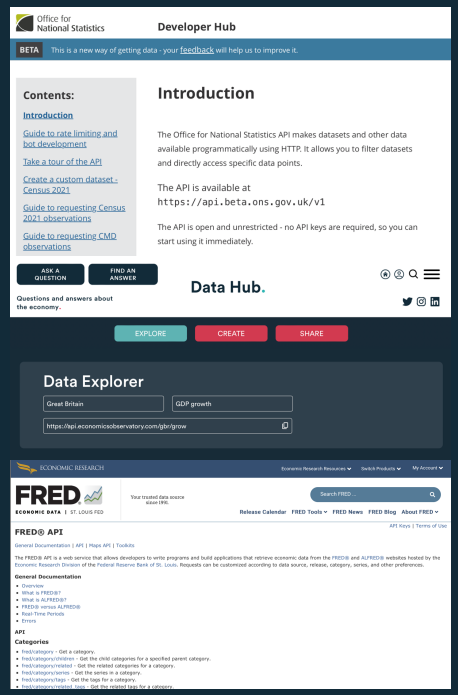
Data, so far...

Today we have used data from



	A	B	C	D
1	Country Name	Country Code	Year	GDP pc
2	United Kingdom	GBR	01/01/1990	17091.3051
3	Korea, Rep.	KOR	01/01/1990	8355.33277
4	United States	USA	01/01/1990	23888.6
5	United Kingdom	GBR	01/01/1991	17420.4212
6	Korea, Rep.	KOR	01/01/1991	9474.6426
7	United States	USA	01/01/1991	24342.2589
8	United Kingdom	GBR	01/01/1992	17840.551
9	Korea, Rep.	KOR	01/01/1992	10184.8557
10	United States	USA	01/01/1992	25418.9908
11	United Kingdom	GBR	01/01/1993	18673.3485
12	Korea, Rep.	KOR	01/01/1993	11030.7119
13	United States	USA	01/01/1993	26387.2937
14	United Kingdom	GBR	01/01/1994	19755.2551
15	Korea, Rep.	KOR	01/01/1994	12187.255
16	United States	USA	01/01/1994	27694.8534
17	United Kingdom	GBR	01/01/1995	20595.7082
18	Korea, Rep.	KOR	01/01/1995	13502.5827
19	United States	USA	01/01/1995	28690.8757
20	United Kingdom	GBR	01/01/1996	21946.1081
21	Korea, Rep.	KOR	01/01/1996	14694.0962
22	United States	USA	01/01/1996	29967.7127

Structured Files
(e.g. Excel, CSV, JSON)

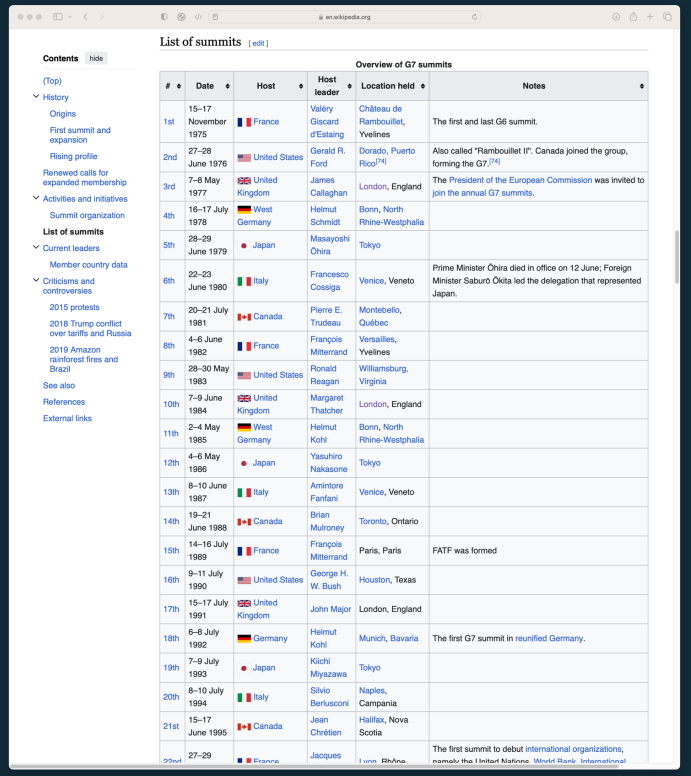


APIs
(e.g. ONS, ECO, FRED)

