Digital Technologies and Value Creation

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

www.bayes.city.ac.uk

Overview – subject to change

Overarching theme	Week	
Introduction	1	Introduction to analytics applications & coding basics
Gathering data	2	Scraping static web content with BeautifulSoup
Gathering data	3	Scraping dynamic web content with Selenium & other advanced tools
Gathering data / descriptive analytics	4	Using social media APIs & descriptives in marketing analytics
Gathering data / descriptive analytics	5	Data pre-processing & descriptive analytics
NO LECTURE	6	NO LECTURE
Descriptive analytics	7	A look at NLP & descriptives in people analytics
Predictive analytics	8	Retaining employees and customers with classification
Predictive analytics	9	Time series analysis & valuing a (social media) customer base
Predictive analytics	10	Segmenting customers and positioning products
Prescriptive analytics	11	Optimizing products and organizations

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Learning objectives of today

Goals: Learn to use some of the key tools used for web scraping in Python

- Which tools should be used for what type of project?
- How tor read information from an HTML?

How will we do this?

- We start by attempting to understand an HTML document. Don’t worry, we won’t be creating websites – we just want to know what makes up a website
- We then use BeuatifulSoup and Requests to gather information from simple websites

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

An overview of scraping

Scraping, crawling, and spiders (don’t worry, they don’t have 8 legs)

(Web) scraping:

- Take any publicly available data and import it (or statistics based on it) to your hardware
- Web scraping (vs. data scraping): data comes from the web

$\Downarrow$

- Taking and downloading data
- Can be adjusted manually

Web crawling:

- Like web scraping, but we keep going from page to page
- → Finding and following links

$\Downarrow$

- Process that goes through targets and finds new ones
- Needs a “bot” for scraping, due to volume → web crawler, crawler, spider, or spiderbot

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

An overview of the technologies we will use (Scrapy and Selenium are optional!)

BeautifulSoup

- Get specific elements from web pages
- Simplifies web scraping
- Additional tools needed to retrieve pages (e.g., Requests)

- Simple to get started
- One-off scraping tasks

Se

- Automates browsers
- Originally a testing suite for running applications on different browsers

- Navigate complex page behavior
- One-off, speed not important

Scrapy

- Create spiders (asynchronous, fast)
- Inbuilt functionality to deal with crawling/scraping difficulties
- Easily adjust use of server resources

- Complex scraping tasks
- Build once, run frequently

- “Render” JavaScript used for dynamic websites

- Deal with JavaScript in Scrapy projects

The legal parts

- The legal situation of scraping can be complicated
- What is never allowed is to republish copyrighted information (unless there is some license granting you the right to, but this usually involves citation)
- It is generally forbidden to violate terms of service
    - When you create an account, you explicitly agree to the terms of service of a site
    - Even without an account, you implicitly agree
    - And even when not illegal, providers may block your account or IP!
- A good first place to check: robots.txt
- Generally, be reasonable: even if there is no problem scraping a site in principle, remember that server resources are finite!

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Understanding Websites

What’s in a website?

- Each time the user clicks on a link, the browser makes a connection to the web server and issues a “GET” request
  - This signals that the browser wants to GET the document at the specified URL

- The server returns the document to the browser, which formats and displays the document to the user

- Most commonly, documents are created in HTML

Hypertext Markup Language (HTML)

- A Markup language: text with ”tags”, which control structure and format
- Hypertext: text with references (hyperlinks) to other text
- HTML: standard language to create documents intended for web browsers

An HTML document

Opening Tag

Closing Tag

Text Content

Attribute

```
<html>
  <head>
    <title>A webpage</title>
  </head>
  <body>
  <a href=”www.website.com">Click here!</a>
  </body>
</html>
```

HTML in practice

Scraping for People Analytics with BeautifulSoup

Requests and BeautifulSoup

- Each time the user clicks on a link (in HTML: an anchor tag with an href= value), the browser makes a connection to the web server and issues a “GET” request
  - This signals that the browser wants to GET the document at the specified URL
- The server returns the (HTML) document to the browser, which formats and displays the document to the user
- We can create a GET request using the requests package (without having to worry too much about the details), and return the relevant HTML document
- BeautifulSoup allows us to parse through the returned HTML more easily

