BRIGHT CAPE

SHOW AND APPLY THE POWER OF DATA

Training in Web-scraping/Web-automation

01/04/2019

Contents

What are we going to do today?

Contents

What are we going to do today?

Overview of the course

What skills you will learn during this course and how we will apply them

Main Goal: Be able to build your own web-scraping solution

At the end of this course you should have a basic understanding of how the internet works, how you can use tools like R or Python in combination with Selenium to automate a web browser and how you automatically extract large quantities of data from the internet

Instruction

- During this presentation I will explain what tools you will need to effectively implement web-scraping, how to set up your device, and explain some of the most important concepts

Code Examples

- There is a github repository containing code snippets and examples meant to illustrate the various challenges
- This repository also contains answers to the practice cases

Practice

- We will work with several practice cases, the answers of which can be found on the github repository
- Web-scraping is a programming skill and as such it is important that you practice

Contents

What are we going to do today?

Introduction to the internet

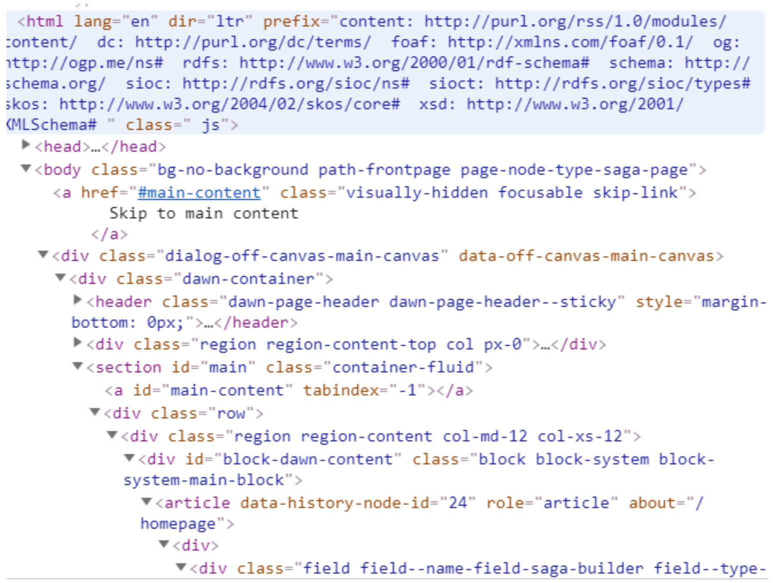
How the internet works

Contents

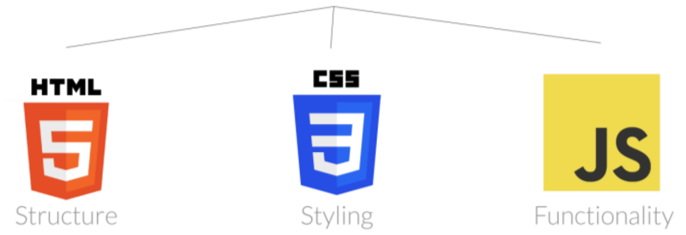
What are we going to do today?

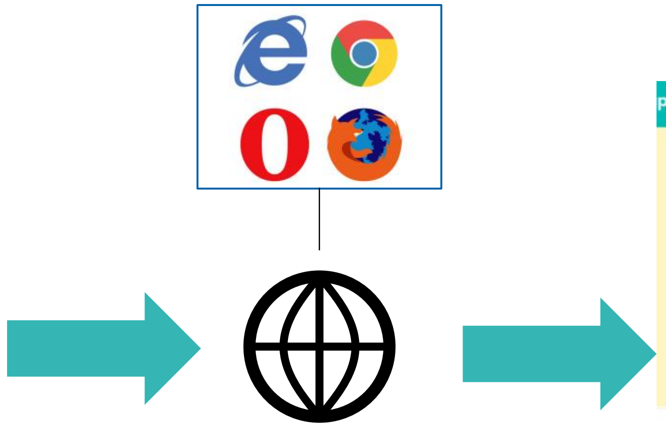
Behind the screens

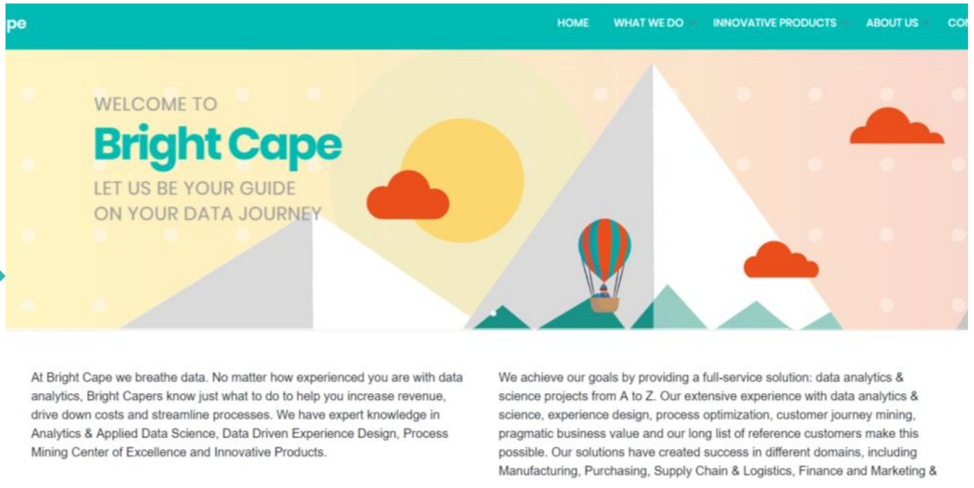
How a webpage is made up



```
<html lang="en" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/
content/  dc: http://purl.org/dc/terms/  foaf: http://xmlns.com/foaf/0.1/  og:
http://ogp.me/ns#  rdfs: http://www.w3.org/2000/01/rdf-schema#  schema: http://
schema.org/  sioc: http://rdfs.org/sioc/ns#  sioct: http://rdfs.org/sioc/types#
skos: http://www.w3.org/2004/02/skos/core#  xsd: http://www.w3.org/2001/
XMLSchema# " class=" js">
 ▶<head>…</head>
 ▼<body class="bg-no-background path-frontpage page-node-type-saga-page">
    <a href="#main-content" class="visually-hidden focusable skip-link">
          Skip to main content
        </a>
  ▼<div class="dialog-off-canvas-main-canvas" data-off-canvas-main-canvas>
    ▼<div class="dawn-container">
      ▶<header class="dawn-page-header dawn-page-header--sticky" style="margin-
      bottom: 0px;">…</header>
      ▶<div class="region region-content-top col px-0">…</div>
      ▼<section id="main" class="container-fluid">
          <a id="main-content" tabindex="-1"></a>
        ▼<div class="row">
          ▼<div class="region region-content col-md-12 col-xs-12">
            ▼<div id="block-dawn-content" class="block block-system block-
            system-main-block">
              ▼<article data-history-node-id="24" role="article" about="/
              homepage">
                ▼<div>
                  ▼<div class="field field--name-field-saga-builder field--type-
```





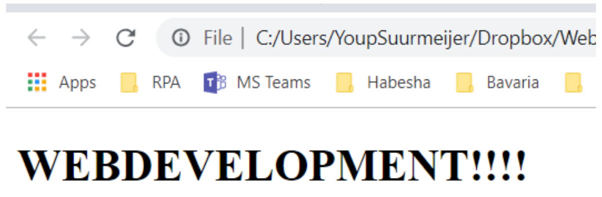


At Bright Cape we breathe data. No matter how experienced you are with data analytics, Bright Capers know just what to do to help you increase revenue, drive down costs and streamline processes. We have expert knowledge in Analytics & Applied Data Science, Data Driven Experience Design, Process Mining Center of Excellence and Innovative Products.

We achieve our goals by providing a full-service solution: data analytics & science projects from A to Z. Our extensive experience with data analytics & science, experience design, process optimization, customer journey mining, pragmatic business value and our long list of reference customers make this possible. Our solutions have created success in different domains, including Manufacturing, Purchasing, Supply Chain & Logistics, Finance and Marketing &

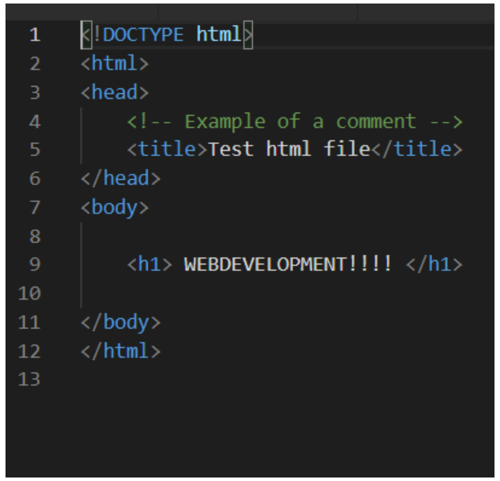
Relationship between HTML and CSS

HTML determines what is placed where on the page, and CSS determines how it is styled

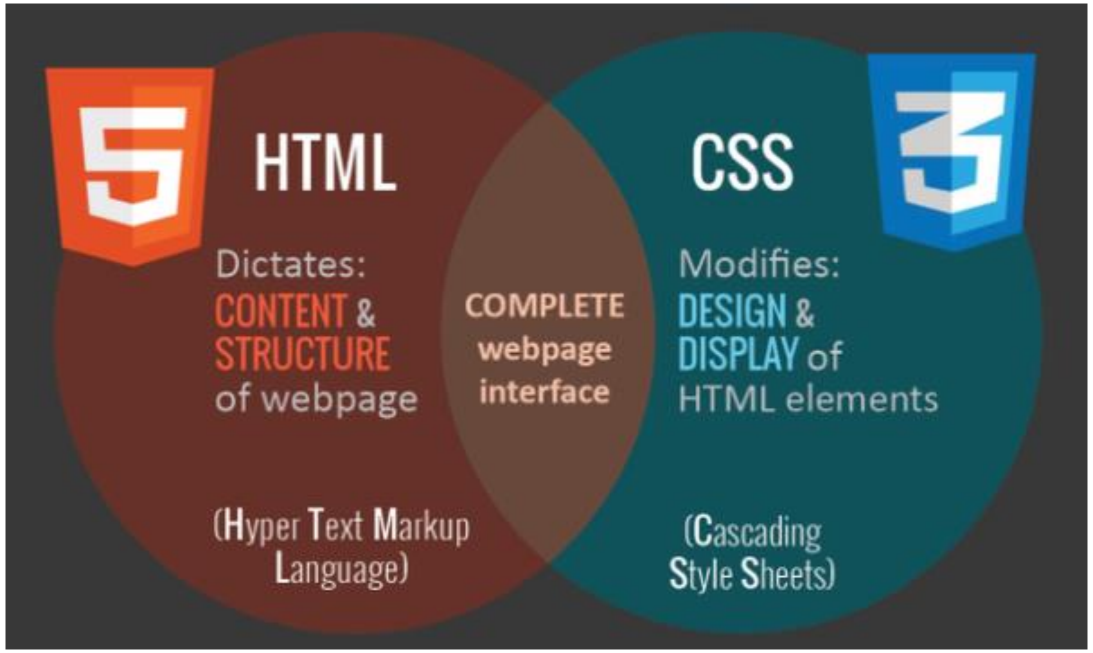

File | C:/Users/YoupSuurmeijer/Dropbox/Web

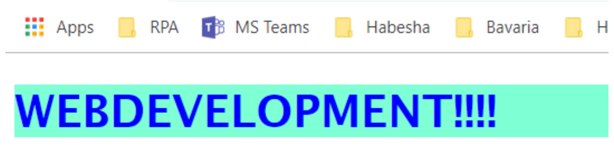
Apps RPA MS Teams Habesha Bavaria

WEBDEVELOPMENT!!!!

