HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

IT4142E
Introduction to Data Science

Chapter 2: Data scraping

Lecturer:
Muriel VISANI: murielv@soict.hust.edu.vn

Acknowledgements:
Khoat Than
Viet-Trung Tran

Department of Information Systems
School of Information and Communication Technology - HUST

Contents of the course

- Chapter 1: Overview
- Chapter 2: Data scraping
- Chapter 3: Data cleaning, pre-processing and integration
- Chapter 4: Introduction to Exploratory Data Analysis
- Chapter 5: Introduction to Data visualization
- Chapter 6: Introduction to Machine Learning
  - Performance evaluation
- Chapter 7: Introduction to Big Data Analysis
- Chapter 8: Applications to Image and Video Analysis

Contents of the course

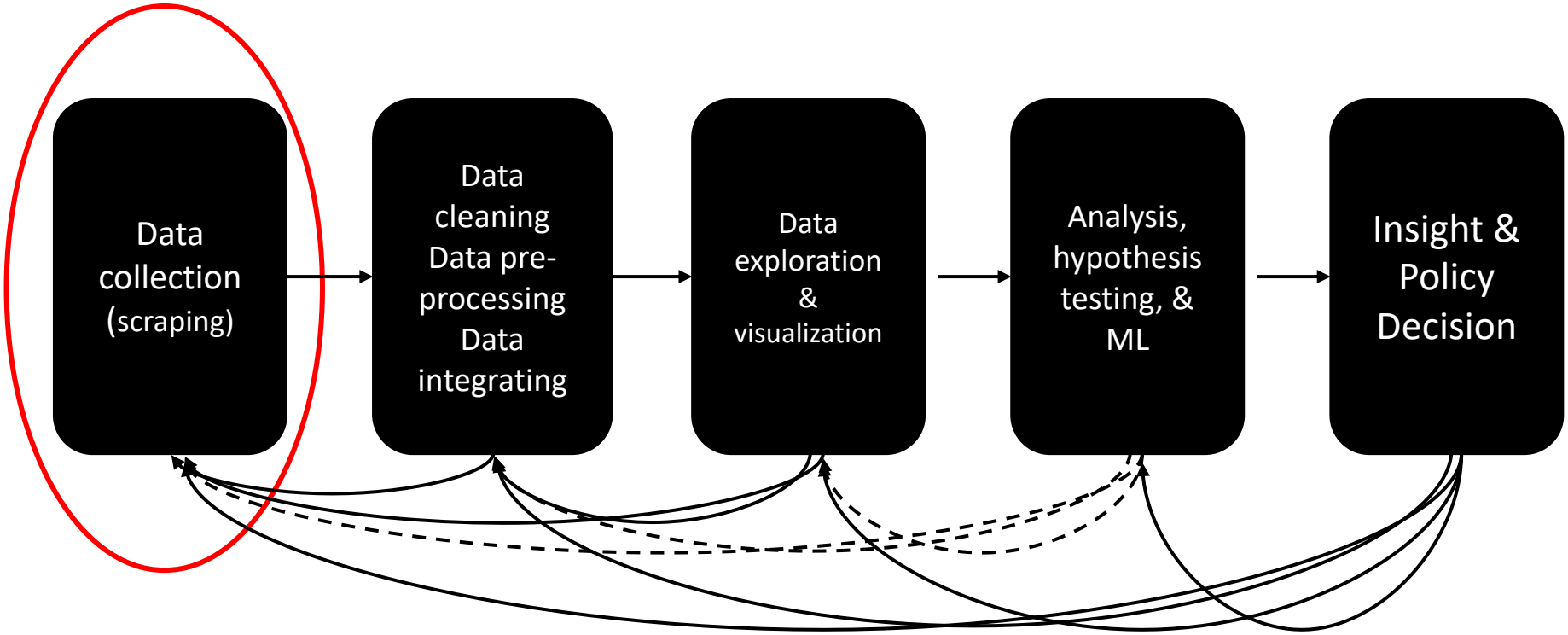
- Chapter 1: Overview
- Chapter 2: Data scraping

Contents

- **Data Scraping**
  - Introduction
  - Definitions
    - Data scraping, screen scraping, report mining, web scraping
    - Web-crawling *vs.* web-scraping
  - Web crawling
    - Some principles
  - Web scraping
    - Some principles
    - Techniques
  - Practice
    - web scraping with *Web Scraper*
    - web scraping with *Scrapy*
  - Exercises / homework