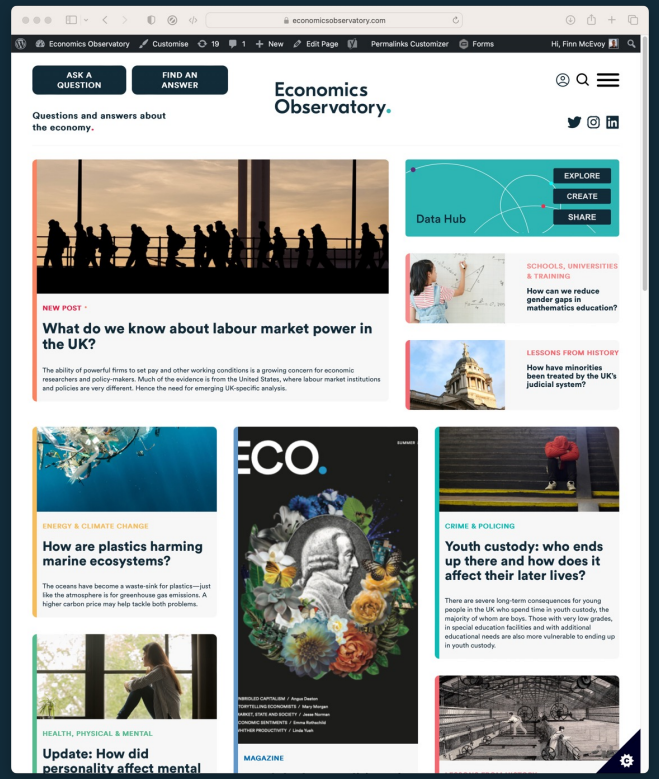
But what do we do when the data we want isn't available?

Data, so far...

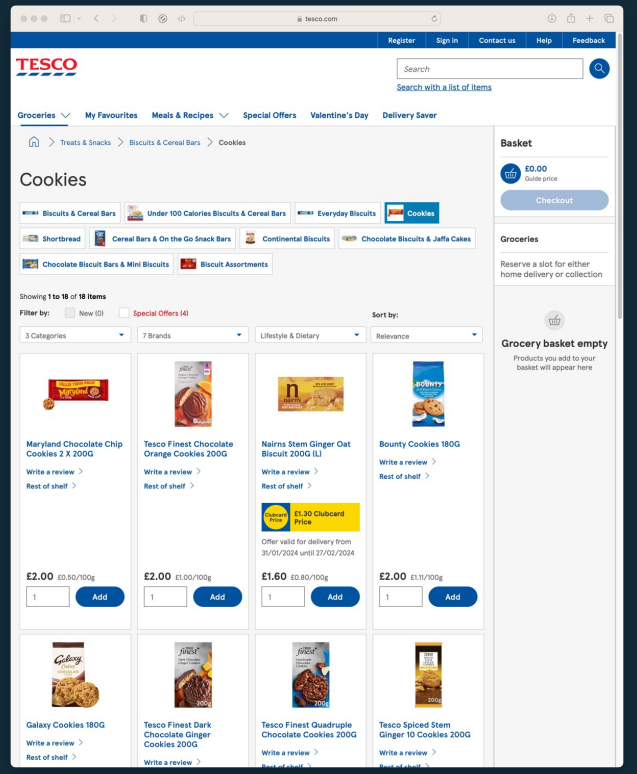
What if we want



Data from Wikipedia



News and Media



Prices from Supermarkets

Scraping.

- The automated extraction of data from websites
- Scraping the HTML source. Easy. Automated.
- Scraping static HTML pages.
  - A bit more difficult. Can be automated.
  - CAPTCHA. Impersonating a human user. Zombie browser.
- Scraping HTML pages generated on-the-fly with JavaScript.
  - Hard. Only zombie browser works, and only in some cases.

hiQ labs vs. LinkedIn, 2019, US Court of Appeals for the Ninth Circuit, 17-16783
LinkedIn vs. hiQ labs, 2021, US Supreme Court, 19-1116

Parsing HTML.