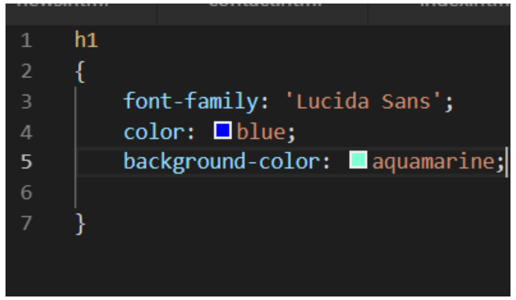


```html
<!DOCTYPE html>
<html>
<head>
    <!-- Example of a comment -->
    <title>Test html file</title>
</head>
<body>

    <h1> WEBDEVELOPMENT!!!! </h1>

</body>
</html>
```




Apps RPA MS Teams Habesha Bavaria H

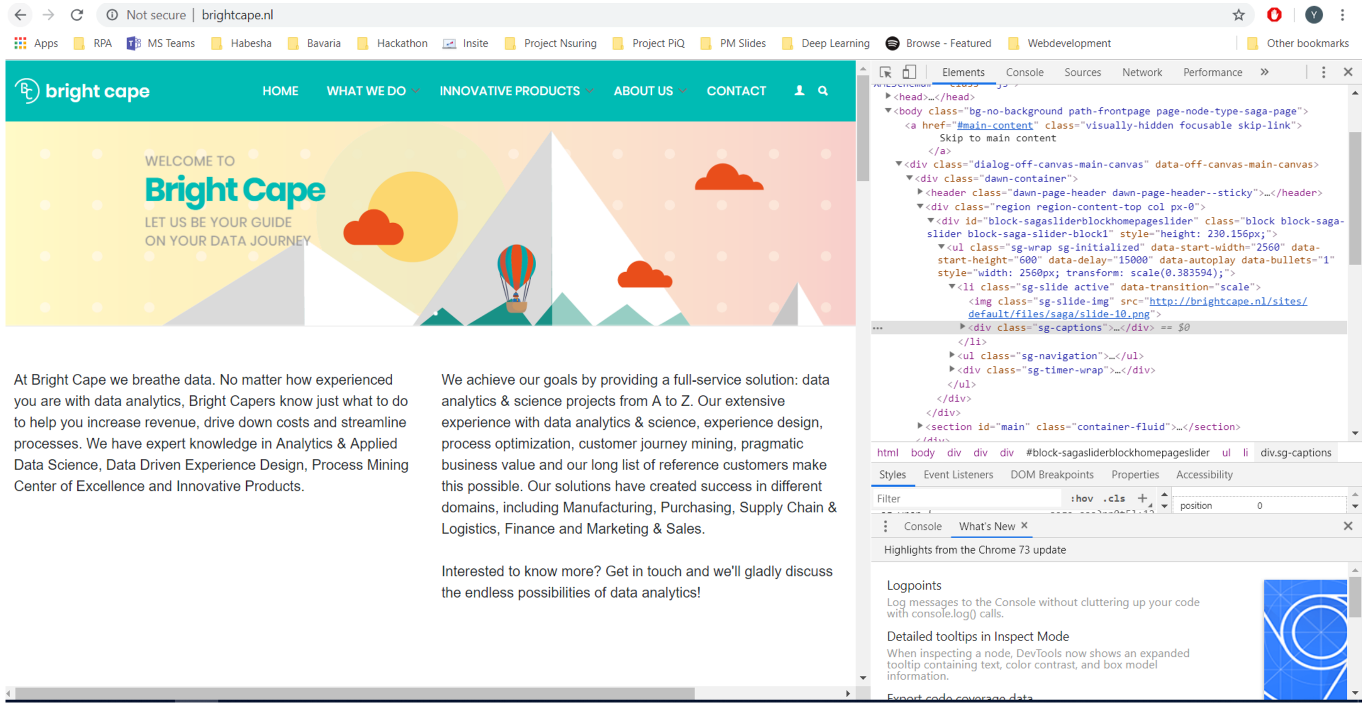
WEBDEVELOPMENT!!!!



```css
h1
{
    font-family: 'Lucida Sans';
    color: blue;
    background-color: aquamarine;
}
```

A look behind the screens of a web browser

Most web browsers have a built-in functionality to view the raw HTML behind a page



At Bright Cape we breathe data. No matter how experienced you are with data analytics, Bright Capers know just what to do to help you increase revenue, drive down costs and streamline processes. We have expert knowledge in Analytics & Applied Data Science, Data Driven Experience Design, Process Mining Center of Excellence and Innovative Products.

We achieve our goals by providing a full-service solution: data analytics & science projects from A to Z. Our extensive experience with data analytics & science, experience design, process optimization, customer journey mining, pragmatic business value and our long list of reference customers make this possible. Our solutions have created success in different domains, including Manufacturing, Purchasing, Supply Chain & Logistics, Finance and Marketing & Sales.

Interested to know more? Get in touch and we'll gladly discuss the endless possibilities of data analytics!

```
▶<head>…</head>
▼<body class="bg-no-background path-frontpage page-node-type-saga-page">
   <a href="#main-content" class="visually-hidden focusable skip-link">
      Skip to main content
   </a>
  ▼<div class="dialog-off-canvas-main-canvas" data-off-canvas-main-canvas>
    ▼<div class="dawn-container">
      ▶<header class="dawn-page-header dawn-page-header--sticky">…</header>
      ▼<div class="region region-content-top col px-0">
        ▼<div id="block-sagasliderblockhomepageslider" class="block block-saga-
        slider block-saga-slider-block1" style="height: 230.156px;">
          ▼<ul class="sg-wrap sg-initialized" data-start-width="2560" data-
          start-height="600" data-delay="15000" data-autoplay data-bullets="1"
          style="width: 2560px; transform: scale(0.383594);">
            ▼<li class="sg-slide active" data-transition="scale">
               <img class="sg-slide-img" src="http://brightcape.nl/sites/
               default/files/saga/slide-10.png">
...           ▶<div class="sg-captions">…</div> == $0
             </li>
            ▶<ul class="sg-navigation">…</ul>
            ▶<div class="sg-timer-wrap">…</div>
           </ul>
         </div>
       </div>
      ▶<section id="main" class="container-fluid">…</section>
```

html body div div div #block-sagasliderblockhomepageslider ul li div.sg-captions

Highlights from the Chrome 73 update

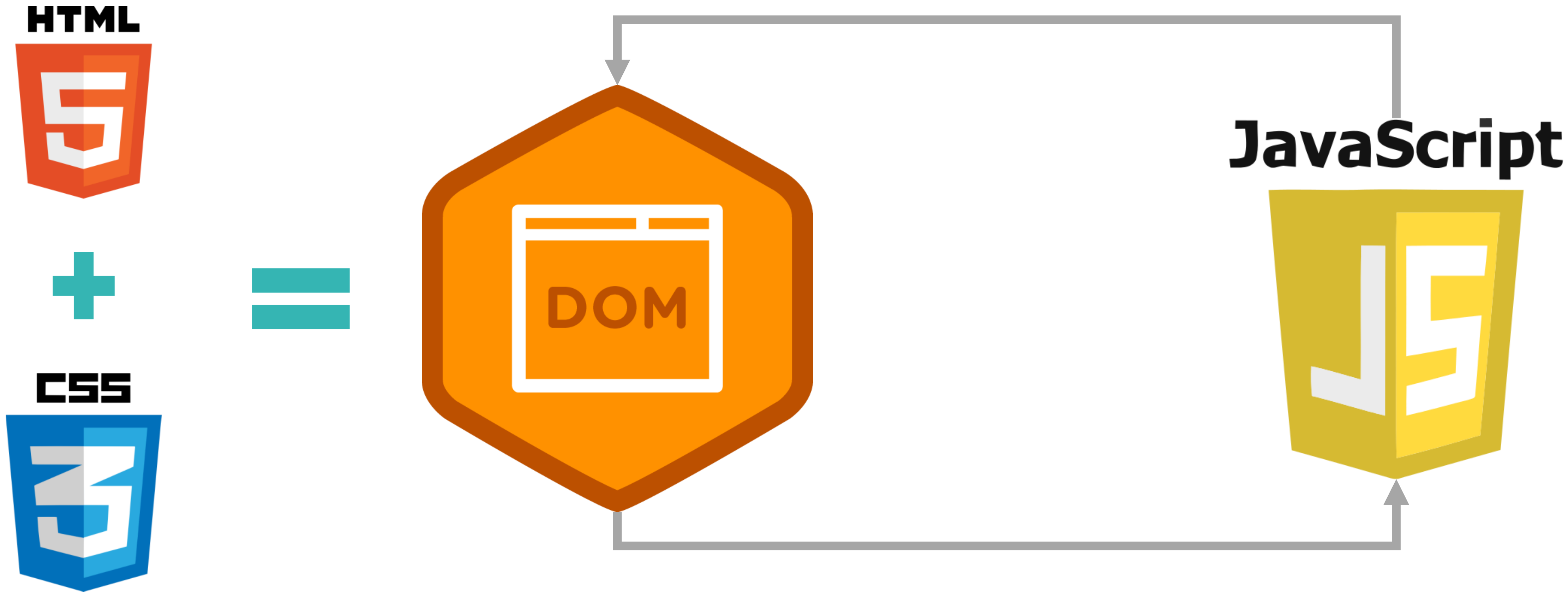
Logpoints
Log messages to the Console without cluttering up your code with console.log() calls.

Detailed tooltips in Inspect Mode
When inspecting a node, DevTools now shows an expanded tooltip containing text, color contrast, and box model information.

1. Open your browser (preferably Chrome)
2. Go to www.brightcape.nl
3. Right click on any element in the screen and select 'Inspect'
4. You can now see the raw HTML that makes up this webpage

Javascript and the DOM

What makes up the DOM and what role does Javascript play in all this?



Example of Javascript and DOM manipulation

The example with Rstudio's shiny tables

https://shiny.rstudio.com/gallery/basic-datatable.html

Contents

What are we going to do today?

The main objective in web-scraping

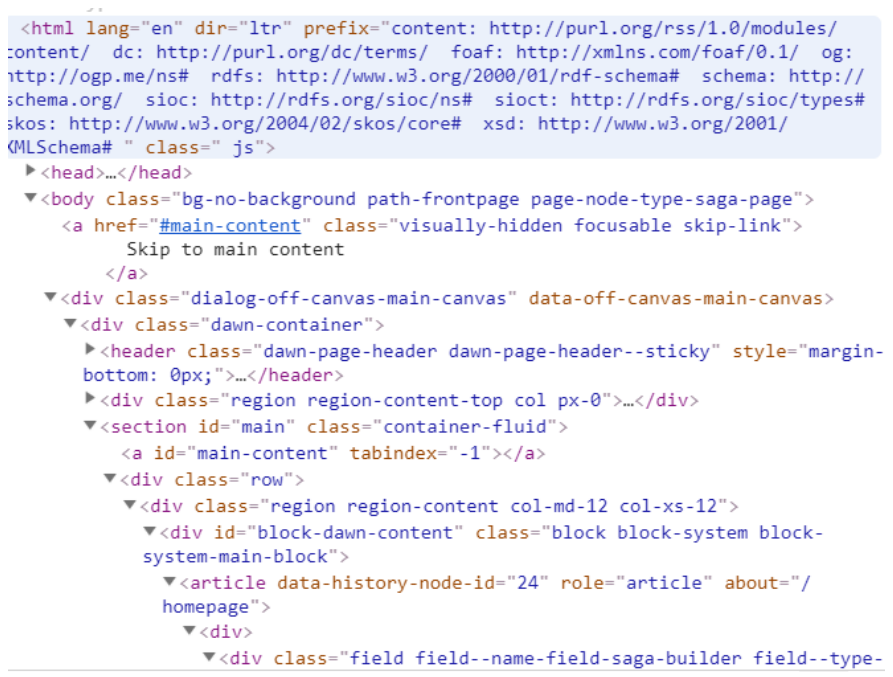
Get the right HTML document, locate the elements and parse them into an understandable dataframe