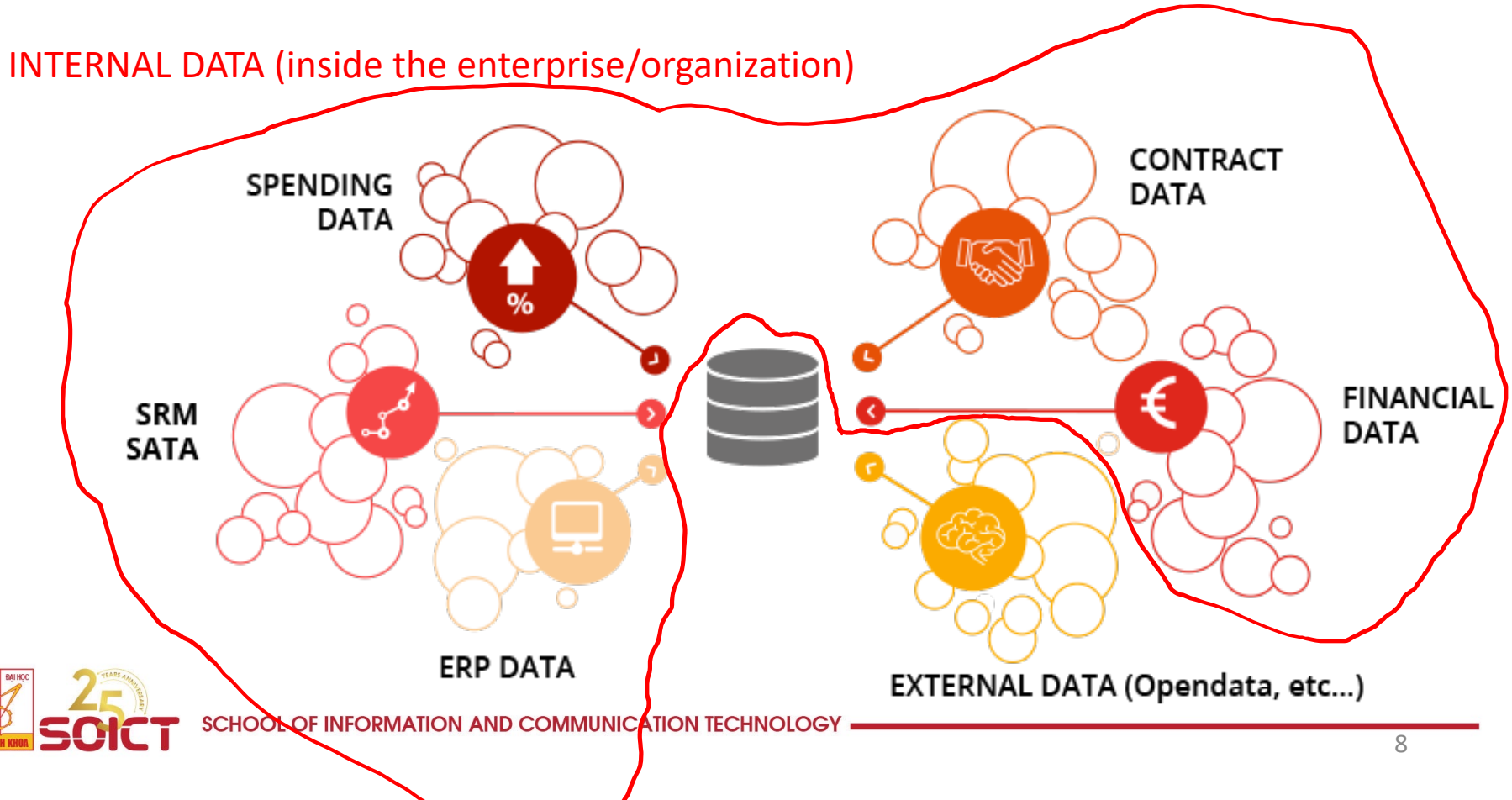
Introduction

Recall: insight-driven DS methodology



What is Data Scraping?

- First (crucial) step of data science
  - Gathering all relevant data for future data analysis
    - Both internal and external data



How to access the data?

