DATA SCIENCE CONSULTING
Session 1

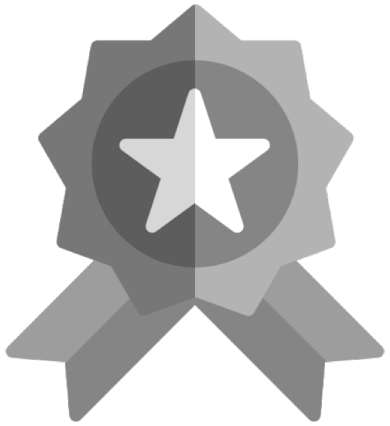
February 3nd, 2020

Agenda

1. **Who are we?**
2. Course modalities
3. Case presentation
4. Analysis objectives & approach
5. Data collection
6. Html presentation & Selectors
7. Scrapping with Scrapy
8. Summary of the session

Capgemini Invent, a leader in digital & data transformation

Strategic focus,
of our firm since 2012, now standing for 50% of our project portfolio



Leader on the market,
with award winning thought leadership, partnership with MIT, and world class recognition

World class footprint,
16 offices, 3,000 consultants, coverage of more than 80% companies of CAC 40 and DAX 30 at CxO level

Acknowledgements of business expertize

ALM

Full spectrum digital specialist, Best-of breed provider - 2016

Gartner

Leader position in the Magic Quadrant for Business Analytics Services 2017

Capital

N°1 consultancy for digital, data and transformation - 2016

source forconsulting.com

Top 3 worldwide thought leadership consultancy - 2016

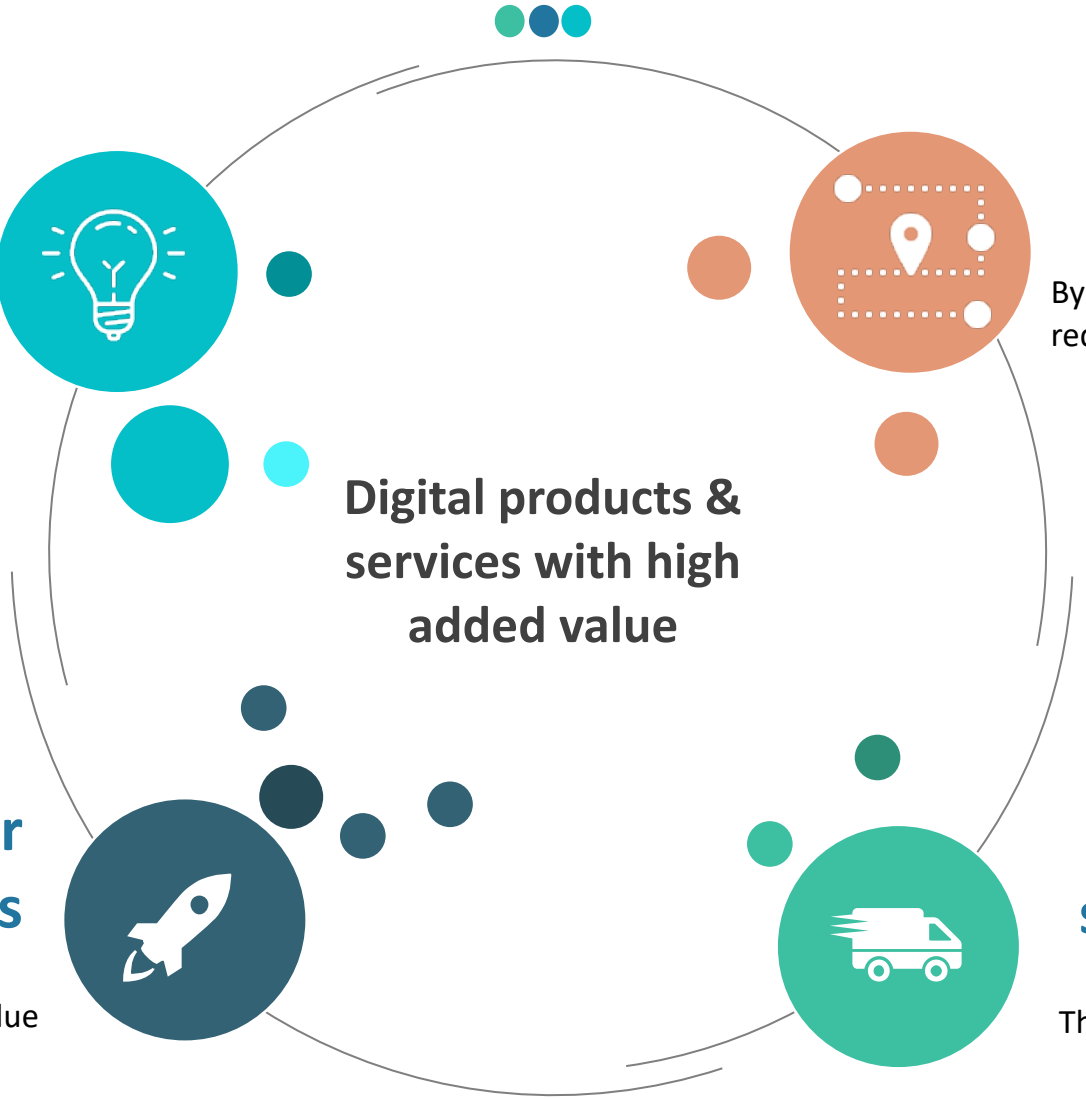
Insight Driven Enterprise, a hybrid agency orchestrating consulting, data science, technologies, creative skills from vision to delivery

We stimulate ideation & disruptive vision

By helping you finding the unconventional ideas to hack your business

We provide end-to-end commitment & delivery

By involving & orchestrating the different required capabilities



We ensure go-to-market for your best ideas

From digital assets to business value

We innovate at startup speed for a fast delivery

Through proven agile and lean methodologies

IDE, a full-speed capability, focused on building data-driven products & services and transforming organization

Global network, gathering more than 120 data scientists in 6 countries:

Skills

- Data Science
- Consulting
- Technology

Partners

- Big data editors
- Specialized players
- Agencies
- Academic institutions

IT platform

- State of the art
- That can be made available for projects

Showroom

- Ability to showcase data projects by creating a dedicated rooms gathering teams, MVPs, demos etc.