BeautifulSoup in pratice

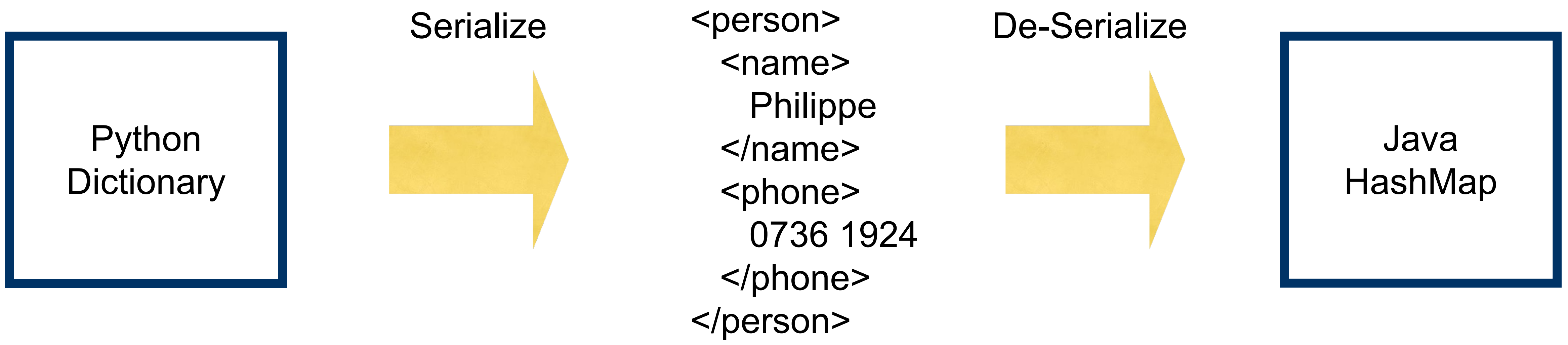
Other formats to exchange data online

Communicating data between computers

- Given the usefulness of HTTP(S) to transfer HTML document, there was a natural move toward exchanging data between programs using these protocols

  → need agreed way to represent data going between applications and across networks
- There are two commonly used formats: XML and JSON

Sending data across the web – XML



```
<person>
   <name>
      Philippe
   </name>
   <phone>
      0736 1924
   </phone>
</person>
```

In more detail: eXtensible Markup Language

- Primary purpose: help information systems share structured data
- Started as simplified subset of the Standard Generalized Markup Language (SGML)
- Designed to be relatively human-legible