I. Setup your device

II. Get the right HTML documents

III. Find and retrieve the right HTML elements

IV. Parse everything correctly into a dataframe



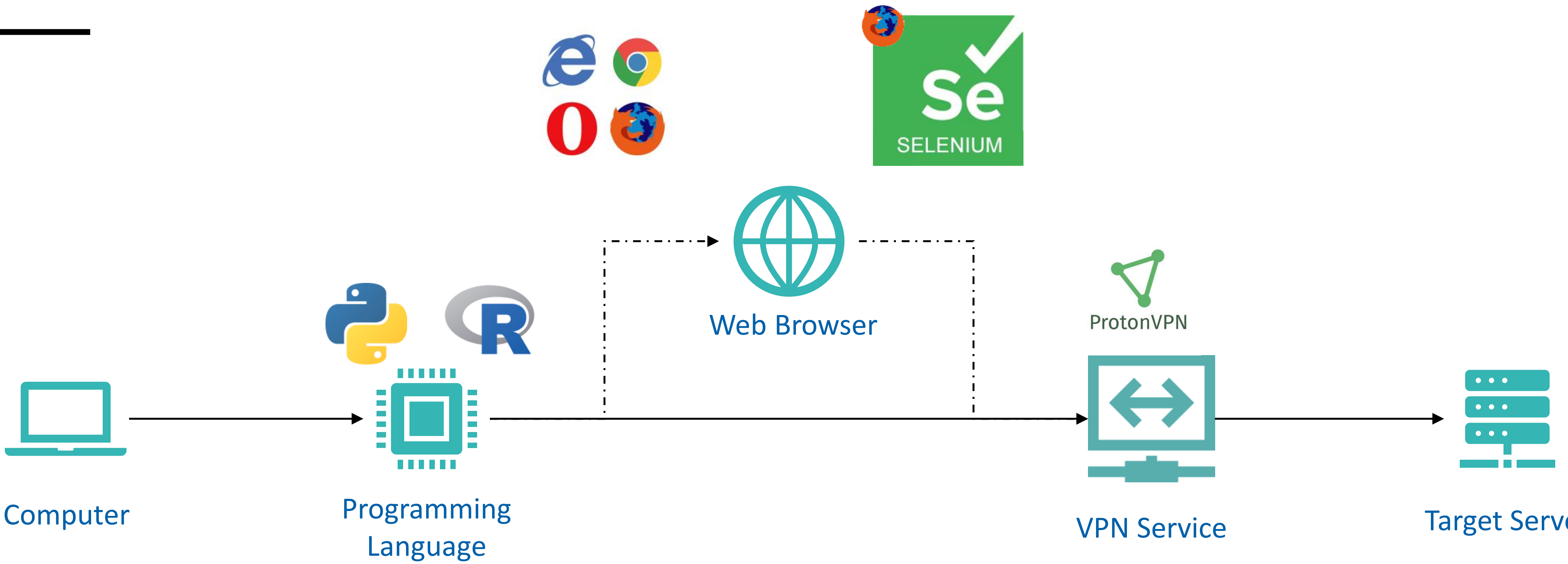
```
 <html lang="en" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/
content/  dc: http://purl.org/dc/terms/  foaf: http://xmlns.com/foaf/0.1/  og:
ttp://ogp.me/ns#  rdfs: http://www.w3.org/2000/01/rdf-schema#  schema: http://
chema.org/  sioc: http://rdfs.org/sioc/ns#  sioct: http://rdfs.org/sioc/types#
kos: http://www.w3.org/2004/02/skos/core#  xsd: http://www.w3.org/2001/
MLSchema# " class=" js">
 ▶<head>…</head>
 ▼<body class="bg-no-background path-frontpage page-node-type-saga-page">
    <a href="#main-content" class="visually-hidden focusable skip-link">
          Skip to main content
        </a>
   ▼<div class="dialog-off-canvas-main-canvas" data-off-canvas-main-canvas>
     ▼<div class="dawn-container">
       ▶<header class="dawn-page-header dawn-page-header--sticky" style="margin-
       bottom: 0px;">…</header>
       ▶<div class="region region-content-top col px-0">…</div>
       ▼<section id="main" class="container-fluid">
          <a id="main-content" tabindex="-1"></a>
         ▼<div class="row">
           ▼<div class="region region-content col-md-12 col-xs-12">
             ▼<div id="block-dawn-content" class="block block-system block-
             system-main-block">
               ▼<article data-history-node-id="24" role="article" about="/
               homepage">
                 ▼<div>
                   ▼<div class="field field--name-field-saga-builder field--type-
```

Contents

What are we going to do today?

Step1: Setting up your device

An overview of the required tools



- A place to run your code and install all required software
- Your favorite programming language to automate the requests and extract and format the target HTML elements
- Some websites check for automated requests and could potentially ban your IP from the website, therefore it's advisable to reroute your traffic just to be sure
- The website/web-adress you whish to scrape

VPN Service

Introduction to VPN services

Proton VPN

Free and secure internet connection

I. Free-tier VPN services available

II. Easy to setup and start using

III. Secure and private services

IV. (same people as behind Proton email services)

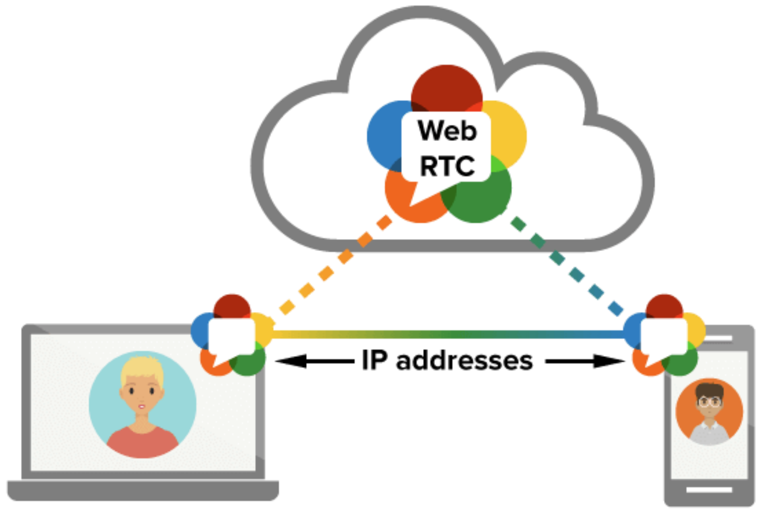
ProtonVPN

IP-leak and WebRTC

An important note on securing your internet connection

Even using VPN services it can potentially be very easy for websites to detect your actual IP by using something called WebRTC.

In short, WebRTC is a connection protocol that allows you to make secure video call to other devices via the internet. Most web browsers have WebRTC enabled by default. When using VPN it is recommendable to disable WebRTC for your browser



More info on disabling WebRTC can be found here

Tip 1: Use Firefox as Selenium browser since disabling WebRTC is easy in Firefox

Tip 2: Always check if your IP address can be detected using the following link

Contents

What are we going to do today?

Step 2: Getting the right HTML/Webpage

The 2 main ways of getting the information you need using your favorite programming language

Direct (Most Common)

- The information you require **is directly available via a URL get request** (it's immediately displayed when you open webpage)
- No need to manipulate/traverse different pages

requests/httplib/urllib

httR/curl

Indirect

- The information you **require is not directly available via a URL get request**
- You do need to manipulate inputs/filters or traverse different pages to unknown URLs
- Page uses Javascript DOM manipulation
- Need to login via form or other difficult page manipulations

Also available as packages for R/Python

What is Selenium

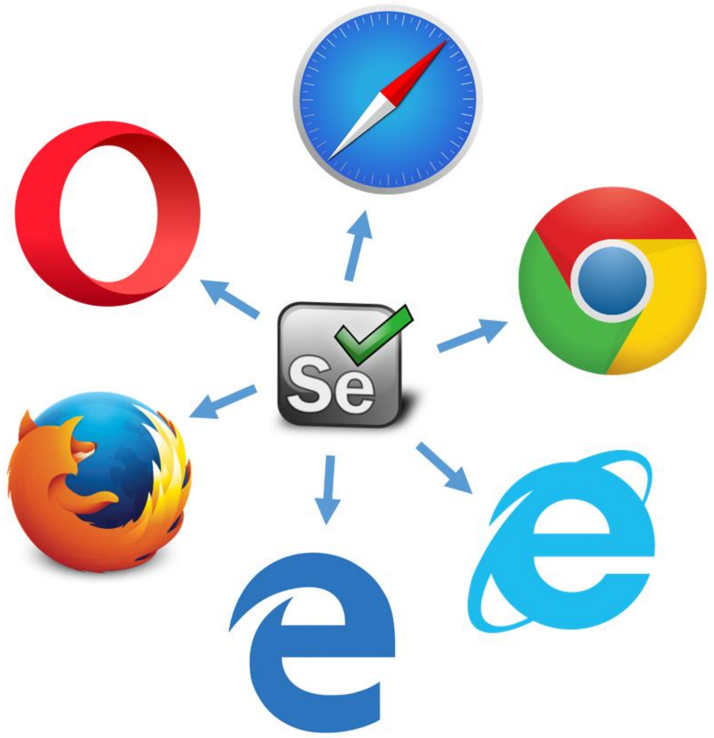
Automated webbrowser, enabling you to automatically get pages, click, type and do all other sorts of things

Selenium is a suite of tools to automate web browsers across many platforms.
Selenium...

- runs in many browsers and operating systems
- can be controlled by many programming languages and testing frameworks.

https://www.seleniumhq.org/

Selenium can be used with:



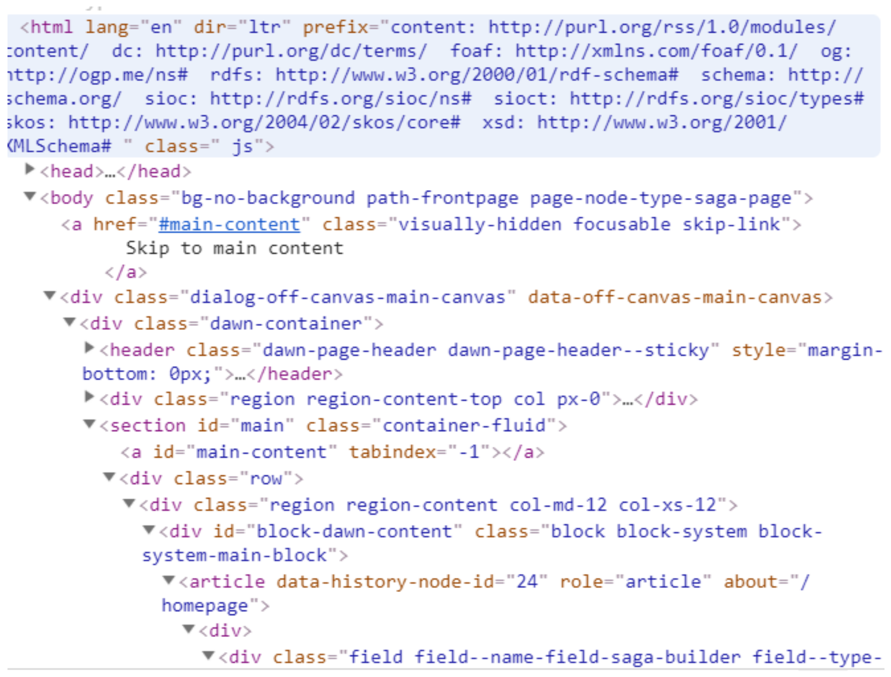
Contents

What are we going to do today?

Step 3: Find and retrieve the right HTML elements

Filtering out only the information you need from the webpage

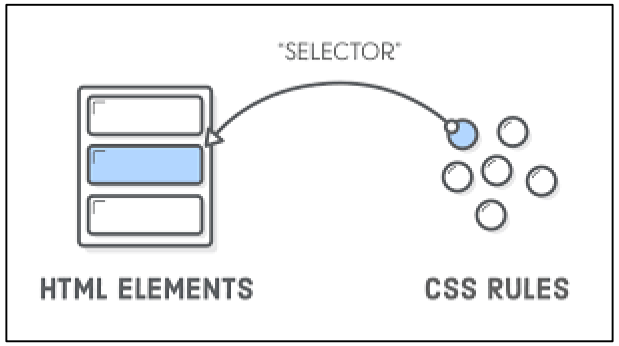
If you think about it, an HTML document is nothing more than a very long text file/string with a lot of redundant information



```
 <html lang="en" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/
content/  dc: http://purl.org/dc/terms/  foaf: http://xmlns.com/foaf/0.1/  og:
http://ogp.me/ns#  rdfs: http://www.w3.org/2000/01/rdf-schema#  schema: http://
schema.org/  sioc: http://rdfs.org/sioc/ns#  sioct: http://rdfs.org/sioc/types#
skos: http://www.w3.org/2004/02/skos/core#  xsd: http://www.w3.org/2001/
XMLSchema# " class=" js">
 ▶<head>…</head>
 ▼<body class="bg-no-background path-frontpage page-node-type-saga-page">
    <a href="#main-content" class="visually-hidden focusable skip-link">
          Skip to main content
       </a>
   ▼<div class="dialog-off-canvas-main-canvas" data-off-canvas-main-canvas>
     ▼<div class="dawn-container">
       ▶<header class="dawn-page-header dawn-page-header--sticky" style="margin-
       bottom: 0px;">…</header>
       ▶<div class="region region-content-top col px-0">…</div>
       ▼<section id="main" class="container-fluid">
          <a id="main-content" tabindex="-1"></a>
         ▼<div class="row">
           ▼<div class="region region-content col-md-12 col-xs-12">
             ▼<div id="block-dawn-content" class="block block-system block-
             system-main-block">
               ▼<article data-history-node-id="24" role="article" about="/
               homepage">
                 ▼<div>
                   ▼<div class="field field--name-field-saga-builder field--type-
```

1.