We started today by writing HTML

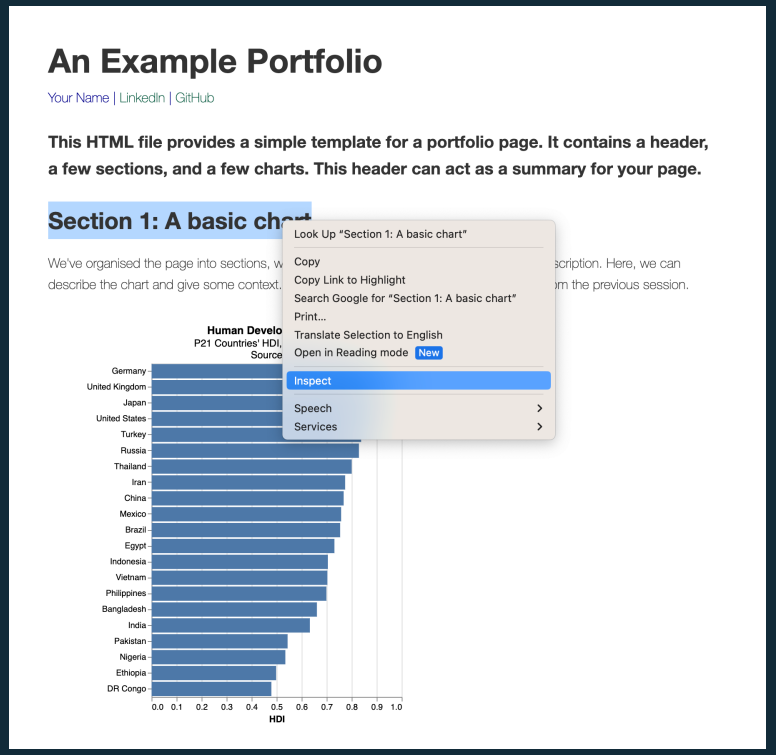
```
<html>
<head>
    <title>My Portfolio</title>
</head>
<body>
    <h1> My Portfolio</h1>
    <p> A collection of masterclass-work </p>
</body>
</html>

```

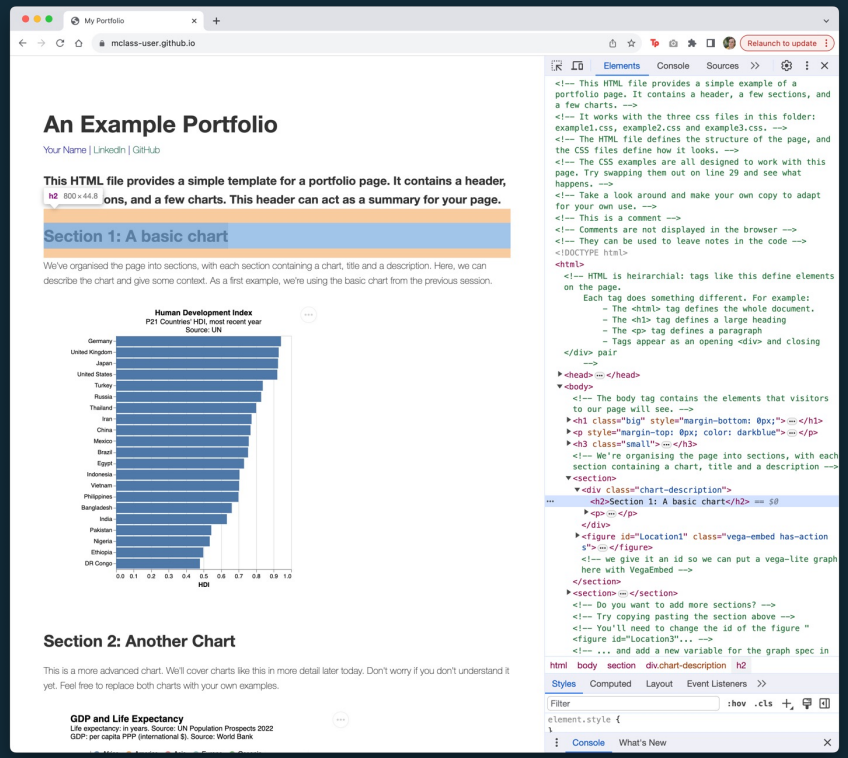
```
<a href="https://www.economicsobservatory.com/
could-a-new-policy-institution-help-solve-the-uks-productivity-probl
em">
    <div>
        <h3 class="home__blocks-item-title">Could a new policy
        institution help solve the UK's productivity problem?
        </h3>
        <div class="home__blocks-item-teaser display">
            <p>Comparatively sluggish productivity growth is one of
            the UK's biggest policy challenges. Past strategies
                to solve the problem have lacked commitment and
                proper evaluation. One solution could be to
                establish a
                growth and productivity institution to coordinate
                policies over the long term.</p>
        </div>
    </div>
</a>
```