- Everything depends on the type of source!
  - Internal data:
    - Databases, data warehouses
    - Flat files
      - Structured (Excel, log files..)
      - Unstructured
        - Readable by humans
          - Text
          - Screen display
          - Reports
  - External data
    - API (SOAP or REST): https://youtu.be/bPNfu0IZhoE
    - Flat files (same as internal data)

Examples of open-source data

- Global Health Facts (www.globalhealthfacts.org/)—Health-related data about countries in the world
- UNdata (http://data.un.org/)—Aggregator of world data from a variety of source
- World Health Organization(www.who.int/research/en/)—Again, a variety ofhealth-related datasets such as mortality and life expectancy
- OECD Statistics (http://stats.oecd.org/)—Major source for economic indicators
- World Bank (http://data.worldbank.org/)—Data for hundreds of indicators and developer-friendly
- Census Bureau (www.census.gov/)—Find extensive demographics here.
- Data.gov (http://data.gov/)—Catalog for data supplied by government organizations. Still relatively new, but has a lot of sources.
- Data.gov.uk (http://data.gov.uk/)—The Data.gov equivalent for the United Kingdom.
- data.gouv.fr (http://data.gouv.fr) — The French equivalent
- DataSF (http://datasf.org/)—Data specific to SanFrancisco.
- NYC DataMine (http://nyc.gov/data/)—Just like the above, but for New York.
- ParisData (http://opendata.paris.fr) — Paris
- OpenData La Rochelle (https://opendata.larochelle.fr/) — La Rochelle

Accessing the data via APIs

- Numerous services are available online *via* http
- Data transfer is made following a **protocol** ruling exchange formats
- Such exchanges are well-structured, well-documented, easy to parse (*a.k.a.* analyze) and minimize ambiguities
- Such exchange formats are usually difficult to read for a human
- Protocols used
  - SOAP: structured using XML
  - REST: often structured using JSON
- Example: about shared bike stations in La Rochelle:
  - https://opendata.agglo-larochelle.fr/visualisation/table/?id=2dcdf9ac-ca89-4e3f-af7e-5d75f255ee5e

Accessing the data via log files

- Log files contain a sequential history of events in a process
  - API, network activity…
- Log entries are generally time-stamped and in chronological order
- Enable to analyse the process's activity and its interactions with its environment

```
Jul 30 01:44:00 maltsev syslogd[56]: ASL Sender Statistics
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: getInterfacesInternal File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 1715 missing PPP destination address for
interface "utun0".  Check profile PPPExclusion (set to Automatic?) or contact your
administrator.
Jul 30 01:44:00 maltsev acvpnagent[65]: A network interface has gone down.
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: logInterfaces File:
../../vpn/AgentUtilities/Routing/InterfaceRouteMonitorCommon.cpp Line: 477 IP Address Interface
List: FE80:0:0:0:6:97F:A26C:21B1 192.168.1.109 FE80:0:0:0:60D3:1E91:4FC8:29E
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: getInterfacesInternal File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 1715 missing PPP destination address for
interface "utun0".  Check profile PPPExclusion (set to Automatic?) or contact your
administrator.
Jul 30 01:44:00 --- last message repeated 1 time ---
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: GetPrimaryInterfaceIndex File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 501 Unable to get global IPv6 information
from system configuration [error 1004].
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: updatePotentialPublicAddresses File:
../../vpn/AgentUtilities/HostConfigMgr.cpp Line: 2245 Invoked Function:
CHostConfigMgr::determinePublicAddrCandidateFromDefRoute Return Code: -28835823 (0xFE480011)
Description: HOSTCONFIGMGR_ERROR_SUPPORTED_PUBLIC_ADDRESS_UNAVAILABLE IPv6
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: getInterfacesInternal File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 1715 missing PPP destination address for
interface "utun0".  Check profile PPPExclusion (set to Automatic?) or contact your
administrator.
Jul 30 01:44:00 maltsev com.avast.proxy[958]: Error connecting to 216.58.204.110:443: connect():
Network is down
Jul 30 01:44:00 --- last message repeated 4 times ---
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: connectTransport File:
../../vpn/Common/IPC/SocketTransport.cpp Line: 1025 Invoked Function: ::connect Return Code: 50
(0x00000032) Description: unknown
```