2.

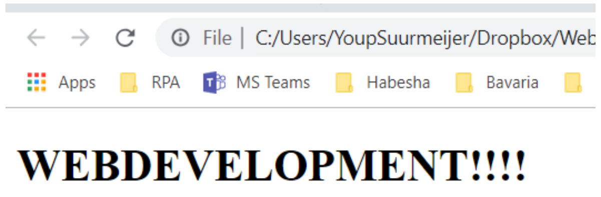


We will treat 2 ways of getting the information we want:

1. Regular Expressions (REGEX)
2. CSS selectors

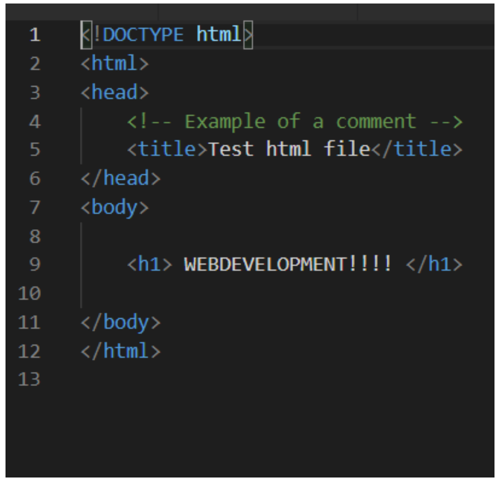
Remember: Relationship between HTML and CSS

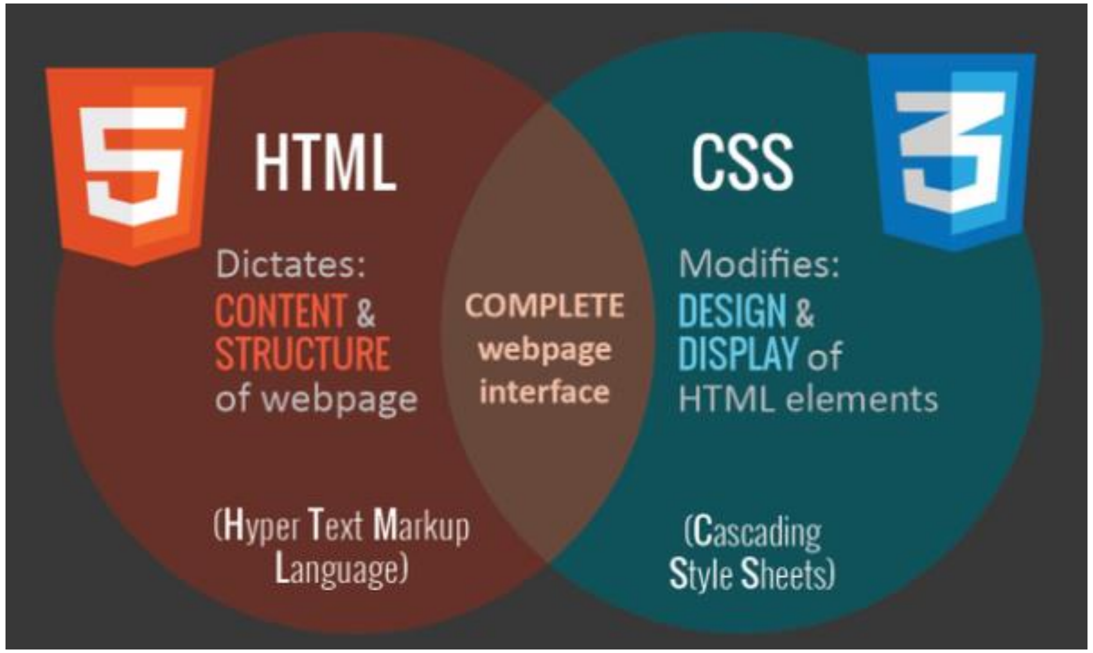
HTML determines what is placed where on the page, and CSS determines how it is styled

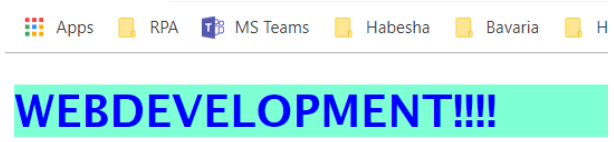

File | C:/Users/YoupSuurmeijer/Dropbox/Web

Apps RPA MS Teams Habesha Bavaria

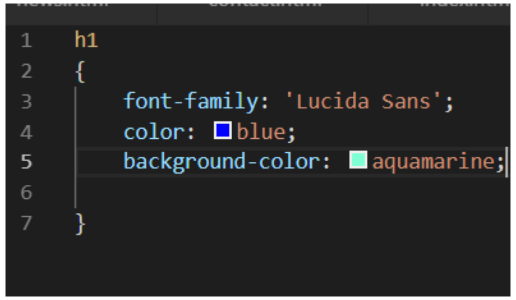
WEBDEVELOPMENT!!!!


```
<!DOCTYPE html>
<html>
<head>
    <!-- Example of a comment -->
    <title>Test html file</title>
</head>
<body>

    <h1> WEBDEVELOPMENT!!!! </h1>

</body>
</html>
```


HTML
Dictates: CONTENT & STRUCTURE of webpage
(Hyper Text Markup Language)
COMPLETE webpage interface
CSS
Modifies: DESIGN & DISPLAY of HTML elements
(Cascading Style Sheets)

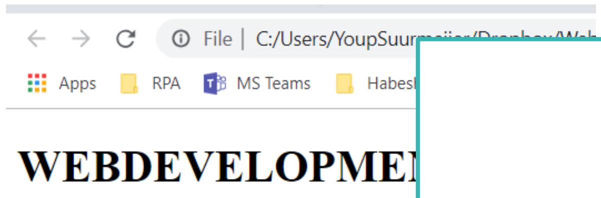

Apps RPA MS Teams Habesha Bavaria H

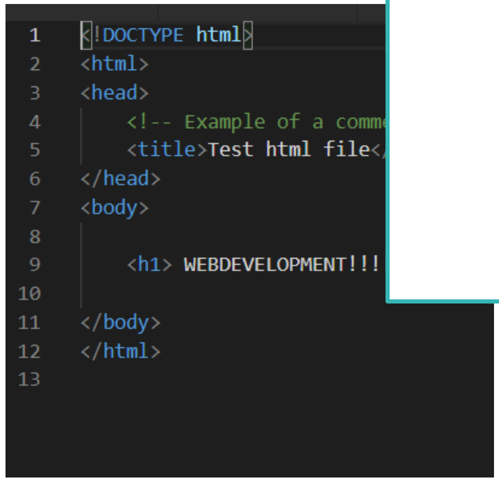
WEBDEVELOPMENT!!!!


```
h1
{
    font-family: 'Lucida Sans';
    color: blue;
    background-color: aquamarine;

}
```

Remember: Relationship between HTML and CSS

HTML determines what is placed where on the page, and CSS determines how it is styled

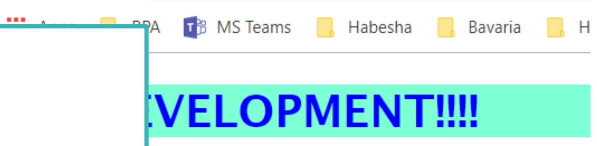


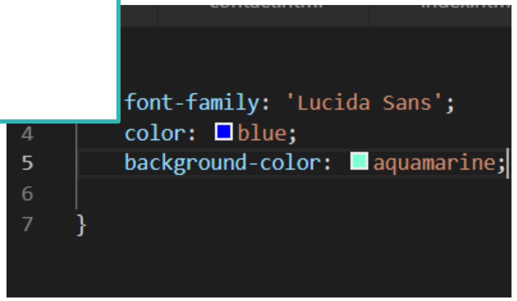


```
<!DOCTYPE html>
<html>
<head>
    <!-- Example of a comm
    <title>Test html file</
</head>
<body>

    <h1> WEBDEVELOPMENT!!!

</body>
</html>
```

How does your webbrowser know which HTML elements should get what CSS styling?

The simple answer….. CSS selectors!

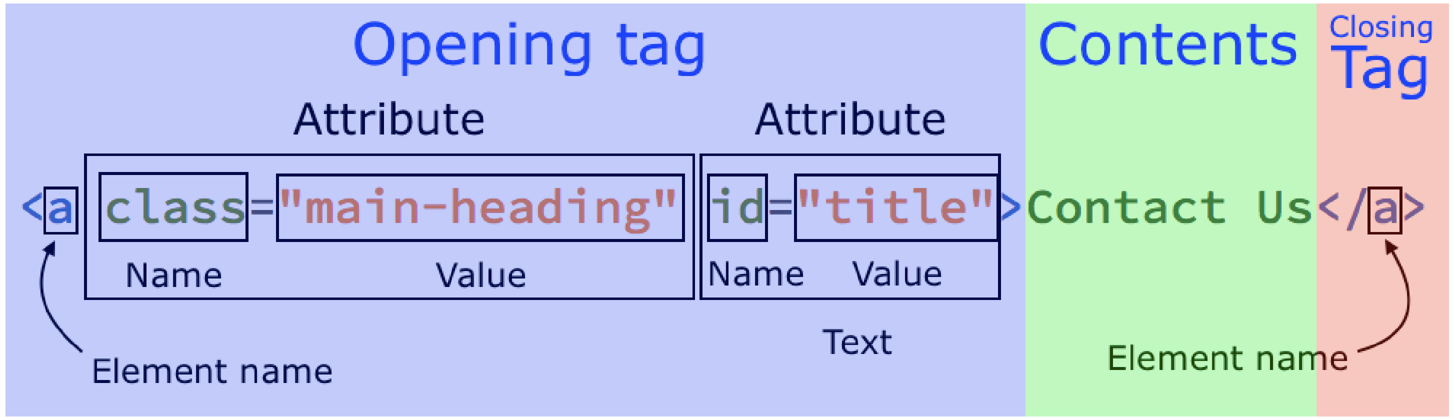


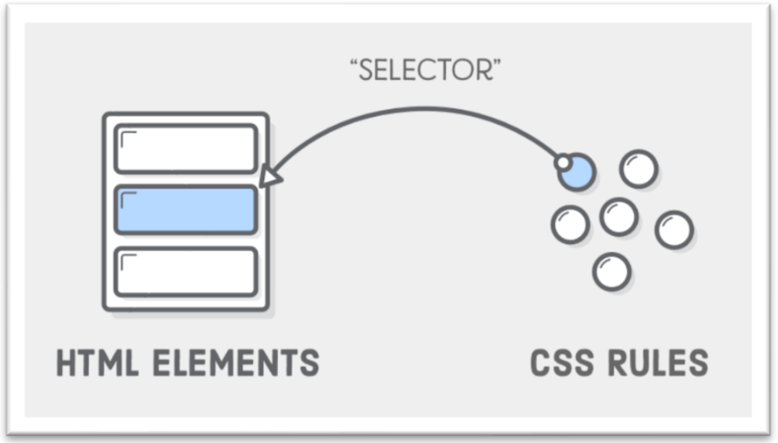


```
    font-family: 'Lucida Sans';
    color: blue;
    background-color: aquamarine;

}
```

CSS selectors

The link between HTML elements and CSS rules





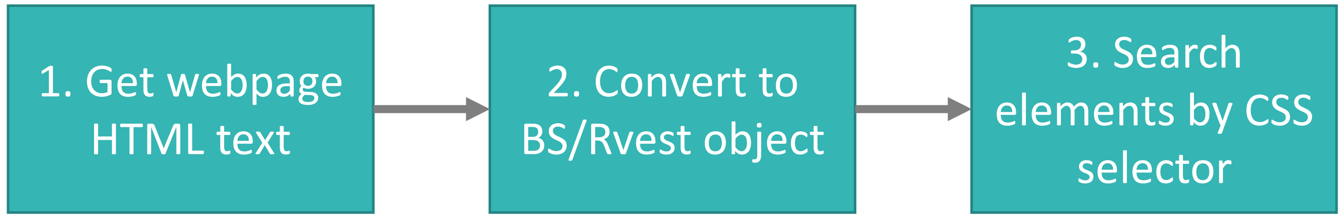
Some CSS selectors you will often use:

- Tag property
- Class
- Href
- ID

BeautifulSoup/Rvest

Using CSS selectors in your favorite programming language

Wouldn’t it be easy if your programming language knew how to select certain elements from the DOM, just like your browser?
Luckily there are packages available that spefically deal with this issue


1. Get webpage HTML text
2. Convert to BS/Rvest object
3. Search elements by CSS selector

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

https://blog.rstudio.com/2014/11/24/rvest-easy-web-scraping-with-r/

Regular expressions (REGEX)

What are regular expressions and what do we use them for?