We'll end it by reading HTML

Using Inspect-Element.

Our most important tool is ‘inspect-element’





In Chrome, Right-click or Ctrl. Click and select ‘Inspect’

Parsing HTML.

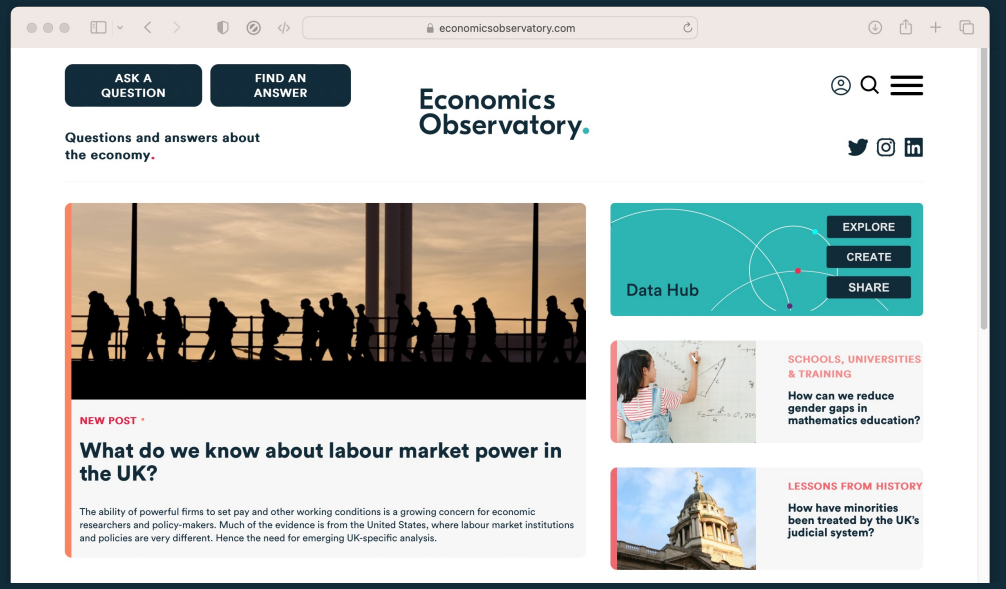
- (Almost) all the data displayed on websites is found in the HTML
- We can extract data by searching the HTML
- Everything is defined in the HTML, we just have to find it

Overview.

1. Determine how data is defined in HTML (inspect element)
2. Parse the HTML (Beautiful Soup)
3. Clean the data (Pandas)
4. Visualise (Vega-lite)