Code library

- Gathering of reusable algorithms
- Enabling to accelerate delivery

Data library

- Gathering of several open data sources
- To be integrated to analyses

Viz library

- Gathering of visualizations with interactive examples
- To be used for dashboards building

Use case library

- Lists of use cases for each application areas
- To be used as inputs for use cases identification

The team with whom you will spend your Bootcamp sessions

Olivier AULIARD
Chief Data Scientist
olivier.auliard@
capgemini.com

François LEMEILLE
Data Scientist
francois.lemeille@
capgemini.com

Sofiane MEDJKOUNE
Managing Data Scientist
sofiane.medjkoune@
capgemini.com

Ismail MEBSOUT
Data Scientist
ismail.mebsout@
capgemini.com

Sophie LY
Data Scientist
sophie.ly@
capgemini.com

Johan ATTIA
Data Scientist
johan.attia@
capgemini.com

Maëva DERRIEN
Consultant
maeva.derrien@
capgemini.com

Thomas CLAVIER
Senior Data Scientist
thomas.clavier@
capgemini.com

Thibaud LAMOTHE
Data Scientist
thibaud.lamothe@
capgemini.com

Sim BOZKO
Senior Consultant
sim.bozko@
capgemini.com

Svetlana OLLIVIER
Managing Consultant
svetlana.ollivier@
capgemini.com

Carole RIVIERE
Managing Consultant
carole.riviere@
capgemini.com

Kwassi Beno-Charles DOKODJO
Data Scientist
kwassi-beno-charles.dokodjo@
capgemini.com

Benoit PAYET
Senior Consultant
benoit.payet@
capgemini.com

Khemon BEH
Managing Data Scientist
khemon.beh@
capgemini.com

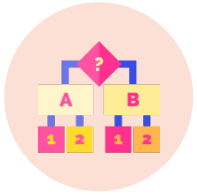
Agenda

1. Who are we?
2. **Course modalities**
3. Case presentation
4. Analysis objectives & approach
5. Data collection
6. Html presentation & Selectors
7. Scrapping with Scrapy
8. Summary of the session

Objectives of the case study

Handle a business problematic associated to data

Increase both knowledge and skills on these topics



Learn how to determine & realize the required analysis

Handle a data project from the beginning to the end

Understand the strategic & transformation stakes

Qualify and quantify the associated stakes

Grasp the consulting aspects

Learn how to manage these kinds of projects

Organization of the course

Content

- Each session focused on a **specific topic**
- **Alternation** between theoretical explanations, brainstorming and applications
- **Homework** to be prepared between sessions and to be presented in the beginning of the next session

Format

- **Weekly sessions on Monday and Friday afternoons** – from 2pm to 6pm
- **Within Capgemini Invent's** offices (le 147)– except for Mar. 02 & Mar. 13 which will take place within Ecole Polytechnique's campus
- **Animation by Data Scientists & Consultants** from Insight Driven Enterprise (IDE)

Practical organization

- **Teamwork :** divided in 7 groups of 5, each group assigned to a coach
- **Contact via Slack** https://nlpinvent2020.slack.com in order to
  - **Discuss** among yourselves
  - **Ask questions** to coaches
  - **Get access to all the questions & answers** – including from the other students & groups

Planning & key steps of the course

7 courses

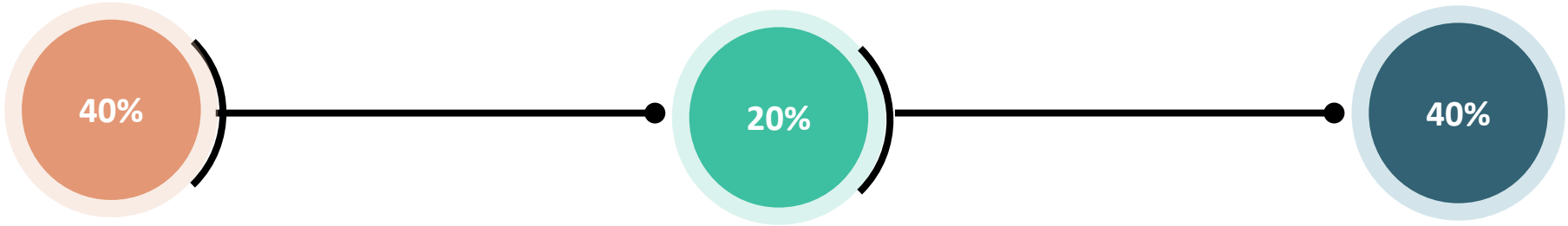
Feb 03.	Feb. 10	Feb. 24	Mar. 02	Mar. 06	Mar. 09	Mar. 13
Session 1	**Session 2**	**Session 3**	**Session 4**	**Session 5**	**Session 6**	**Session 7**
Kick-off	**Data Prep.**	**Embedding**	**SteerCo**	**Embedding**	**Sentiment**	**Final presentation Hackathon**

Legend :

Regular sessions: Synthetic status update on the progress made and foreseen objectives to be prepared

Committees: Complete pres. of the progress made, results; difficulties and foreseen next steps to be prepared

Evaluation through the class



Homeworks

At each session you might have homeworks. We'll ask you to send us yout results and will evaluate them !

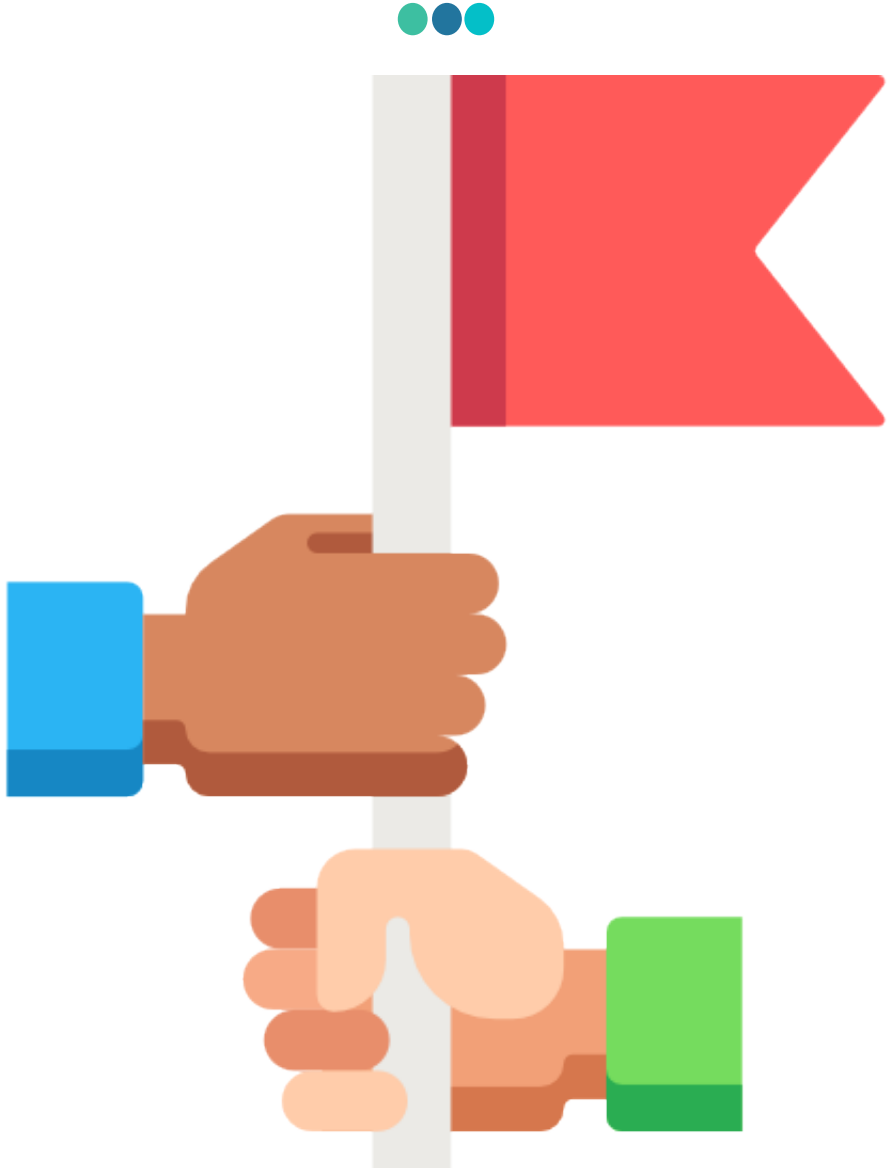
1st restitution

At the 4th session, we'll simulate a client meeting with intermediary results. It will be a presentation where you have to summarize what you've seen/done until then.