What is a Regular Expression?

Basically, a regular expression is a pattern describing a certain amount of text. Their name comes from the mathematical theory on which they are based. But we will not dig into that. You will usually find the name abbreviated to "regex" or "regexp".

What do we use Regular Expressions for?

Regex is used to extract substrings with a certain pattern from a larger string object. More plainly put, with regex you can extract certain parts of a text based on some form of logic. Regex can come in very handy in webscraping it's not always possible to extract the required data from the DOM using CSS selectors

Example:

Say I want to extract the number of sheep in from the following text:

"The farmer has over 20 sheep on his farm, and nearly 100 cows"

We could try to just remove all the non-integer elements from the string, but then we'd end up with 2 numbers. A much easier way would be to use regex.

https://regexr.com/

Regex example continued

It’s not always possible to use CSS selectors to get the right elements, in that case regex comes in handy!

Example 1: Extract Look, Feel, Smell and Taste separately

https://www.beeradvocate.com/beer/profile/184/1402/ (click inspect element)

Example 2: Finding out which beers are on sale

https://www.supermarktaanbiedingen.com/aanbiedingen/jan_linders/2019/14

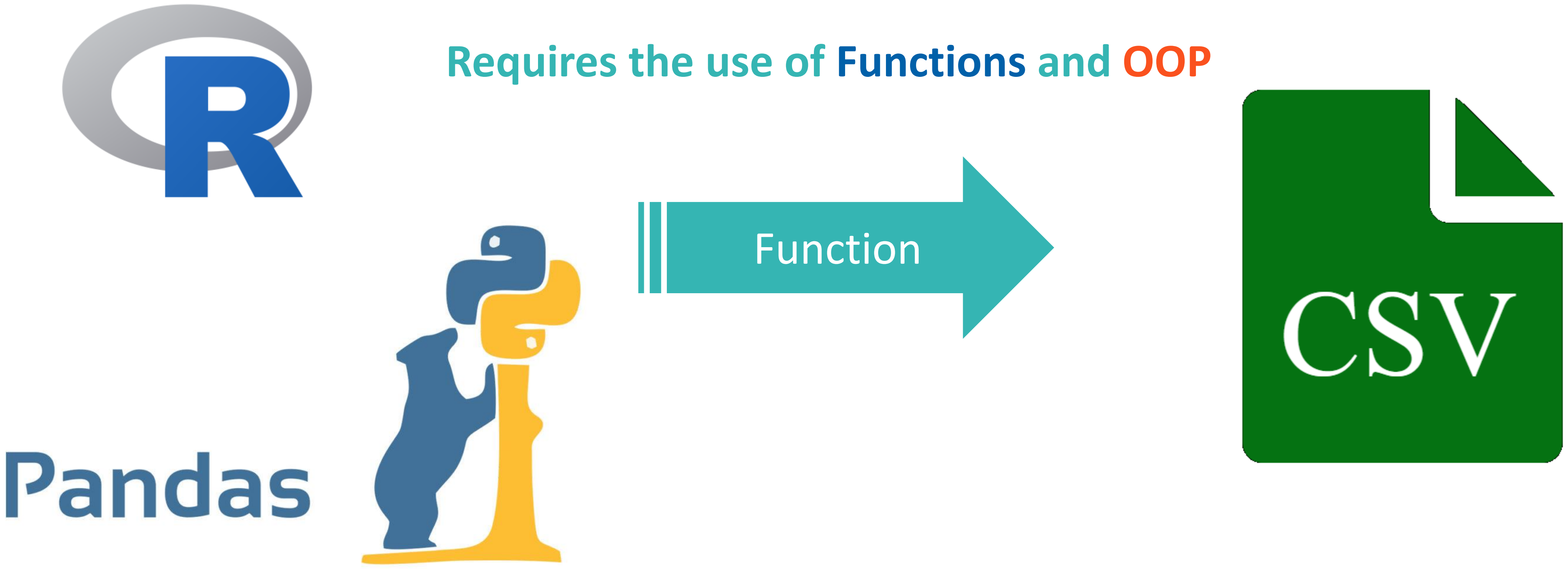
Contents

What are we going to do today?

Step 4: Storing the information in a table

Use functions to convert the data to the required format for analysis (usually CSV)

Requires the use of Functions and OOP



Contents

What are we going to do today?

Code Examples

There is a github repository available containing notebooks and code snippets

I have made a private Github repository containing several examples, jupyter notebooks and code snippets to help you get started with webscraping.

- Simple example for a get request and parsing data using Python and BeautifulSoup4
- Example code for the Mathematicians problem
- Code containing the webscraping bot for Beeradvocate.com

https://github.com/ysuurmei/WS_Training

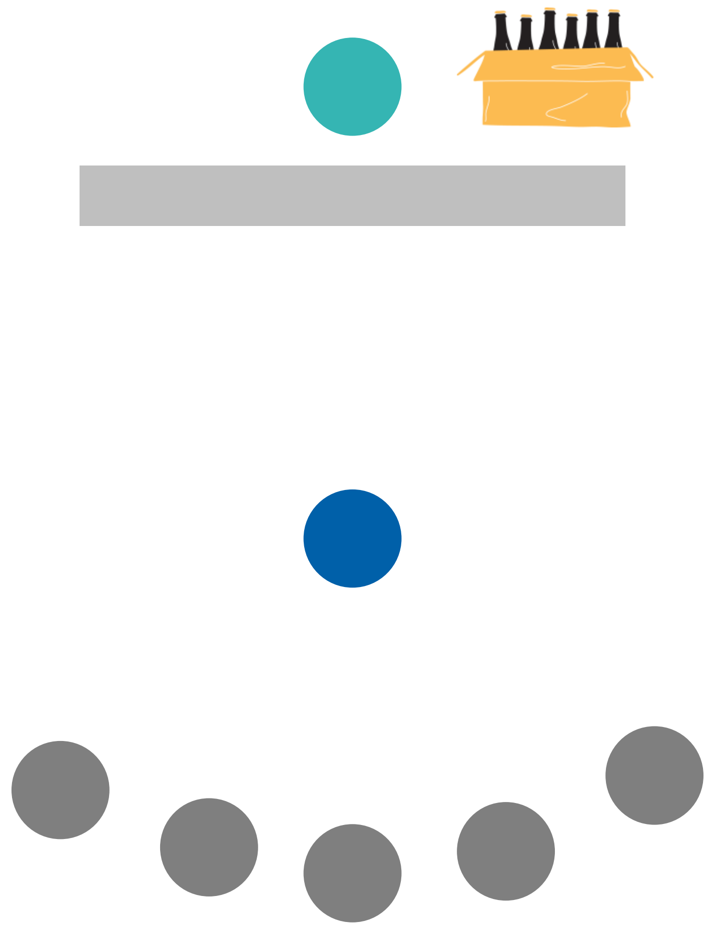
Contents

What are we going to do today?

Advanced Topics: Scraping strategies

Getting simply one page is easy, but what is you need to get 10,000+ pages?

Say it’s your birthday and you’re out in a bar celebrating with your friends. Because it’s your birthday you want to buy all your friends a round of beer, how would you go about this process?

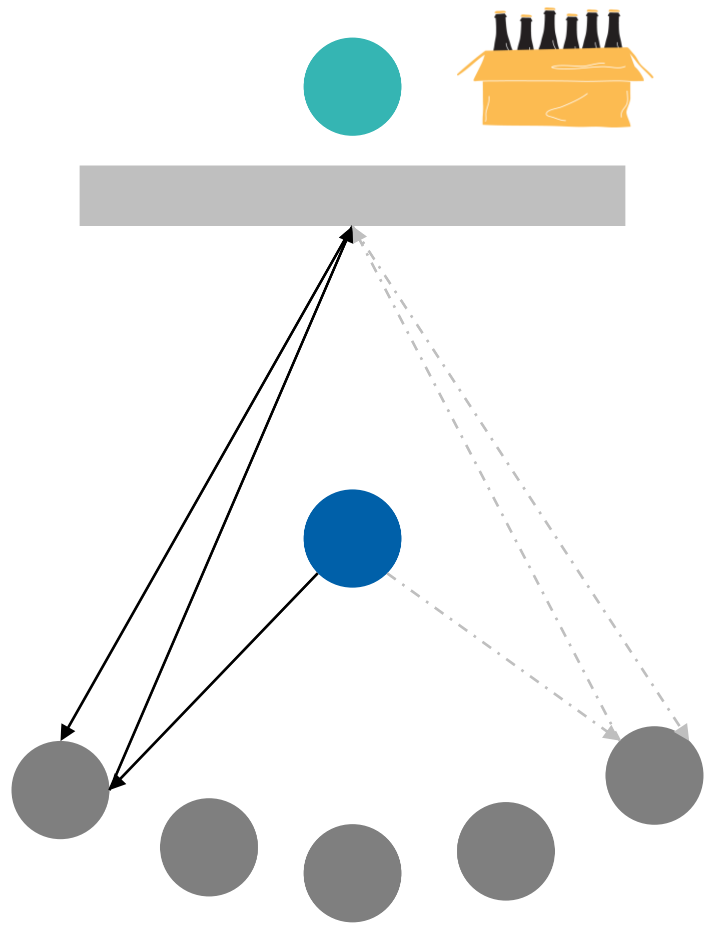


Advanced Topics: Scraping strategies

Choosing the right strategy for your scraping application

One way to go about it is to ask each friend individually if he wants a beer and subsequently go to the bar and get it for him

However this will mean a lot of extra walking….