Start Tag

End Tag

Text Content

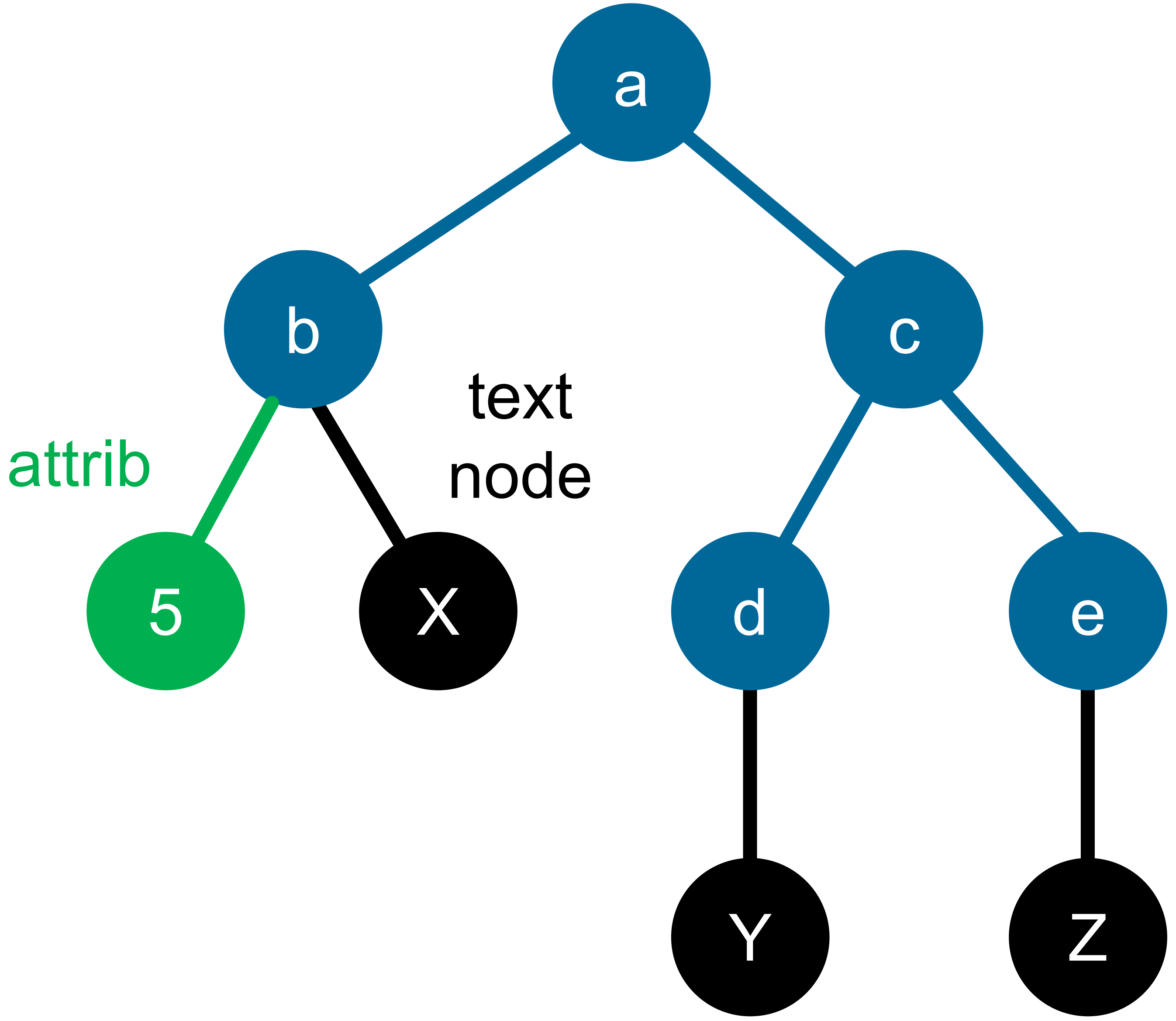
Attribute

Self Closing Tag

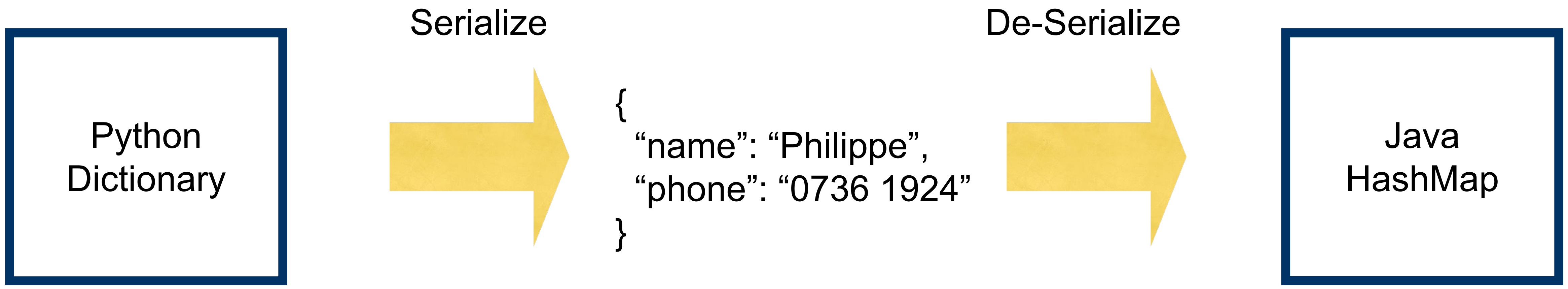
```
<person>
  <name>Philippe</name>
  <phone type="intl">
    +44 736 1924
   </phone>
   <email hide="yes" />
</person>
```

XML as a tree or a path

```
<a>
  <b w="5">X</b>
  <c>
    <d>Y</d>
    <e>Z</e>
  </c>
</a>
```



Sending data across the web – JSON



```
{
  “name”: “Philippe”,
  “phone”: “0736 1924”
}
```

In more detail: JavaScript Object Notation (JSON)