Source : wikipedia

Accessing the data via log files

- Most often used standard format: **syslog**
  - Date of emission
  - Name of the device
  - Process that triggered emission
  - Priority Level
  - Message contents
  - Message category
  - Seriousness level
- **ELK** is certainly the most used open-source platform for logging (including log visualization)

```
Jul 30 01:44:00 maltsev syslogd[56]: ASL Sender Statistics
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: getInterfacesInternal File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 1715 missing PPP destination address for
interface "utun0".  Check profile PPPExclusion (set to Automatic?) or contact your
administrator.
Jul 30 01:44:00 maltsev acvpnagent[65]: A network interface has gone down.
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: logInterfaces File:
../../vpn/AgentUtilities/Routing/InterfaceRouteMonitorCommon.cpp Line: 477 IP Address Interface
List: FE80:0:0:0:6:97F:A26C:21B1 192.168.1.109 FE80:0:0:0:60D3:1E91:4FC8:29E
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: getInterfacesInternal File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 1715 missing PPP destination address for
interface "utun0".  Check profile PPPExclusion (set to Automatic?) or contact your
administrator.
Jul 30 01:44:00 --- last message repeated 1 time ---
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: GetPrimaryInterfaceIndex File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 501 Unable to get global IPv6 information
from system configuration [error 1004].
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: updatePotentialPublicAddresses File:
../../vpn/AgentUtilities/HostConfigMgr.cpp Line: 2245 Invoked Function:
CHostConfigMgr::determinePublicAddrCandidateFromDefRoute Return Code: -28835823 (0xFE480011)
Description: HOSTCONFIGMGR_ERROR_SUPPORTED_PUBLIC_ADDRESS_UNAVAILABLE IPv6
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: getInterfacesInternal File:
../../vpn/Common/Utility/NetInterface_unix.cpp Line: 1715 missing PPP destination address for
interface "utun0".  Check profile PPPExclusion (set to Automatic?) or contact your
administrator.
Jul 30 01:44:00 maltsev com.avast.proxy[958]: Error connecting to 216.58.204.110:443: connect():
Network is down
Jul 30 01:44:00 --- last message repeated 4 times ---
Jul 30 01:44:00 maltsev acvpnagent[65]: Function: connectTransport File:
../../vpn/Common/IPC/SocketTransport.cpp Line: 1025 Invoked Function: ::connect Return Code: 50
(0x00000032) Description: unknown
```

Source : wikipedia

Definitions

Data scraping, screen scraping, report mining, web scraping

Accessing the data via data scraping

- Data scraping is used when the system to request does not have interface nor API to access the data
- Data scraping is a technique for extracting data from a document published in order to be read by humans
  - Often, web-pages from which we want to gather information
  - But also, any other kind of information formatted to be displayed on a screen / text terminal

Some limits of data scraping

- The operator that publishes these documents might not like data scraping, because it might lead to
  - The system's overload
  - A loss of revenues generated by the ads on the webpage
  - A loss of control over the documents provided
    - Intellectual property issues
- Data scraping is usually reserved to cases where there is no alternative

What is screen scraping?