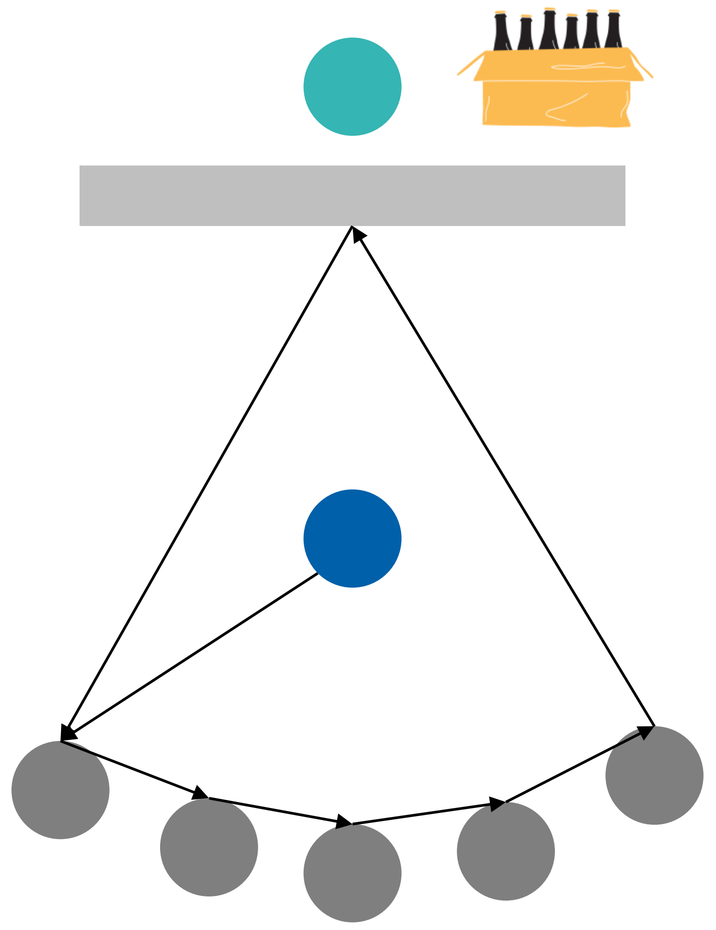


Advanced Topics: Scraping strategies

Choosing the right strategy for your scraping application

Another way to go about this problem is to first ask all your friends if they want a beer and subsequently get all the beers at the bar for everyone at once

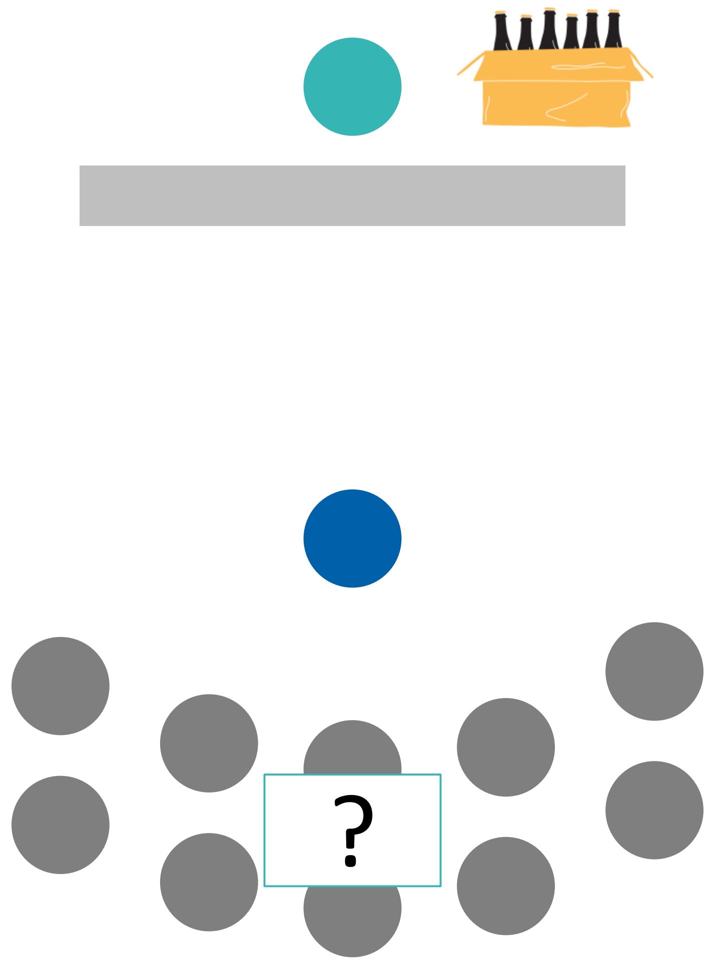
Sounds like a better solution right?



Advanced Topics: Scraping strategies

Choosing the right strategy for your scraping application

Now what do you do when you don't know how many of your friends are in the bar? Or if you don't exactly know where to find all your friends in the bar?



Advanced Topics: Scraping strategies

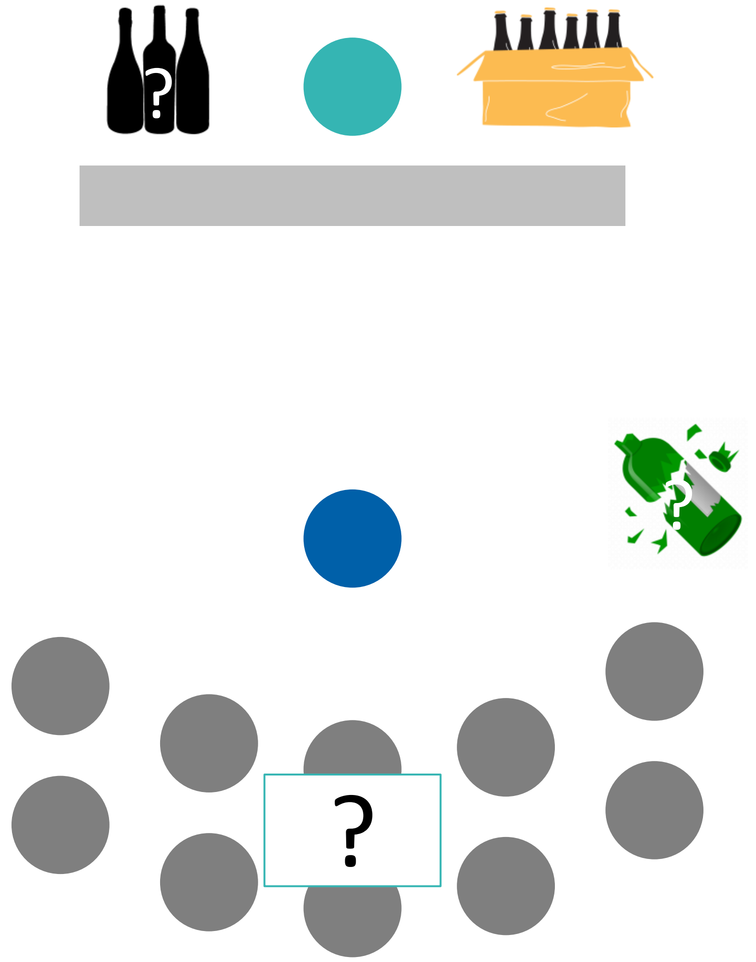
Choosing the right strategy for your scraping application

What if beer isn't the only thing you can get at the bar and you want to get your friends different things?

How would you tackle the issue if you have a small number of friends vs if you have a large amount of friends?

What if there is a chance you drop a load of drinks before they reach your friends?

There is no one given answer to all problem sets, the important thing is that you think beforehand on what your scraping problem looks like and how you could go about the problem most efficiently

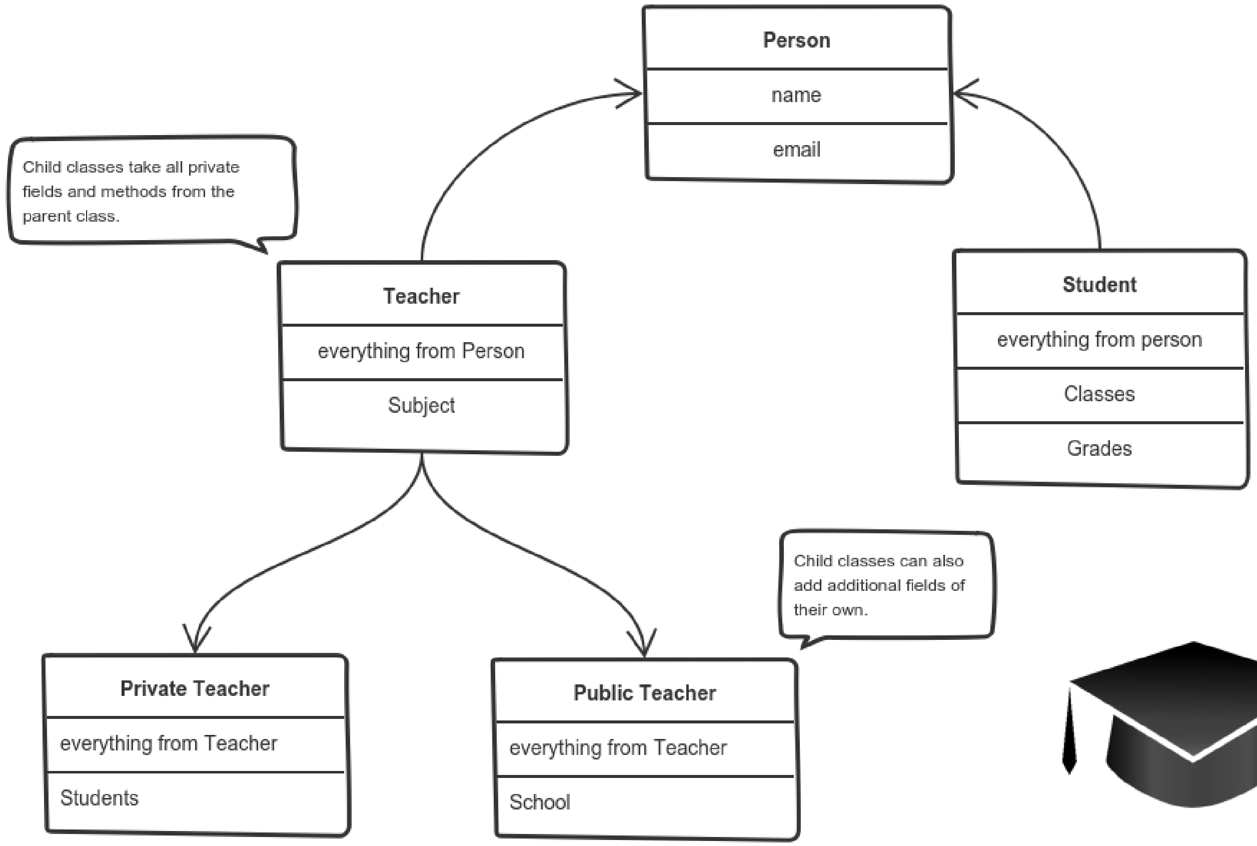


Contents

What are we going to do today?

Advanced Topics: OOP

Taking your programming skills to the next level!



Tutorials & resources:

https://www.youtube.com/watch?v=pTB0EiLXUC8

(Python/R specifics)

https://realpython.com/python3-object-oriented-programming/

https://adv-r.hadley.nz/oo.html

Contents

What are we going to do today?

Advanced Topics: Multi-Threading

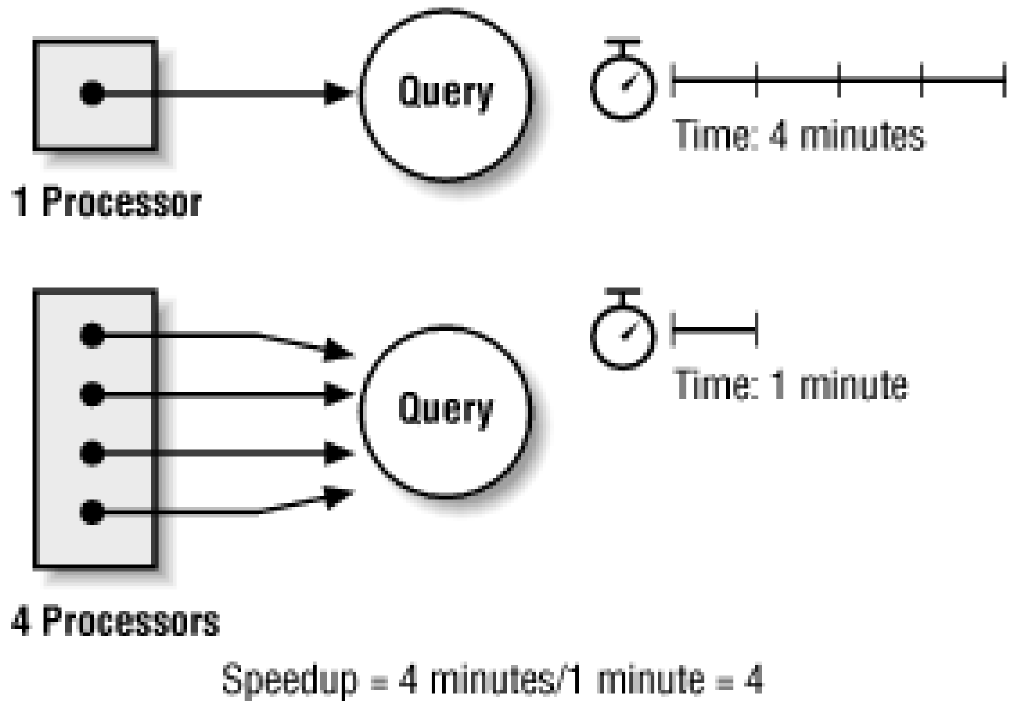
Speeding up your application even more!