Final restitution

Last day you'll have a final presentation where you'll present the business insights extracted from your data Project.

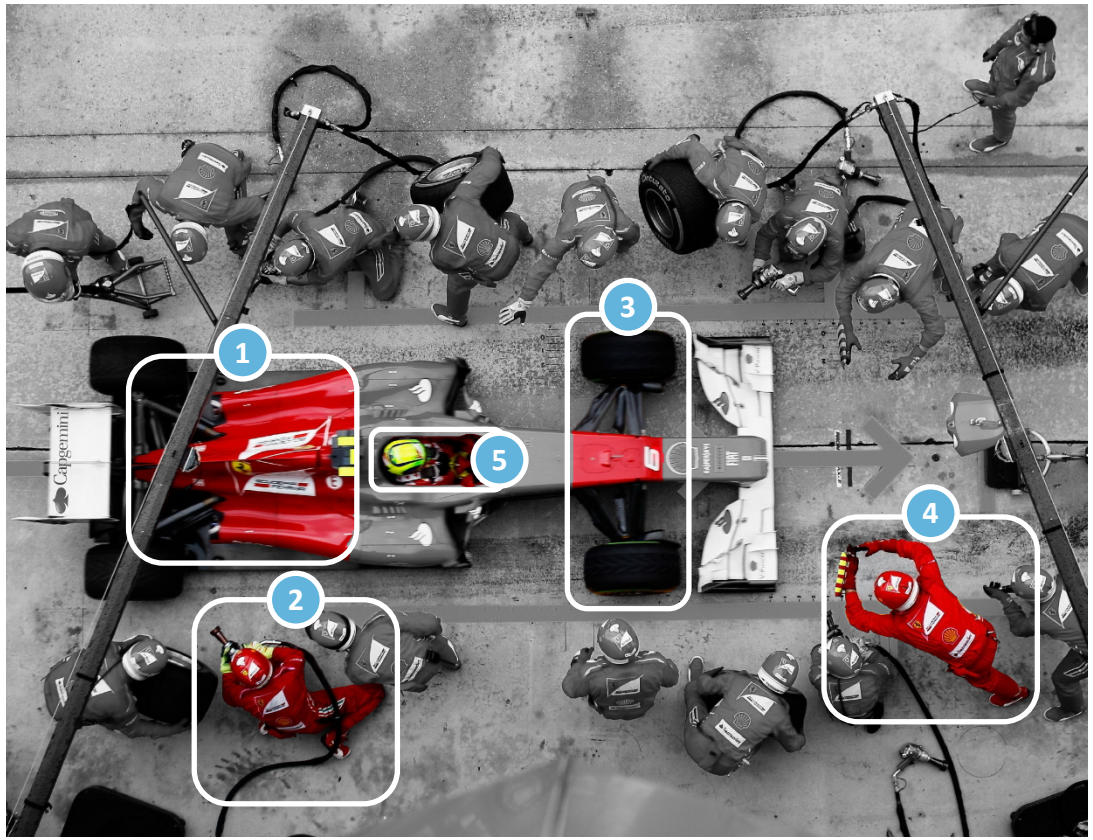
What are your expectations for this course?



Agenda

1. Who are we?
2. Course modalities
3. **Case presentation**
4. Analysis objectives & approach
5. Data collection
6. Html presentation & Selectors
7. Scrapping with Scrapy
8. Summary of the session

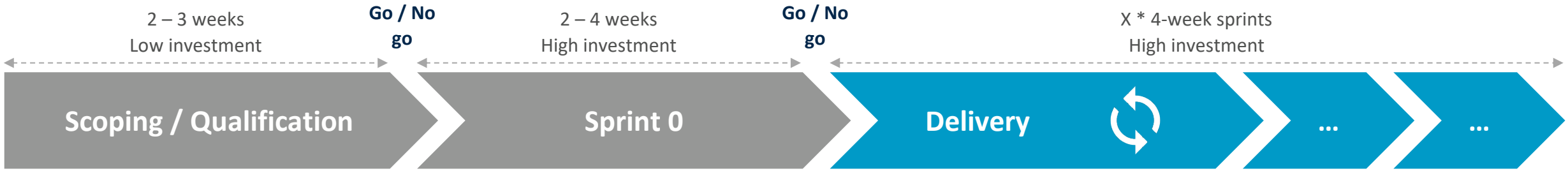
First of all… what is a data use case?



1. **Motor**
   Robust technical platforms
2. **Fuel**
   Comprehensive data sources
3. **Tires, wheel**
   Digital solutions & touchpoints
4. **Technical team**
   Advanced analytics capability
5. **Driver**
   Identified end-users

Typical approach for use case delivery projects

The operating model is based on a progressive approach to ensure the correct product delivery



Is the project **mature enough**?

Is **everything ready** to launch the project?

Delivery of industrialized data products

PoC / PoV

If necessary, launch of a PoC / PoV to prove the project feasibility / value

Presentation of the client

Multinational hospitality group founded in 1967 in France

Covers variety of segments : luxury, midscale and economy
Subsidiaries in events organization and digital hospitality (catering, coworking…)

Operates in 100 countries – 4800 hotels and 280 000 employees worldwide

Rebranding strategy since 2011 on existing assets and new acquisitions

« Life styled for you » - Complementarity between group brands for better customer experience

Focus of case study : Novotel Canary Wharf

Rebranding of Bokan 39

The bar & restaurant above our Novotel in Canary Wharf, London with the goal of increasing its number of customers

- **Use data to make the right choices** for the rebranding of the space
- Suggest a **new branding** and type of **cuisine**
- Estimate the cost, benefits and **breakeven** point

Agenda

1. Who are we?
2. Course modalities
3. Case presentation
4. **Analysis objectives & approach**
5. Data collection
6. Html presentation & Selectors
7. Scrapping with Scrapy
8. Summary of the session

Group constitution

5’

Divide the class into 7 balanced groups

The group should include different school backgrounds

The group should include a gender mix

Let’s create the groups!

7 Groups of 5 students

Group 1

- Cevizci BERFIN
- Andrea BACCONI
- Adrien TOULOUSE
- Paul-Antoine GIRARD
- Samuel SOUCI

Group 2

- Paul-Emile DUGNAT
- Constantin VODÉ
- Arthur DUCASSE
- Etienne WINDELS
- Sylvanus MAHE
- Victoire de TERMONT

Group 3

- Jiayu GAN
- Honghao YU
- Qiwen ZHAO
- Delong LI
- Doha KADDAF

Group 4

- Charles CAZALS
- Jean CHILLET
- Antoine DEMEIRE
- Katrin DIMITROVA
- Alexandre LEBOUCHER

Group 5

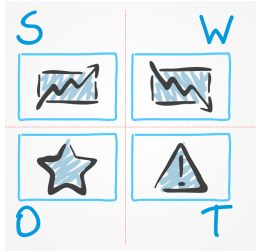
- Abderrahmane LAZRAQ
- Sarah JALLOT
- Leonardo NATALE
- Leon LEITAO
- Naomi SERFATY

Group 6

- Tommy TRAN
- Thomas de MAREUIL
- Akhila VANGARA
- Ismail MAJJAD
- Badr El IDRISSI MOKDAD

Group 7

- Corentin SENE
- Arthur KRIEFF
- Gabriel PERSOZ
- Romain BESOMBES
- Eva FRANÇOIS



SWOT Analysis, what's about ?