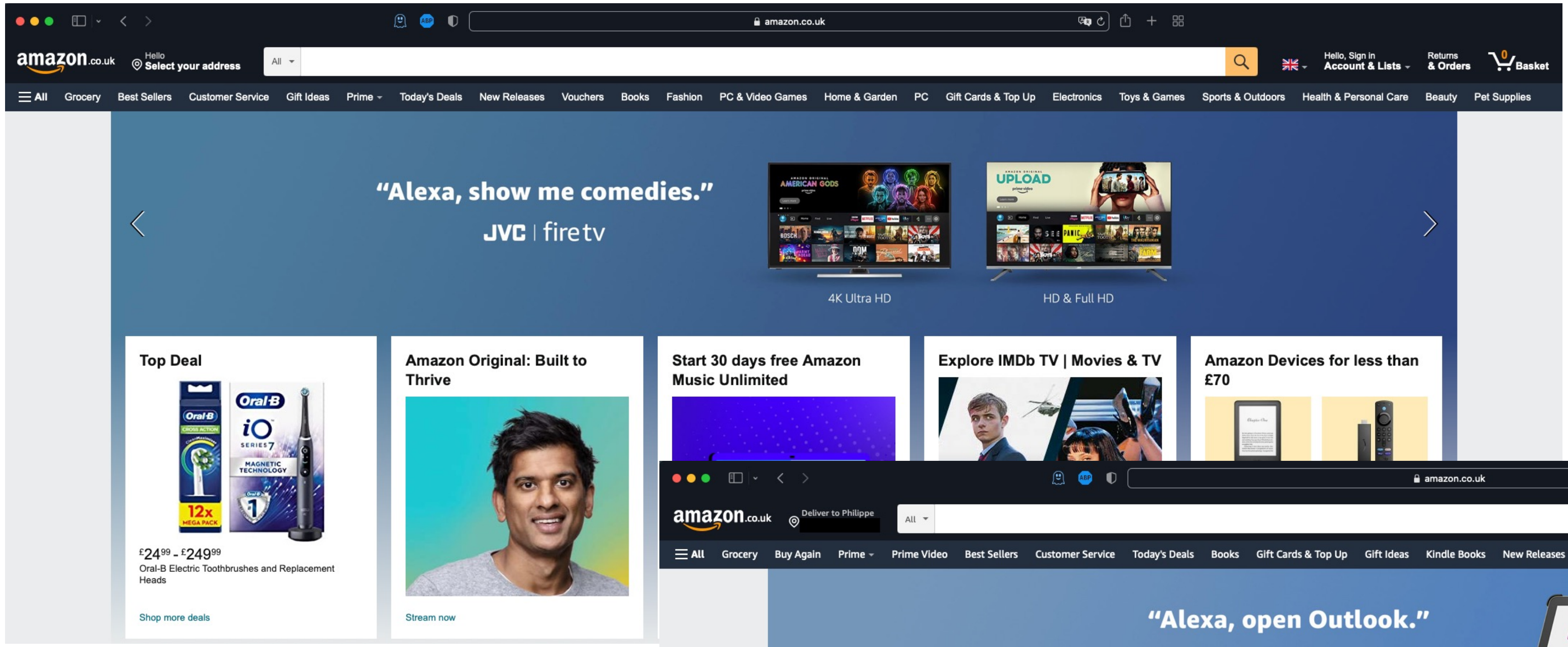
```
{
    “name” : “Philippe”,
    “phone” : {
        “type” : “intl”,
        “number” : “+44 736 1924”
    },
    “email” : {
        “hide” : “yes”
    }
}
```

An Initial Look at Dynamic Content

Dynamic content

- Recall that XML documents can be treated as a tree
- The same is true for HTML → the Document Object Model (DOM) does just that
- Many web pages require your browser to run JavaScript to query the DOM and also modify it
  - E.g., you haven't accepted cookies yet and the content won't be loaded until you do
  - E.g., you tick something or click a button and the page changes – the web server will not have a completely specified HTML document for each combination of clicks!
  - E.g., an online shop displays different items to you, depending on your past behavior – no shop stores an HTML document for each of its users!

Dynamic content – an example



amazon.co.uk

amazon.co.uk Hello Select your address All

Hello, Sign in Account & Lists Returns & Orders 0 Basket

All Grocery Best Sellers Customer Service Gift Ideas Prime Today's Deals New Releases Vouchers Books Fashion PC & Video Games Home & Garden PC Gift Cards & Top Up Electronics Toys & Games Sports & Outdoors Health & Personal Care Beauty Pet Supplies

"Alexa, show me comedies."