Speeding up your scripts:

Using multi-threading you can execute multiple Python scripts at the same time. You can use this method to significantly speed up your code by running multiple instances at the same time.

Important!!!

You can use multi-threading to run multiple instances of your code, but since you are executing HTML requests the speed of your internet connection is always going to limit how fast you can execute you webscraping script!!!



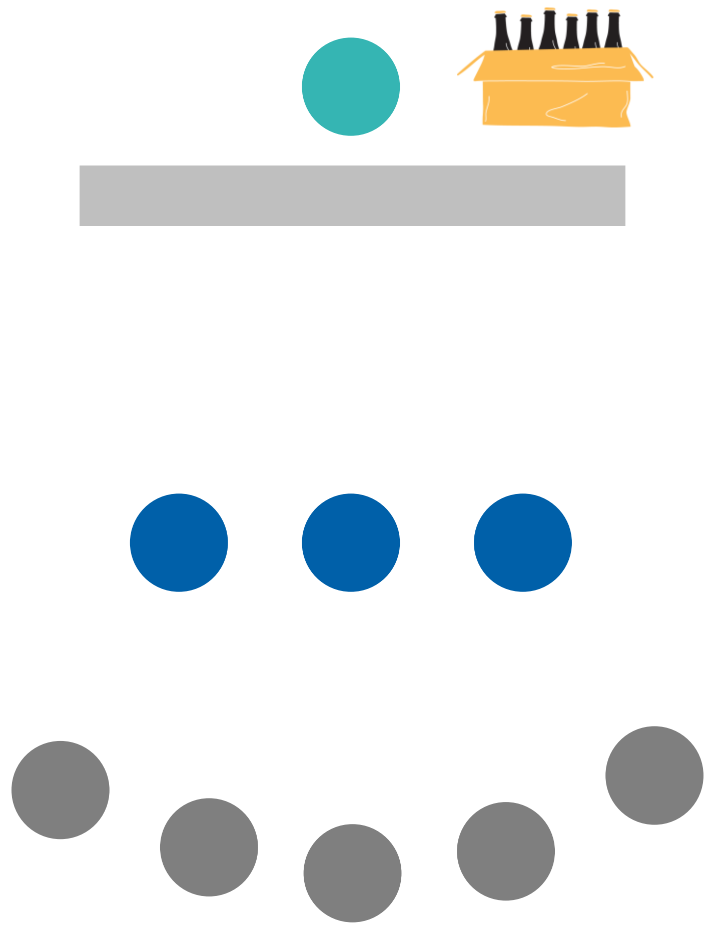
Advanced Topics: Multi-Threading

Back to the bar example…

Going back to the bar analogy, using multi-threading basically boils down to employing multiple people at the same time to get everyone drinks, this will mean you'll get your orders in quicker

However, the barman can only pour you drinks as quickly as the tap runs, in practice this means your internet connection speed determines how fast you can get the beer for all your friends

There are ways to employ more barkeepers, or to make the tap run faster, but that's a topic for another time….



Contents

What are we going to do today?

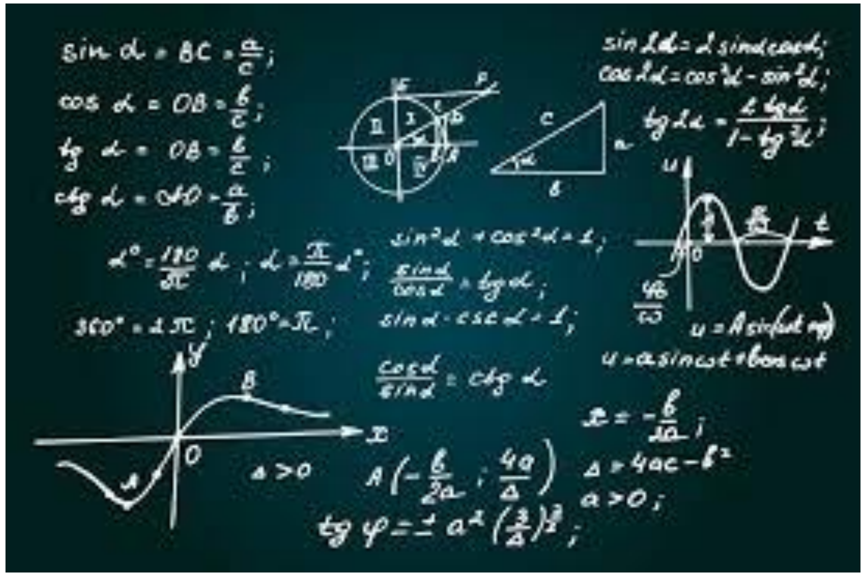
Example code: Popular Mathematicians

Get the 100 most well known mathematicians from a website

The website http://www.fabpedigree.com/james/mathmen.htm contains a list of the 100 greatest mathematicians of all time. The site is an example of a very simple HTML static webpage and therefore ideal as a starting point for the training.

Now say we would be interested finding out which of these mathematicians published the most books. We can’t find that information on this page, but perhaps there is another data source that does contain such info, i.e. the website of the Toronto Public Library https://www.torontopubliclibrary.ca

What if we could combine data from these 2 sources to determine the most literate mathematicians in history?

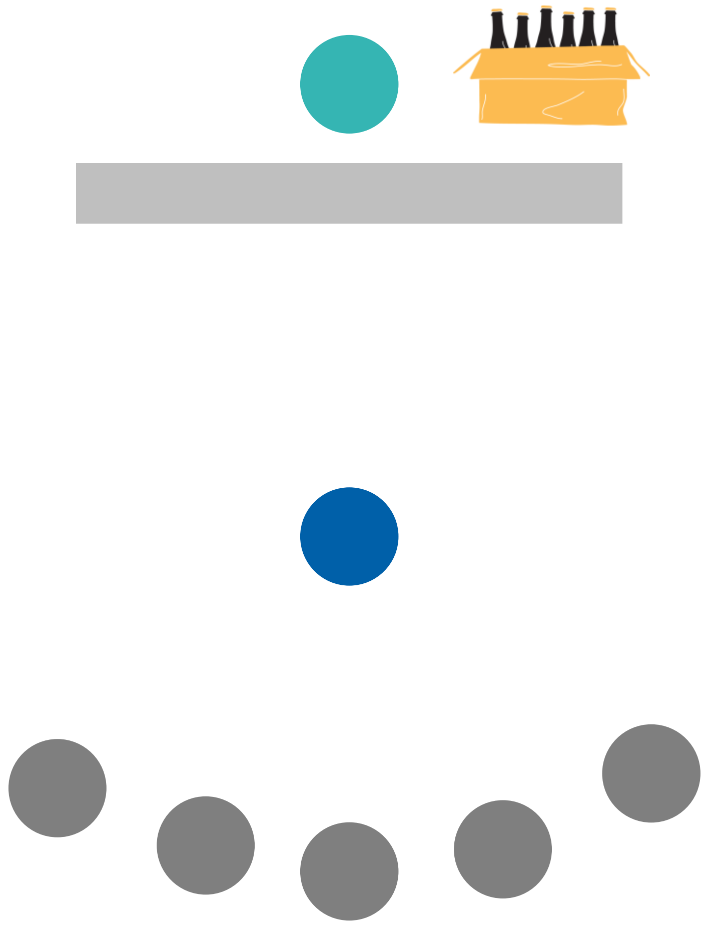


Example code: Popular Mathematicians

The webscraping strategy

?

What strategy would you use?



Example code: Popular Mathematicians

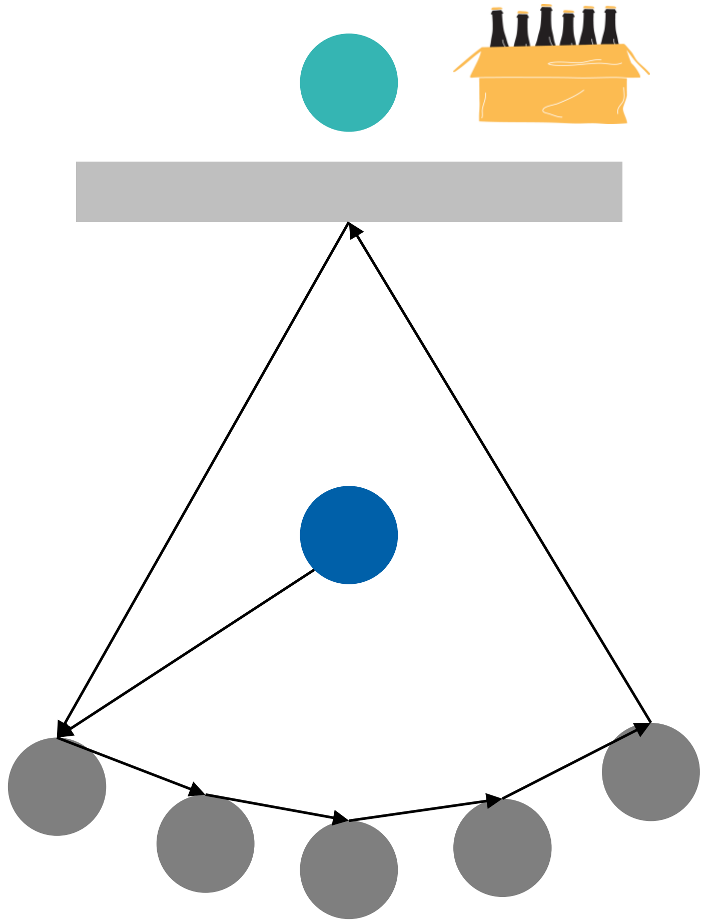
The webscraping strategy

We Know:

✓How many mathematicians

✓The location of each mathematician

✓Where to get the search results for each mathematician



Example code: Beer Reviews

Example of the 'real-deal' using webscraping to access beer-reviews

In this next example we'll dive into the code for the Beer-Advocate webscraping tool. The scraping problem faced at BA is far more complex than the mathematicians example.

Focus of this example will be on the use of advanced topics for webscraping:

- Selenium
- Object Oriented Programming
- Try – Except error handling
- Scraping strategies

The goal is for you to understand what is happening in the code, and to give you examples that will help you build your own web-scraping algorithm.

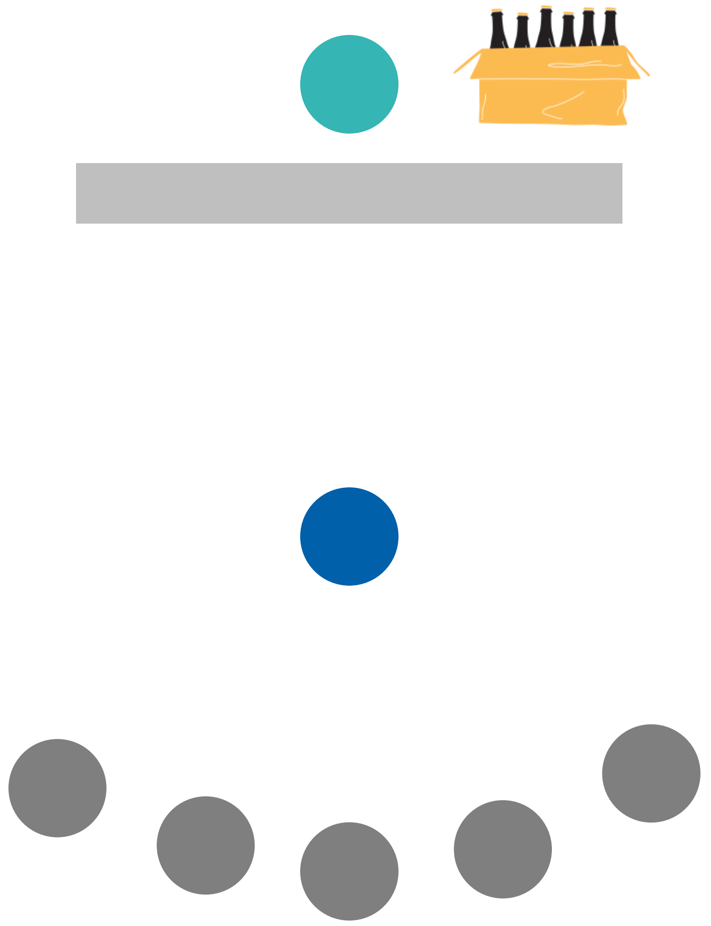
https://www.beeradvocate.com/

Example code: Beer Advocate

The webscraping strategy

?