Definition

A **SWOT analysis** is a framework that is used to analyze a company's competitive positioning in its business environment.

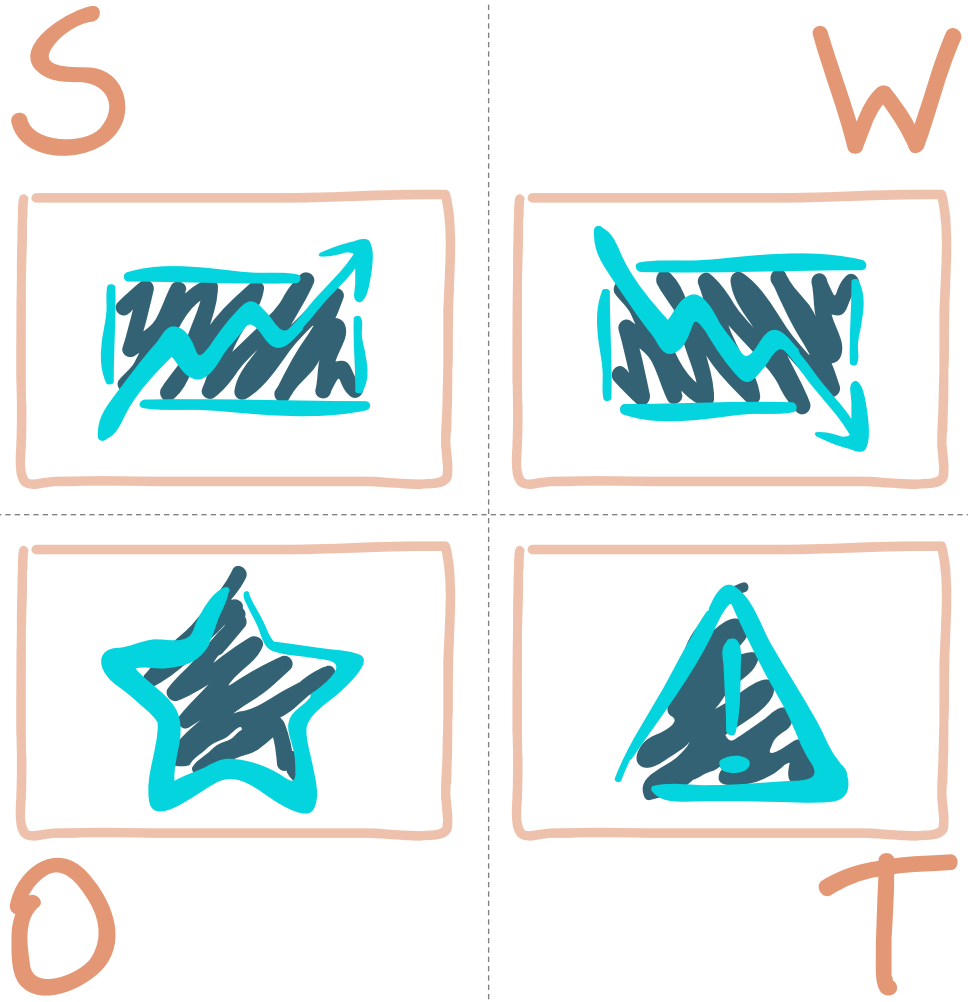
The main objective of the SWOT analysis is to help in **identifying the strategies** that can be used by the company to build on its strengths, eliminate its weaknesses while making the most of opportunities and countering threats.

Context

An **interactive process** needs to be undertaken by **coordinating among all the departments of the firm** such as finance, marketing, operations, human resource, logistics, strategic planning, management information systems etc. to conduct a SWOT analysis

SWOT Analysis, what's it about ?

INTERNAL	STRENGTH	WEAKNESS
EXTERNAL	OPPORTUNITIES	THREAT



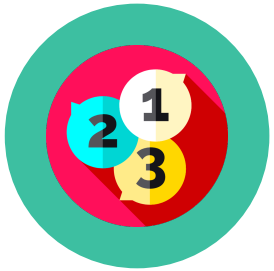
Agenda

1. Who are we?
2. Course modalities
3. Case presentation
4. Analysis objectives & approach
5. **Data collection**
6. Html presentation & Selectors
7. Scrapping with Scrapy
8. Summary of the session

Data pipeline

Data Collection

Data Cleaning



Word Embedding

Topic Extraction

Sentiment Analysis

Data Channels

Database & SQL

APIs

Web Scrapping

Application Programming Interfaces

Available Websites APIs

Many Website provide their APIs for Data collection

Few lines of code

The main advantage of using APIs is that they involve much less programming than scraping

```python
from twython import Twython


CONSUMER_KEY= 'CRqjrVd0ZW5ADisG3KIW3ILZ0'
CONSUMER_SECRET= 'wIaRhn4NeENTal8eQkX4Ww3JbHRcTQfDa3A4ddb4WAffAIVEjr'
twitter = Twython(CONSUMER_KEY, CONSUMER_SECRET)


for status in twitter.search(q="data science")["statuses"]:
    user = status["user"]["screen_name"].encode('utf-8')
    text = status["text"].encode('utf-8')
    print(user, ":", text)