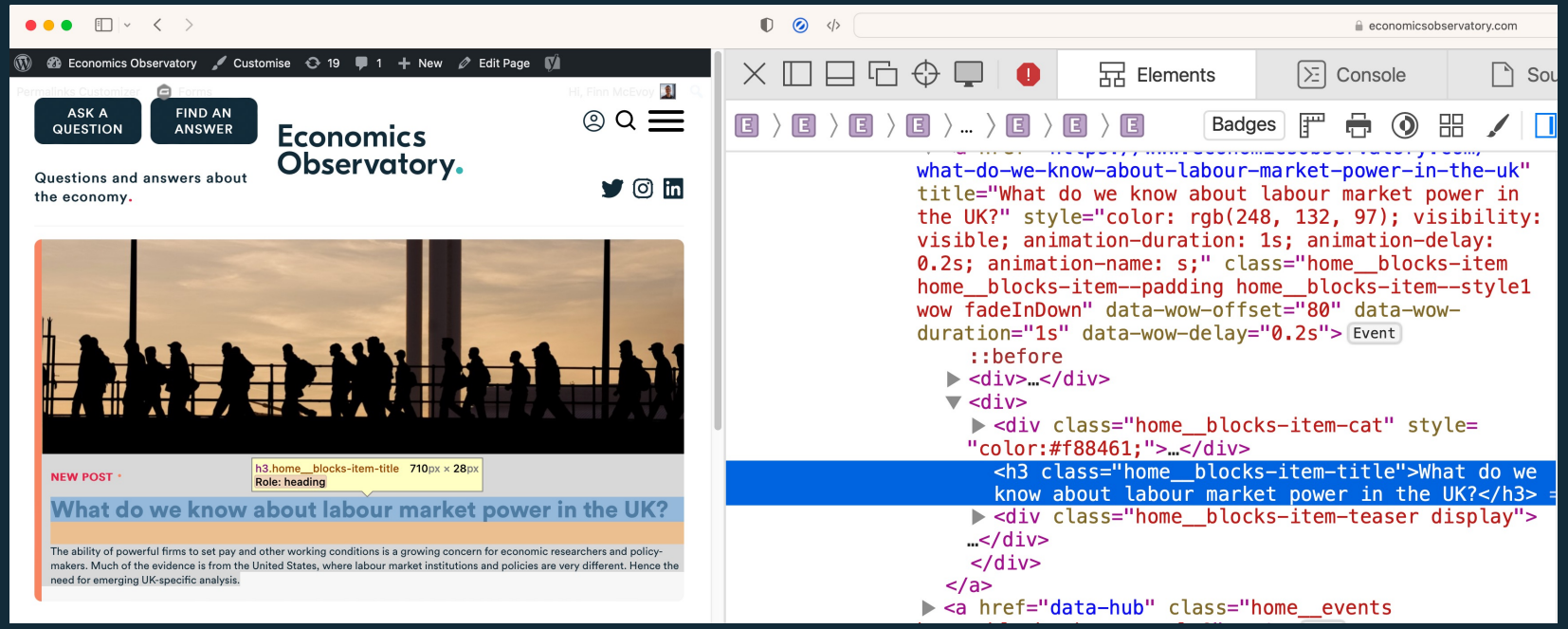
An Example.

For example, we can scrape ECO headlines and tag-lines by ‘parsing’ the HTML



Using Inspect-Element.

We determine how the target data is defined using ‘inspect-element’



We see titles have a class “home_blocks-item-title”

Parse the HTML

We'll use use a Python module, BeautifulSoup, to interpret the HTML.

For example, we can look for every title by searching for:

class "home_blocks-item-title"

with

```
soup.find_all(class_="home__blocks-item-title")
```

Code-along.

A more advanced scraper

In this bonus practical session, we will use Google Colab to use scrape data from the Economics Observatory website using Python. Again, we can also embed a chart displaying the scraped data into your website, using VS Code and GitHub.

- Work through the following guided notebook: “s5_Scraping.ipynb” (open in Google Colab)

Learn more.

- In this session we have tried basic data scraping with BeautifulSoup
- There's still much more to learn
  - Choose your own projects (how can you make your job easier?)
  - Try bigger projects (scrape 100 pages, not just 1)
  - Try advanced tools (e.g. Selenium)

Learn more, responsibly.

- **Rate-limiting**: avoid making too many requests at once
- **Ethics**: Ensure your scraping activities do not harm the website's operation
- **Data Privacy**: Be mindful of personal data collection. Comply with relevant data protection laws (like GDPR).