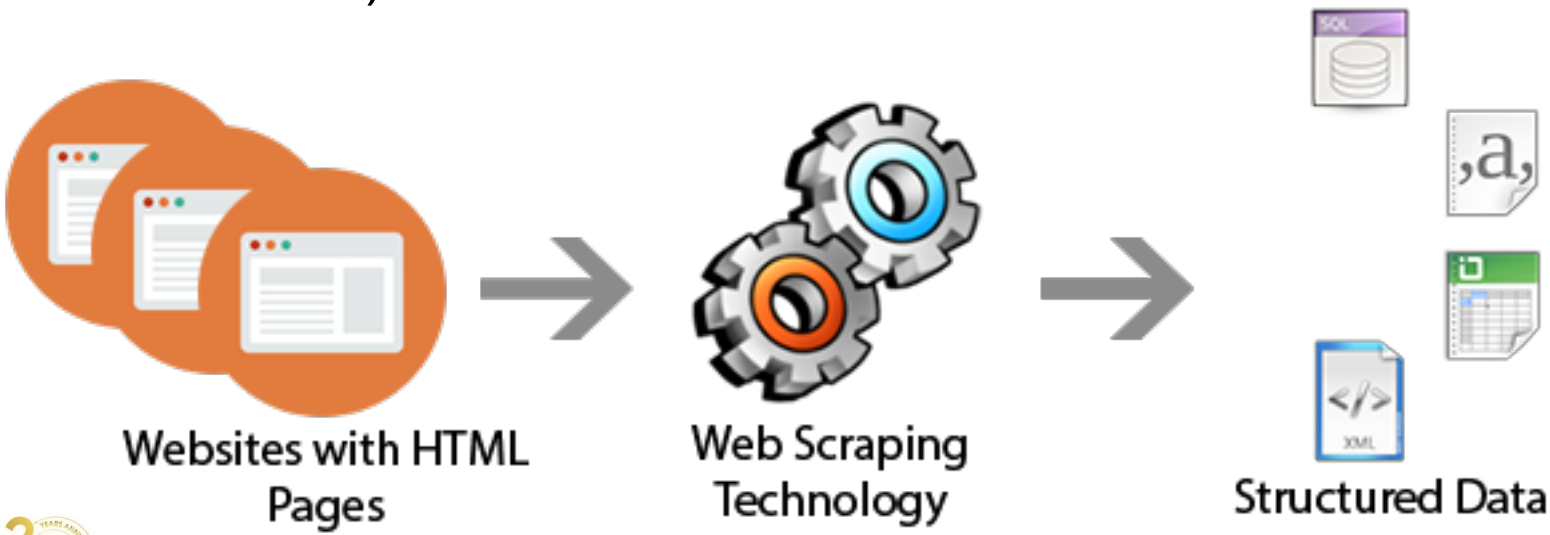
- Consists in extracting the text from the displaying screen of a device
- Usual screen scraping methods use bitmap screenshots and OCR
- In some cases, a program is used to simulate the user's behavior and contol the GUI
  - Hence, a sequence of screens can be automatically captured and fed to a database

What is report mining?

- Consists in extracting the text from the reports written and formatted to be readable by humans (PDF, text, etc.)
  - Simple and quick way to access data without the need for an API
- Examples of softwares: Tabula, import in Tableau

What is web scraping?

- Web pages are text files using a markup-based language (HTML and XHTML) that often contain relevant data
- However, most web pages are conceived for a final (human) user, not for their automatic use
  - That is why web scraping tools were conceived
- In order to defend themselves from web scraper, some sites use defense methods (limiting the number of requests / IP, CAPTCHA…)

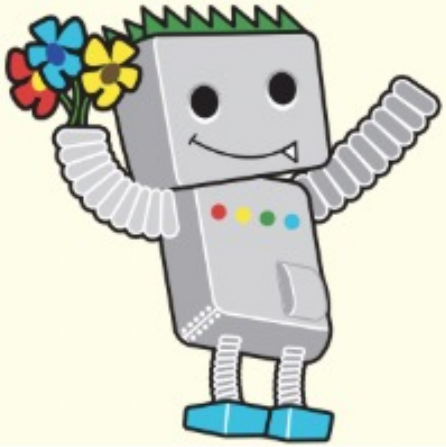


Definitions

Web-crawling *vs.* web-scraping

Web scraping *vs.* web crawling

- Web scraping constists in extracting **relevant** information from a web page in order to re-use this data in another framework and / or under another form
- Web crawling is often used for web page indexing by search engines
  - The term crawling comes from the way a spider would crawl
  - It is used for indexing **all** information on the page
  - A Spider, also known as a robot or a crawler, is actually a program that follows, or ”crawls” links throughout the Internet, grabbing content from sites and indexing them
    - Crawler
    - Spider
    - Robot
    - Web agent



Googlebot
Crawling content on the Internet for Google's index every day, every night, non stop.