```

Structured Output

The output is in general a JSON file which can be easily turned into a database

```json
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```


Well known social network APIs

Policies might vary on different platforms

- No direct scraping unless authorized (fill in authorization form – 2 weeks for FB answer). See the terms.
- APIs exist for app developers

- Limited collection of the data (speed/volume is compared to what « human can reasonably produce »), it allows read & write operations on videos with a limited quota
- No personal data

- No direct scraping
- API with limited number of call by 15 min window

- Prohibition of scraping software (check internet news for their law suits)
- APIs are for app development

- The free API is for “non-automated” apps, user authorization needed, Python/Ruby versions exists, 5000 calls per hour, it is being depreciated in favor of the new “Business version” (Instagram Graph API)
- Sensitive to user content/media (owned by users)

Getting information from the web

Web Scraping

The art of extracting information on a specific topic from the Internet using automating requests

Why should we use Web scraping?

- Market Price Analysis (real-time competitiveness)
- Market Intelligence
- Sentiment Analysis (social reactions)

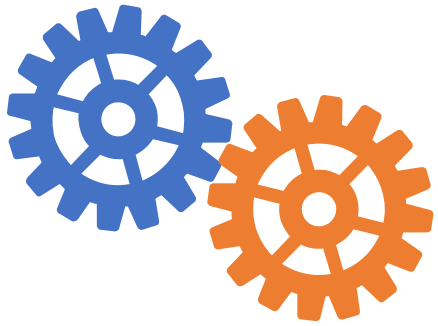
Which kind of Data are we looking for?

- Price / Articles / Reviews
- Metadata (number of connections, …)
- Find sources we can use (API, Scraping)

What is the typical process?

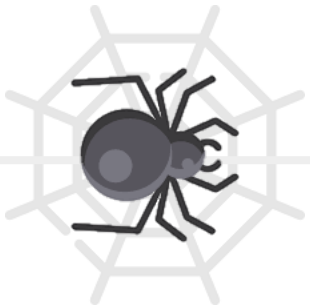
- Standardize data frame structure
- Build a Mapping (Excel, Trello)

Two main blocks of Scraping



Parsing

Breaking down the scraped data into smaller bits to understand it



Crawling

Going through specified website and related links, to get raw data (source code)

Scraping's Main tools

BeautifulSoup

HTML parser in Python, which transforms HTML code into a tree object usable with Python to extract content like text, images' URLs…

Scrapy

Python package combining a crawler AND a parser.

Selenium

Tool allowing to automate web browsers, mainly used in scraping to get data which is written in JavaScript through AJAX requests.

RVest

HTML parser like BeautifulSoup, but for R. Great documentation online.

Scrapy

In this course we will focus on **Scrapy** exclusively.

Agenda

1. Who are we?
2. Course modalities
3. Case presentation
4. Analysis objectives & approach
5. Data collection
6. **Html presentation & Selectors**
7. Scrapping with Scrapy
8. Summary of the session

How does the web work (1/2)?

First, some internet culture…

What are the differences between the Internet and the Web ?

Internet

« **inter**connected **net**work »

Global system of interconnected computer networks using the internet protocol suite to link devices worldwide

The Web

Information space where documents and other resources are identified by Uniform Resource Locator (URLs), interlinked by hypertext links, and accessible via the Internet

Web Browser

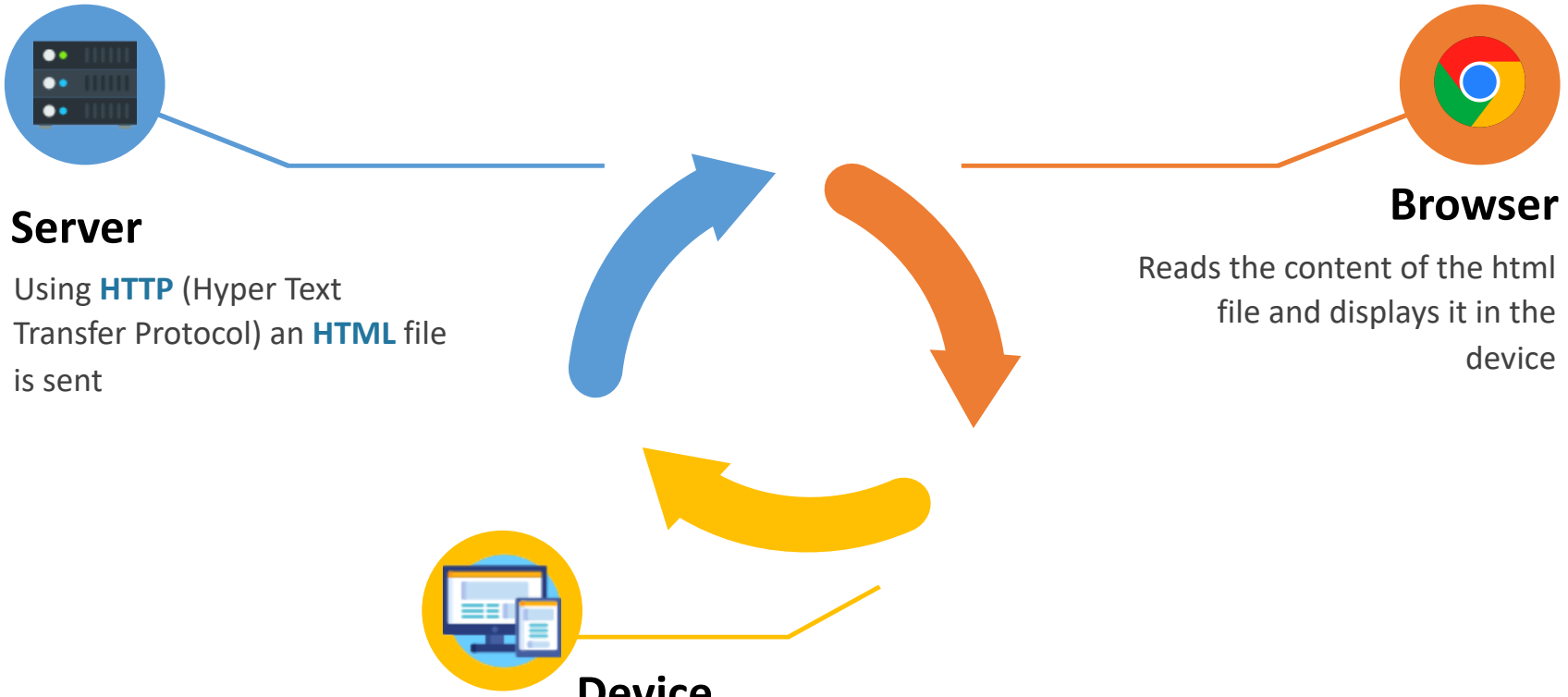
Software application used to « surf » the Web.

Ex: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge…

The web was built using a markup language, **HTML** (HyperText Markup Language), invented in the early 90's.
Understanding **HTML** is useful to learn how to scrape properly the web.

How does the web work (2/2)?

How the web works : summary



Server

Using **HTTP** (Hyper Text Transfer Protocol) an **HTML** file is sent

Browser

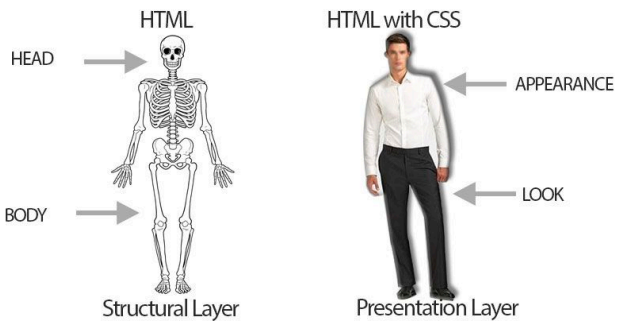
Reads the content of the html file and displays it in the device

Device

Requests web pages (originally HTML files) using their URLs, a kind of a « web address »

Structure of a web page

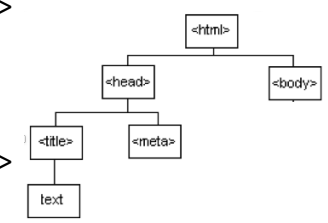
HTML source code




```
<!DOCTYPE html>
<html>
  <head>
        <meta charset="utf-8" />
        <title>Titre</title>
  </head>