JVC | fire tv

4K Ultra HD

HD & Full HD

Top Deal

£24.99 - £249.99

Oral-B Electric Toothbrushes and Replacement Heads

Shop more deals

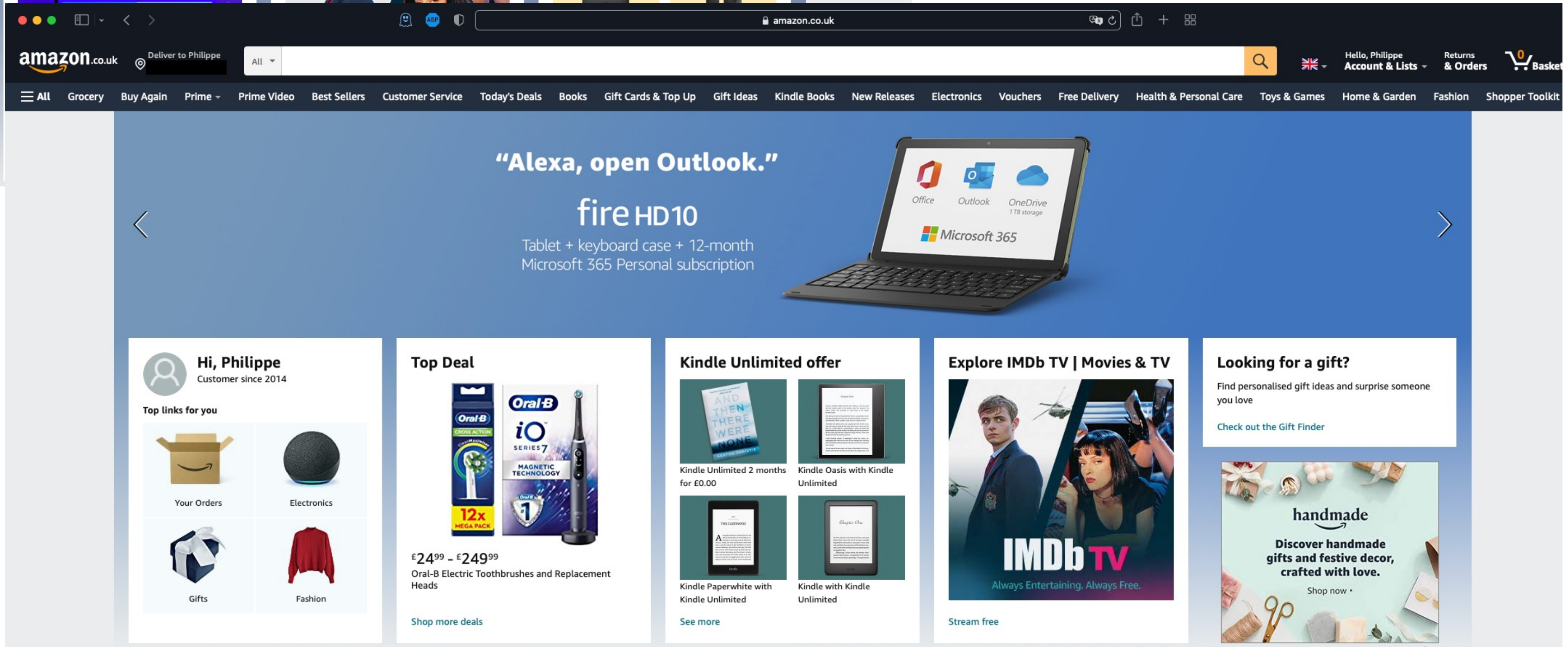
Amazon Original: Built to Thrive

Stream now

Start 30 days free Amazon Music Unlimited

Explore IMDb TV | Movies & TV

Amazon Devices for less than £70



amazon.co.uk

amazon.co.uk Deliver to Philippe [redacted] All

Hello, Philippe Account & Lists Returns & Orders 0 Basket

All Grocery Buy Again Prime Prime Video Best Sellers Customer Service Today's Deals Books Gift Cards & Top Up Gift Ideas Kindle Books New Releases Electronics Vouchers Free Delivery Health & Personal Care Toys & Games Home & Garden Fashion Shopper Toolkit

"Alexa, open Outlook."

fire HD10

Tablet + keyboard case + 12-month Microsoft 365 Personal subscription

Office Outlook OneDrive

Microsoft 365

Hi, Philippe

Customer since 2014

Top links for you

Your Orders Electronics Gifts Fashion

Top Deal

£24.99 - £249.99

Oral-B Electric Toothbrushes and Replacement Heads

Shop more deals

Kindle Unlimited offer

Kindle Unlimited 2 months for £0.00

Kindle Oasis with Kindle Unlimited

Kindle Paperwhite with Kindle Unlimited

Kindle with Kindle Unlimited

See more

Explore IMDb TV | Movies & TV

IMDb TV Always Entertaining. Always Free.