Difference between web scraping and web crawling

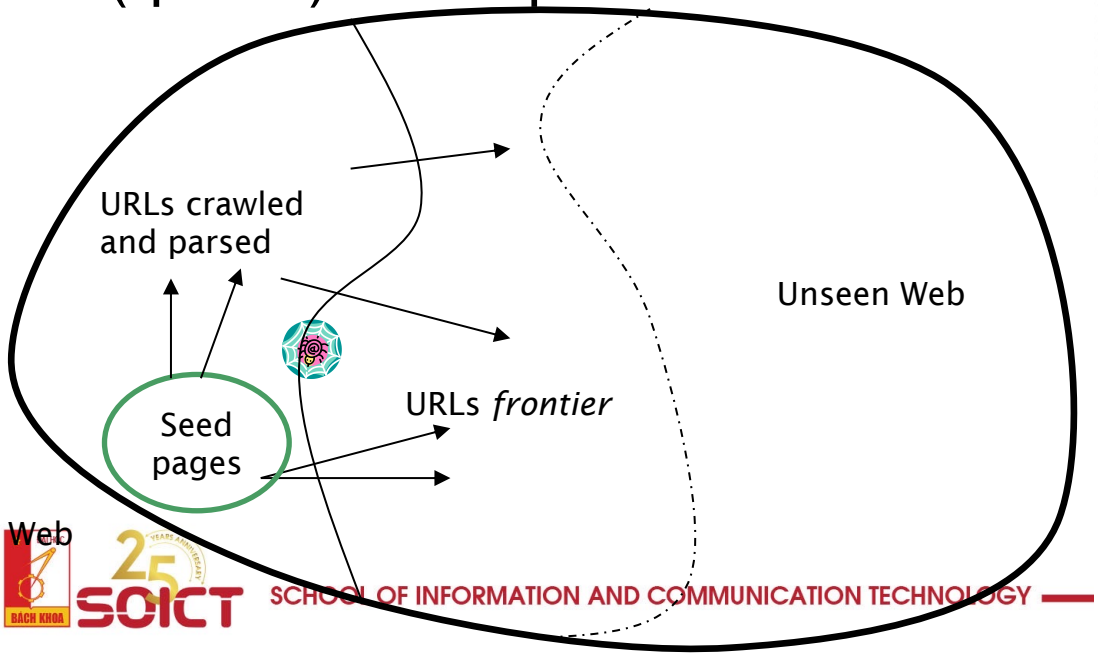
- The difference between web scraping and crawling lies in its **final goal**
  - The final goal of data scraping is data science
  - Hence, it converts un-structured data available in the web into more structured data, that can be analysed
- Example:
  - What Google, Yahoo or Bing does (searching the web for relevant information) is web scraping
  - Their indexing is based on web crawling