  <body>

  </body>
</html>
```




```
<tag attribute="valeur"> … </tag>
```

text tags	
	emphasize with italics
	emphasize with bold
	largest headline
	second largest headline
	smallest headline
<font size="?">	numbers 1-7 to determine size
<font color="?">	set font color w/ name or hex code
	subscripts:
	superscripts[1]
<del></del>	~~strikethrough~~
<code></code>	`insert codes`

image tags	
<img src="URL">	embed an image
title="?"	name an image
alt="?"	an alternate name for an image

table tags	
<table></table>	create a table
<th></th>	create a header row
<tr></tr>	create a row
<td></td>	create a cell
<table border="?">	sets the table border
<table cellspacing="?">	sets the spacing around cells

Structure

- A web page is constituted of two main elements: HTML and CSS:
  - HTML is the backbone which contains text arranged into blocks, which have attributes.
  - CSS is mainly used for the style of the webpage

Sequence

- It is a sequence of html tags which can be seen as a tree

Tags

- Each tag has a specific format
- There are several attributes per tag: class, href, etc.

Scraping web pages: Structure of a web page

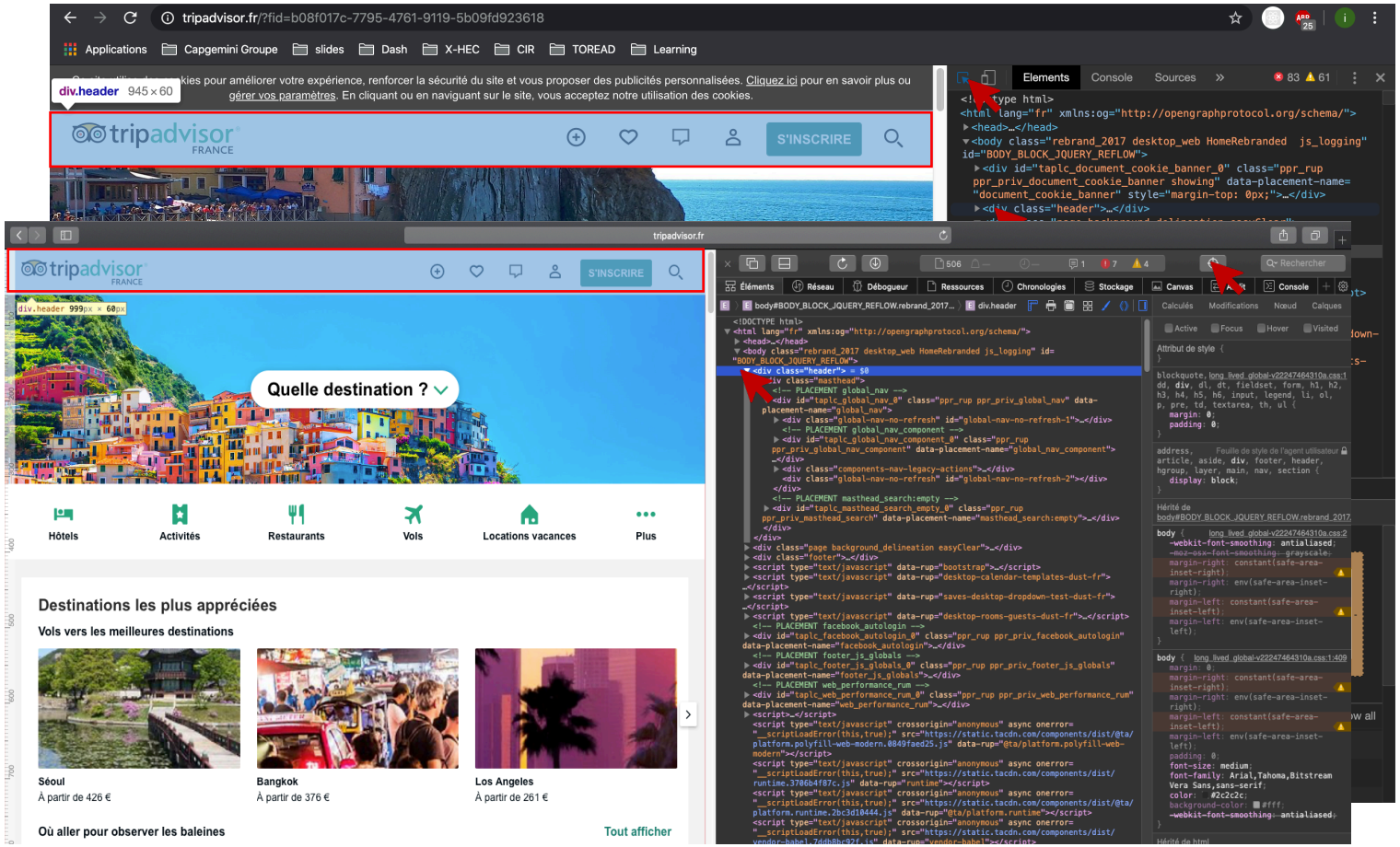
HTML source code

- Scraping packages enables you to extract information from HTML pages and to build a structured data set.

- Some browsers (Chrome, Firefox…) have an “Inspect mode”, which is interactive:

CTRL + SHIFT + I

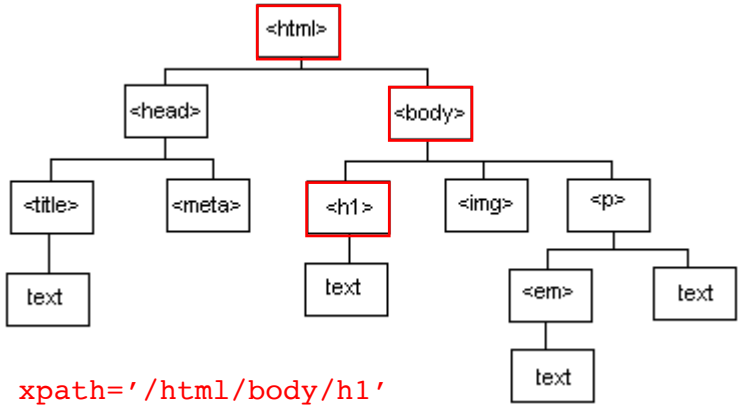
OPTION + CMD + I



Introduction to XPath

What is XPath ?

- Xpath is a string which selects nodes in an html tree
- It can also be seen as the linear representation of the requested element



xpath='/html/body/h1'

- XPath allows you to select:
  - Content of a markup
  - Content of its attributes (hypertext links for example)

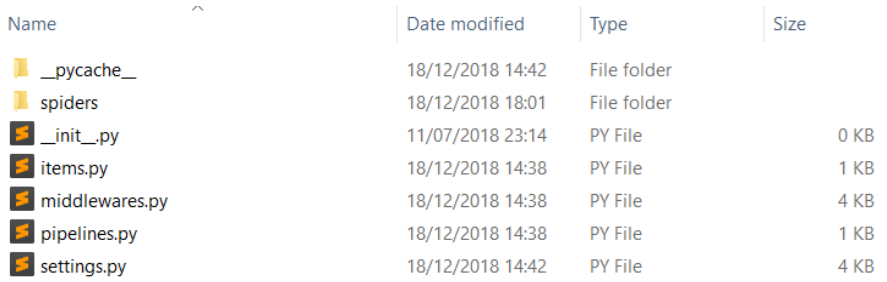
Requested Element	Corresponding Xpath
Second division of the body	'/html/body/div[2]' (index start at 1)
All tables	'//table' #all tables
All tables descendants of the 2nd division	'/html/body/div[2]//table'
All paragraphs directly bellow the body	'//p'
Conditional division	'//div[@id="uid"]'
Wildcard	'/html/body/*'
All elements with a condition on the class	'//*[contains(@class, "class-1")]'
Selection of the attribute href cond. paragraph	'//p[@id="p2"]/a/@href'

Scraping with Scrapy: New Project

- We can create a new project using the following command line:

```
scrapy startproject project_name
```

- This will create a directory with different files and subfolders:
  - *__pycache__* : contains the python cache
  - *spiders* : contains the spiders used for scraping
  - *__init__.py* : project initialization file
  - *items.py* : project items definition file
  - *middlewares.py* : project middlewares file
  - *pipelines.py* : project pipelines file
  - *settings.py* : project settings file



- Having already defined a spider in a .py file stored in the foler « spiders », we can put it to work using the following command line:

```
scrapy crawl spider_name
```

Introduction to CSS Locator

What is a CSS Locator ?

- If the xpaths allow the selection of a node using the html tree, the css locators on the other hand access the node using the css attributes
- The following tables compares the xpaths and the css locators

XPath	CSS Locator
'/html/body/div'	'html>body>div'
'//div/span//p'	' div > span p'
'//div/p[2]'	' div > p:nth-of-type(2)'
'/html/body//div/p[2]'	'html > body div >p:nth-of-type(2)'
'//div[@id="uid"]'	' div#uid'
'//p[@class="class-1"]'	' p.class-1'
'//div[@id="uid"]/a/@href'	'div#uid >a::attr(href)'
'//p[@id="p-example"]/text()'	'p#p-example::text'
//p[@id="p-example"]//text()'	'p#p-example ::text'

Introduction to Selector


```
text_html='''

<html>
    <head>
          <meta charset="utf-8" />
          <title>Titre</title>
    </head>

    <body>

         <div>
             <p>Hello World!</p>
         </div>

         <div>
             <p>Enjoy Scraping!</p>
         </div>

    </body>
</html>
'''
```

What is a selector ?

- A selector is a pyton element imported from scrapy which allows to extract the content of the requested node
- The selection of node is carried out using a xpath
- Using the html structure on the left, we can exctract all the paragraphs:

```
From scrapy import Selector
Sel=Selector(text=text_html)

Sel.xpath('//p').extract()
      ['<p>Hello world!</p>', '<p>Enjoy</p>']

Sel.xpath('//p/text()').extract()
      ['Hello world!', 'Enjoy']

Sel.xpath('//p/text()').extract_first()
      'Hello world!'
```

- Selectors allow chaining:

```
Sel.xpath('/html/body/div[2]') == Sel.xpath('/html').xpath('./body/div[2]')
```

XPath	CSS Locator
Sel.xpath('div/p')	Sel.css('div > p')

Python Set-up

5’

Install Anaconda – Use provided requirements.txt file to install your libraries

Hands-on 1

40’

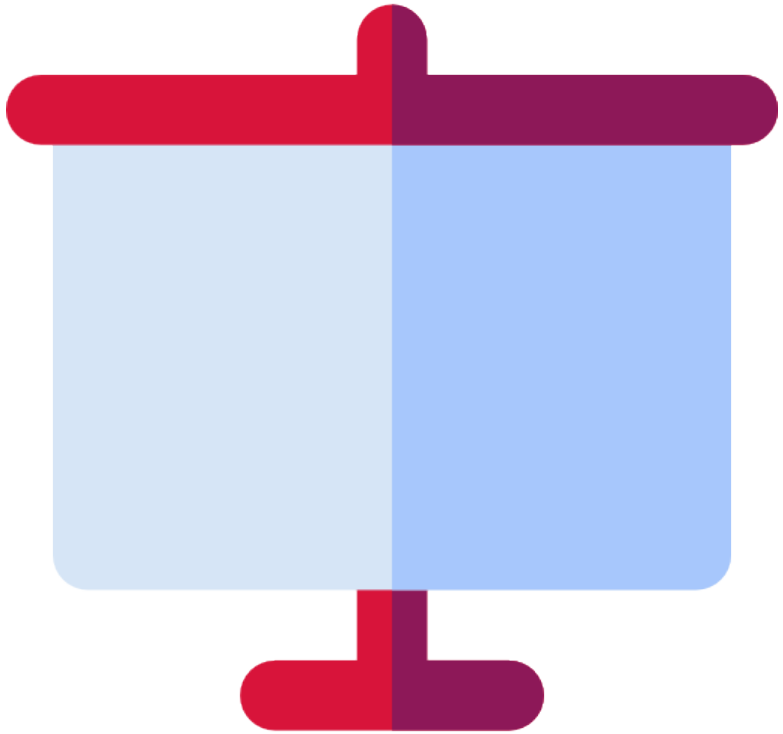
Use the notebook 1 to discover selectors and get some information from the web

If you have any question about Python set-up, feel free to contact us about that !

Restitution

10’

Could you extract data ? What was challenging for you ?



Break

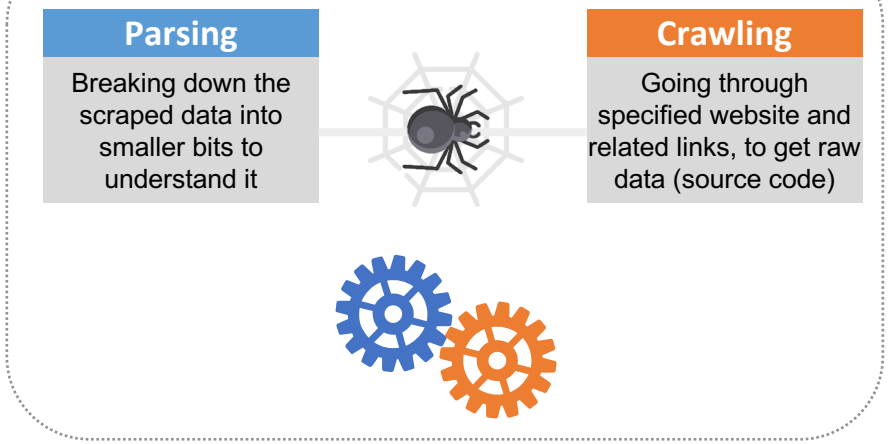
15’

Feel free to help yourself ! See you at 16h30 !

Agenda

1. Who are we?
2. Course modalities
3. Case presentation
4. Analysis objectives & approach
5. Data collection
6. Html presentation & Selectors
7. **Scrapping with Scrapy**
8. Summary of the session

Scraping with Scrapy: Introduction



Scrapy

- Scrapy is a powerful Python scraping package to scrape which combines a crawler and a parser, so it's pretty complete for a scraping project
- Scrapy works with spiders, which are classes that you define and that Scrapy uses to scrape information from a website (or a group of website).

```python
class QuotesSpider(scrapy.Spider):
    name = "quotes"

    def start_requests(self):
        urls = [
            'http://quotes.toscrape.com/page/1/',
            'http://quotes.toscrape.com/page/2/',
        ]
        for url in urls:
            yield scrapy.Request(url=url, callback=self.parse)

    def parse(self, response):
        page = response.url.split("/")[-2]
        filename = 'quotes-%s.html' % page
        with open(filename, 'wb') as f:
            f.write(response.body)
        self.log('Saved file %s' % filename)
```

Spider

- They must define the initial requests to make, optionally how to follow links on pages, and how to parse the downloaded page content to extract data.
- You can write a spider inside a .py script, or you can use Jupyter's notebook.

Documentation: https://docs.scrapy.org/

Scraping with Scrapy: Data Storage

- The easiest way to store your scraped data is by using Feed exports, with the following command:

```
scrapy crawl spider_name –o file.json
```

- Saving your data using the JSON format will only **work once**, since Scrapy will append to a given file instead of overwriting its contents.
- If you run the command twice without deleting the file before the second time, you'll end up with a broken JSON file.

- You can also use JSON Lines format (file.jl) instead, which doesn't have the same problem of JSON if you run twice.

Scraping with Scrapy: Spiders as a class

- Before getting into spiders we first need to define the structure of a python's class which is the basic structure of the spider itself
- A python class allows to
  - Creates new objects
  - Defines the internal structure of an object
  - Defines the methods of an object

Attributes

Methods

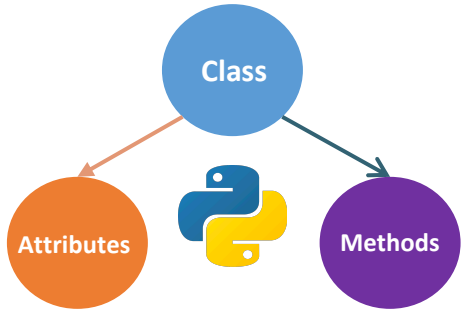
```
class class_name:

    def __init__(self, var1, var2):
        self.attr1=var1
        self.attr2=var2

    def method1(self, x):
        …
        return …

    def method2(self, y):
        …
        return …

object = class_name(var1, var2)
print(object.attr1)  # returns var1
print(object.method1(x))
```



General form of a spyder

```
Class
SpiderClassName(scrapy.Spider):
    name = "spider_name"
    # the code for your spider
    # what website?
    # which techniques
```


Scraping with Scrapy: Spiders

```python
class QuotesSpider(scrapy.Spider):
    name = "quotes"

    def start_requests(self):
        urls = [
            'http://quotes.toscrape.com/page/1/',
            'http://quotes.toscrape.com/page/2/',
        ]
        for url in urls:
            yield scrapy.Request(url=url, callback=self.parse)

    def parse(self, response):
        page = response.url.split("/")[-2]
        filename = 'quotes-%s.html' % page
        with open(filename, 'wb') as f:
            f.write(response.body)
        self.log('Saved file %s' % filename)
```

- We define our spider with the subclass scrapy.Spider, and defines some attributes and methods:
  - `name`: identifies the Spider, must be unique within a project
  - `start_requests()`: must return an iterable of Requests which the Spider will begin to crawl from.
  - `parse()`: method that will be called to handle the response downloaded for each of the requests made.

- The `parse()` method usually parses the response, extracting the scraped data as dicts and finding new URLs to follow and creating new requests from them.

Scraping with Scrapy: Crawling

- Now that you know how to extract data from one page, we need to see how to follow links from one page to another, so that you'll be able to scrape a full website on your own !
- Crawling using Scrapy is made of 3 steps:
  - Identiyfing the link you want to follow with `Selectors` in the Scrapy Shell (or using tools like SelectorGadget…)
  - Extracting this link using the `.extract()` method
  - Modify the spider to follow the extracted link by adding it to your `parse()` method:

```
next_reviews_page_url = "https://www.tripadvisor.com" + response.xpath(
    "//a[contains(@class,'nav') and contains(@class,'next') and contains(@class,'primary')]/@href").extract()[0]
if next_page is not None:
    next_page = response.urljoin(next_page)
    yield scrapy.Request(next_page, callback=self.parse)
```


Hands-on 2

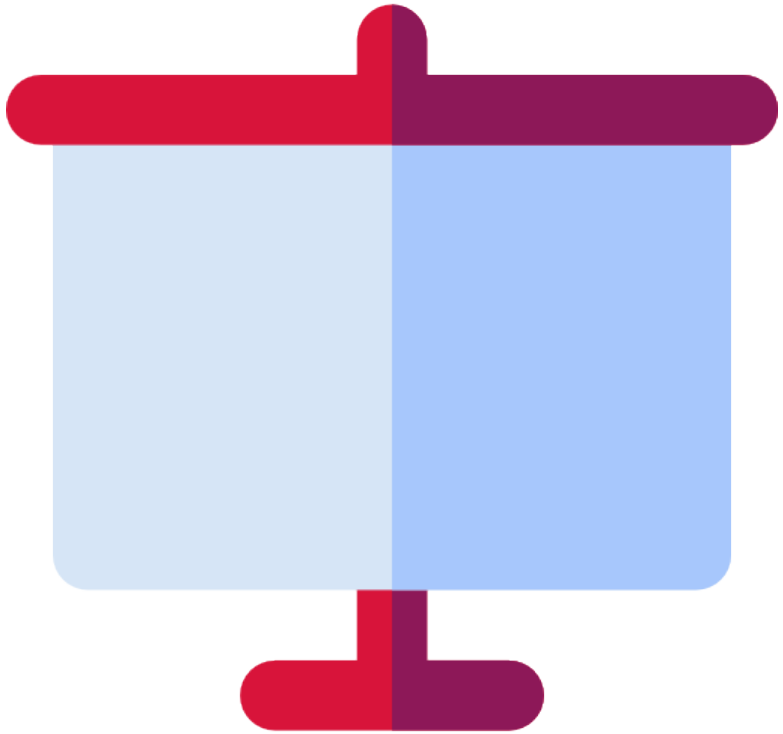
30’

Try to scrap TripAdvisor – You can get inspiration from *x_actu_spider* !

Restitution

5’

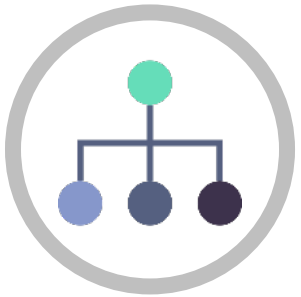
Could you extract data ? What was challenging for you ?



Agenda

1. Who are we?
2. Course modalities
3. Case presentation
4. Analysis objectives & approach
5. Data collection
6. Html presentation & Selectors
7. Scrapping with Scrapy
8. **Summary of the session**

Summary of the session – To remember



Approach & organization of a data science consulting project

Typical approach of this type of project

- Data science workstream: Data scraping, cleansing & feature engineering, running of the different analyses, restitutions (visualizations etc.)
- Business workstream: Diagnosis of the current situation & transformation stakes, quantification of the impacts

Organization & governance :

- Several dedicated meetings all along the project (e.g. weekly status updates, steering committees) to track progress and escalate potential issues

Key steps for data scraping

Necessary tools:

- Scraping algorithms (Python: Scrapy, Selenium ; R: Rvest)
- Web solutions (ParseHub and Import.io)

Data sources

- Web pages (HTML)
- APIs (Google, Yelp etc.)

Extract formats:

- HTML, HTML nodes
- Json

Work for next week

Instructions

To practice what we learnt today, for next week, you'll have to :

- Prepare a SWOT analysis of the current Boken 39
- Create a spider which gets reviews and ratings from multiple pages of reviews for a given restaurant
- As a reminder, you must be able to link a review to its restaurant
- It might be interesting to get other information available on the webpages
- Bonus (+1 pt on final mark) : create a spider which crawls multiple pages of restaurants to get multiple pages of reviews. At least 500 full reviews per restaurant for 100 restaurants.

We expect you to send your code and scraped file by **Friday 7th evening** to *thibaud.lamothe@capgemini.com* and *ismail.mebsout@capgemini.com*

If you have any questions, feel free to contact us through the slack channel or by email.

Course evaluation

Did you like that first course ? It’s time to share your feedbacks !

Thank you for your attention

See you next week @147

GOODBYE !

February 3rd, 2020