What strategy would you use?



Example code: Beer Advocate

The webscraping strategy

We Know:

✓ How many beers we want to scrape

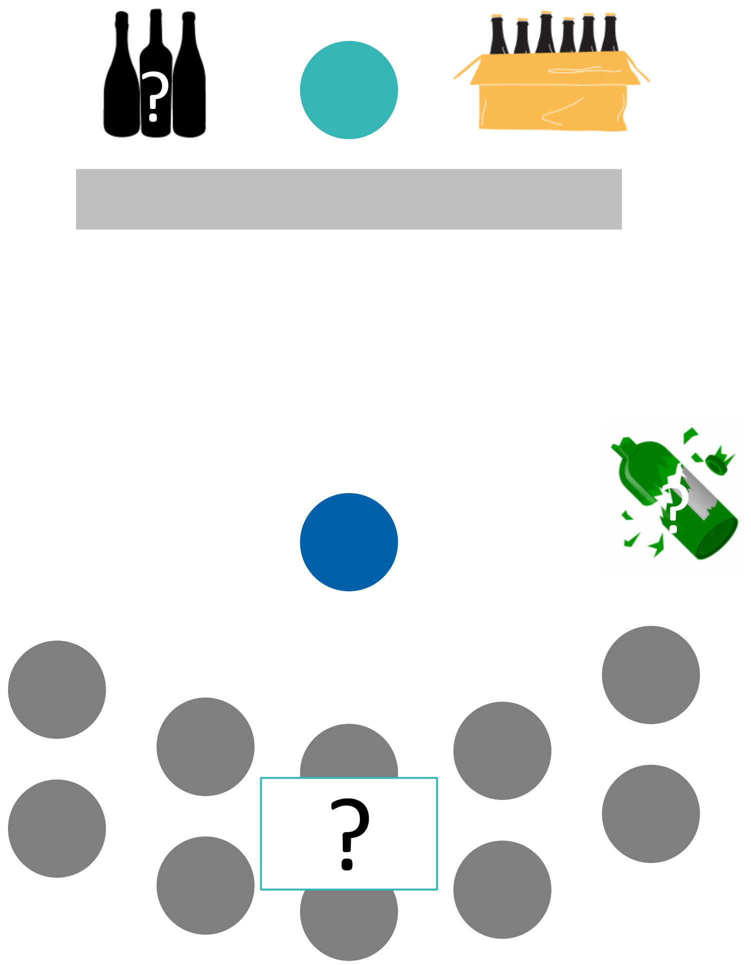
✓ How to search for the landing page of a given beer

We Don't Know:

✓ Where we end up after querying a beer

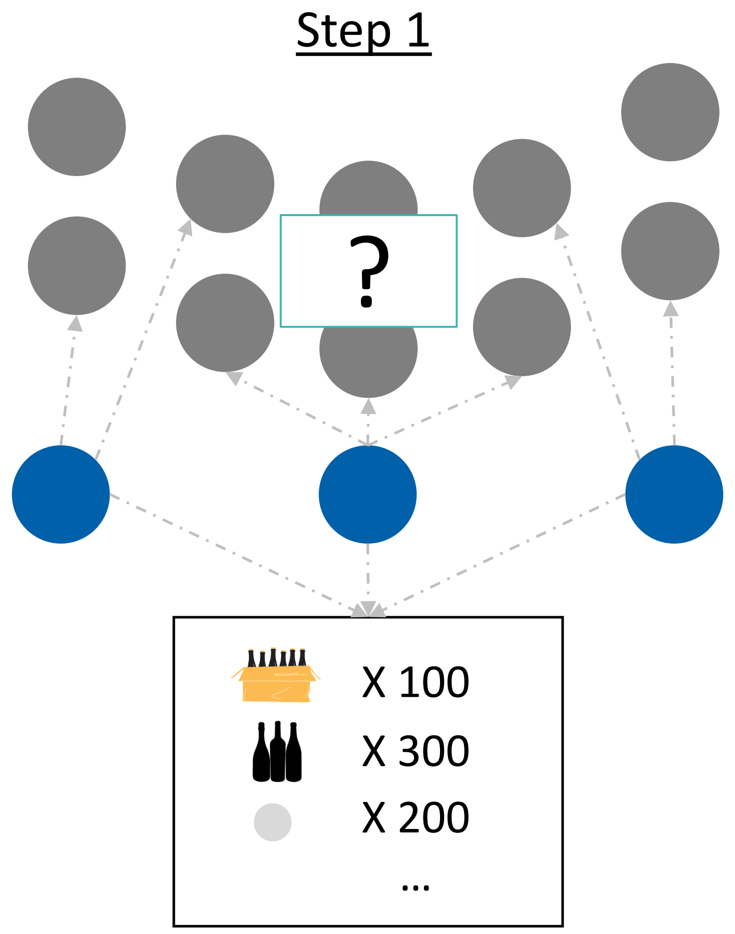
✓ How many reviews there are for each beer

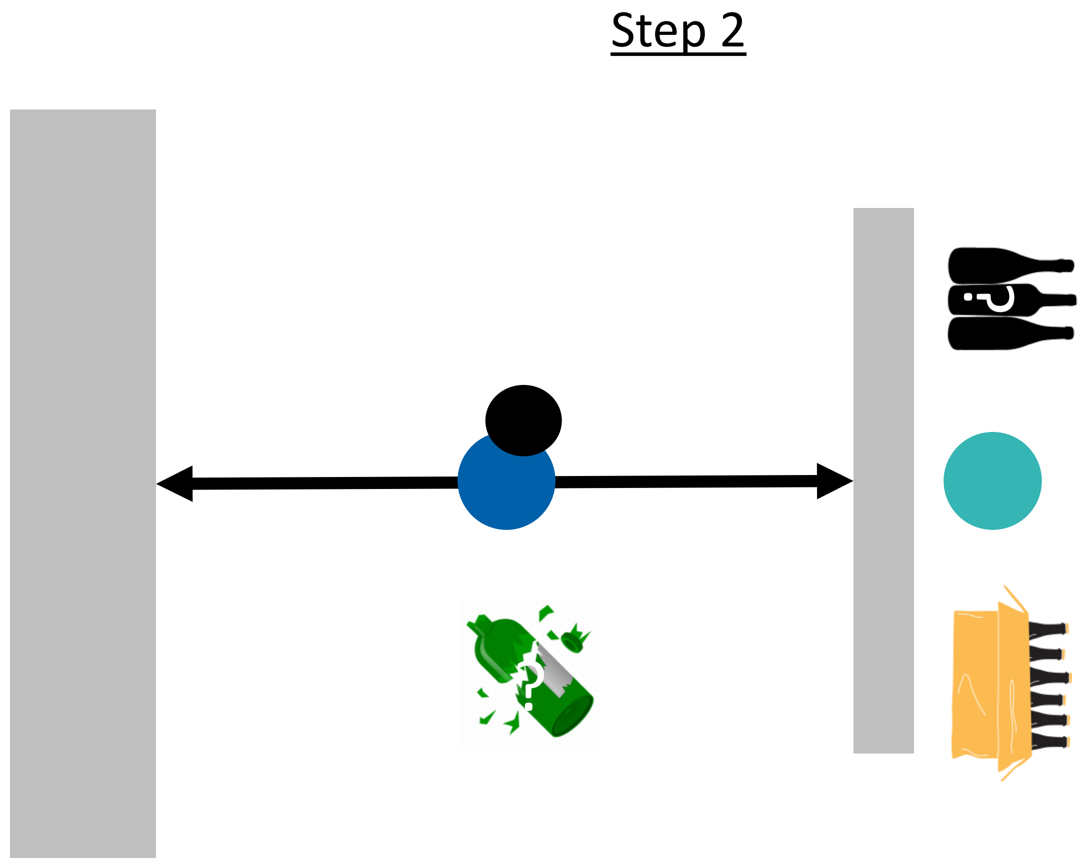
✓ Where these reviews are located



Example code: Beer Advocate

The final solution: a 2 step strategy using multiple agents





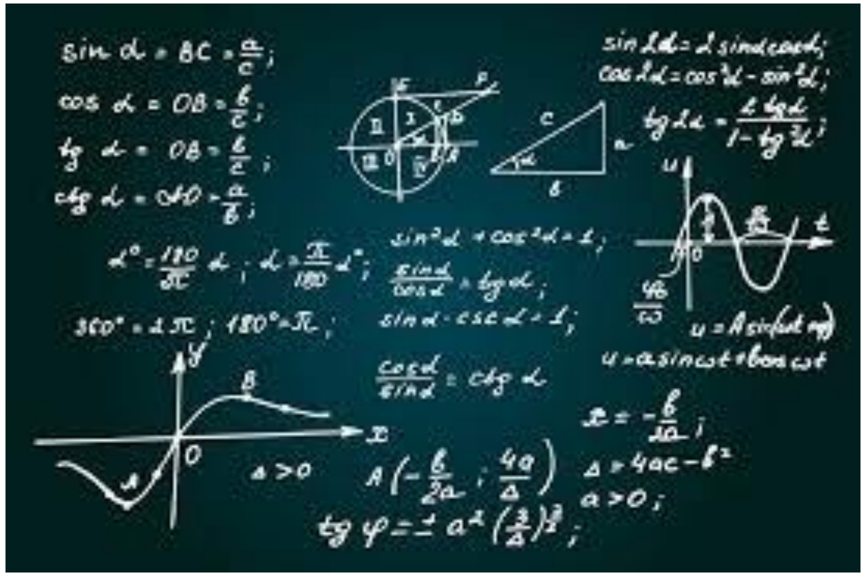
Challenge 1: Popularity of Mathematicians

Basic challenge to test your knowledge of GET requests and querying HTML documents

In this first example we will extend the example of the most popular mathematicians and attempt to combine data from 2 different web-sources to gain insight into the respective number of books of each mathematician in the Toronto Public Library

1. Scrape the names of the 100 most popular mathematicians from the fabpedigree.com website
2. Query the Toronto Public Library website for each mathematician and return the amount of hits found

Upload your code to the github repository including a description file with performance metrics



http://www.fabpedigree.com/james/mathmen.htm

https://www.torontopubliclibrary.ca/search.jsp?Ntt=Isaac%20Newton

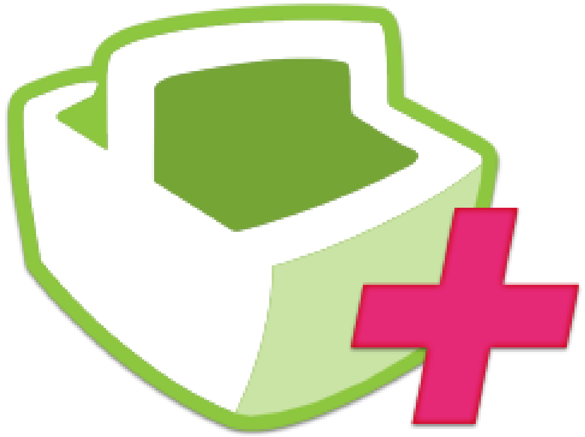
Challenge 2: The supermarket challenge

Doing all the work while Gwen enjoys her holiday

The website listed below contains information on all supermarket special offers in the Netherlands since 2009. This challenge will focus on extracting all relevant data from the

1. Scrape the data related to all offers @ Albert Heijn for the year 2015 from the website as quickly as possible (use a function like time() to measure execution time)
2. Apply your code to scrape the data from other supermarkets
3. Extract only the beer related offers (advanced)

Upload your code to the github repository including a description file with performance metrics



https://www.supermarktaanbiedingen.com/folders/albert_heijn/2017/41

Challenge 3: La Trappe Recipes

Practicing the use of advanced topics, such a Selenium

In this challenge we will use Selenium to scrape the La Trappe website for recipes related to La Trappe beers. As output we require

1. Scrape information on all the La Trappe beers from the website using simple GET requests

2. Use Selenium to build a data frame of every La Trappe beer on the website, and it's related recipes and URL to the recipe. (Tip: Use Selenium to access the filter tab to get the recipes for each individual beer)

Upload your code to the github repository including a description file with performance metrics

https://www.latrappetrappist.com/en/our-trappist-ales/recipes/