Stream free

Looking for a gift?

Find personalised gift ideas and surprise someone you love

Check out the Gift Finder

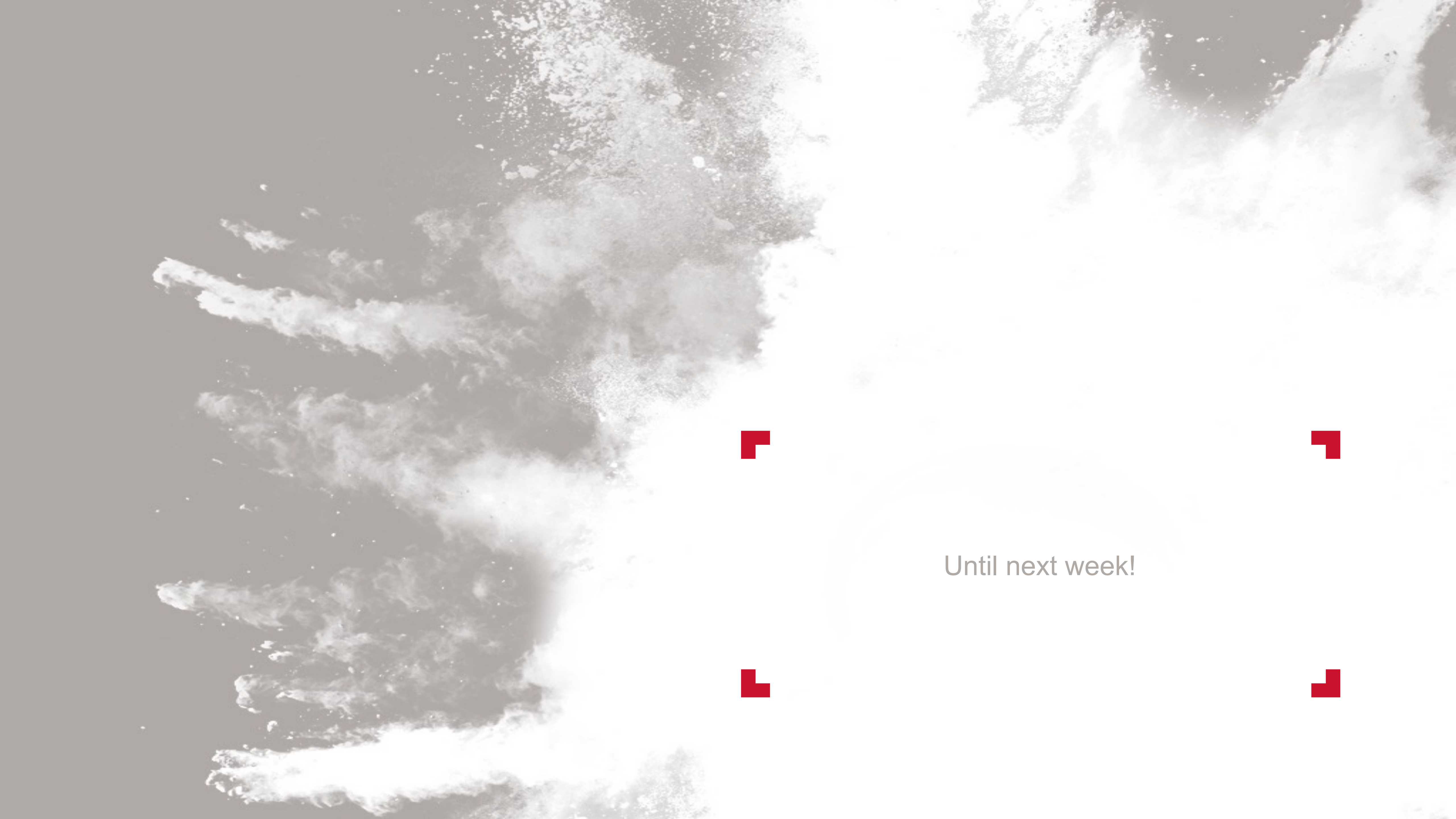
handmade Discover handmade gifts and festive decor, crafted with love. Shop now

The problem with scraping dynamic web sites

- When making a request, the return value doesn’t specify the full HTML document – what we get back in Python will look different than the HTML we inspect
- To extract information based on navigating an HTML, we first need to render the website (as if we are actually using a browser to look at it)
- One way to do this is to simulate a browser

Dynamic websites in practice

python™



Until next week!