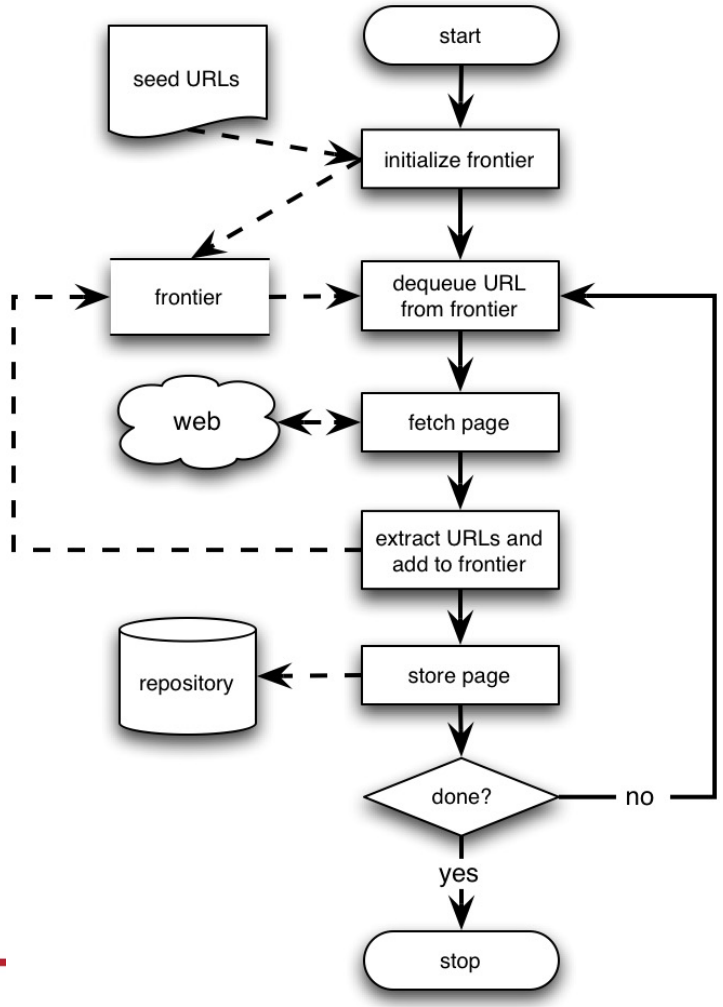
Web-crawling

Some principles

Basic crawling operation

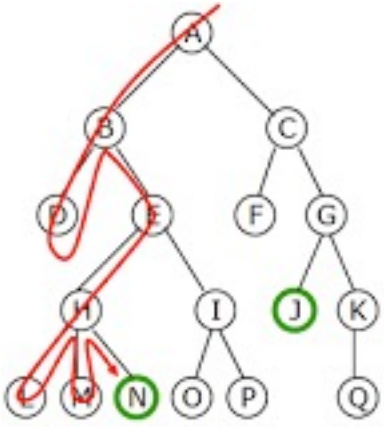
- Begin with known “seed” URLs
- Fetch and parse them
  - Extract URLs they point to
  - Place the extracted URLs on a queue
- Fetch each URL on the frontier (queue) and repeat

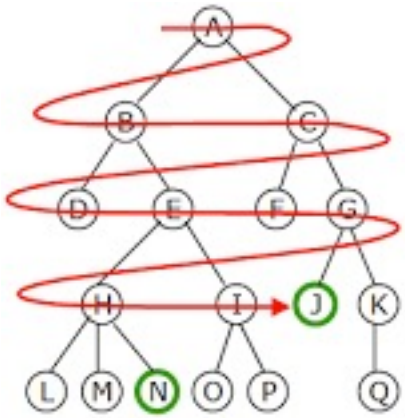




Crawling policy

- Behavior of a Web crawler is the outcome of a combination of policies
  - *Selection policy* which states the pages to download
  - *Re-visit policy* which states when to check for changes to the pages
  - *Politeness policy* that states how to avoid overloading website
  - *Parallelization policy* that states how to coordinate distributed web crawlers





Web crawling challenges

- The Internet is huge
  - Googlebot are distributed
- Filtering interested/non-interested/malicious pages
  - Spam pages
  - Spider traps – pages that are dynamically generated
    - https://en.wikipedia.org/wiki/Spider_trap
- Content freshness
  - Crawlers should be catchup with new, up-to-date contents
- Content deduplication
  - Site mirrors and duplicate pages

Trading-off exploitation vs. exploration

- Exploitation
  - the crawling of pages where the expected value can be predicted with a high confidence
- Exploration
  - the search for new sources of relevant pages

Politeness

- Explicit
  - Specified by webmasters: which parts of the site can be crawled (robots.txt)
- Implicit
  - Avoid hitting any particular site too often, consuming too much web-server resource

Robots.txt

- Protocol for giving spiders “robots” limited access to a website, originally from 1994
  - https://en.wikipedia.org/wiki/Robots_exclusion_standard
- Websites announce their request on what can(not) be crawled
  - For a server, create a file `/robots.txt`
  - This file specifies access restrictions

Robots.txt

Examples [edit]

This example tells all robots that they can visit all files because the wildcard `*` stands for all robots and the `Disallow` directive has no value, meaning no pages are disallowed.

```
User-agent: *
Allow: /
```

The same result can be accomplished with an empty or missing robots.txt file.

This example tells all robots to stay out of a website:

```
User-agent: *
Disallow: /
```

This example tells all robots not to enter three directories:

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /junk/
```

This example tells all robots to stay away from one specific file:

```
User-agent: *
Disallow: /directory/file.html
```

All other files in the specified directory will be processed.

This example tells a specific robot to stay out of a website:

Web-scraping

Some principles

Web scraping

- Web scraping constists in extracting **relevant** information from a web page in order to re-use this data in another framework and / or under another form

Use-cases

- **Private use:** online services; comparing information from different websites (*e.g.* price comparison of flight tickets / fares for the same itinerary but with different companies
- **Academic research:** the web is a huge multi-domain data source that can provide researchers with a huge volume of data (for machine learning for instance, see Chapter 6)
- **Marketing**: A web scraping software can be used to generate leads for marketing. Email and Phone lists can be built by scraping the data from relevant websites. For example, business contact details like phone number and email addresses can be scraped from Google Maps business listings or from institutional webpages.

Extraction modes

- **Semi-automatic extraction**: using a software or an app to suck / clean selected contents from one or several web pages
  - Only the contents that is relevant to the user
  - This is what we will do during exercises / homework
- **Automatic extraction** : using a software or an app to create a corpus of web-pages linked with each other
  - All contents is extracted
  - The software / app emulates a web browser that visits pages and is able to follow all hyperlinks to generate the corpus of all web-pages

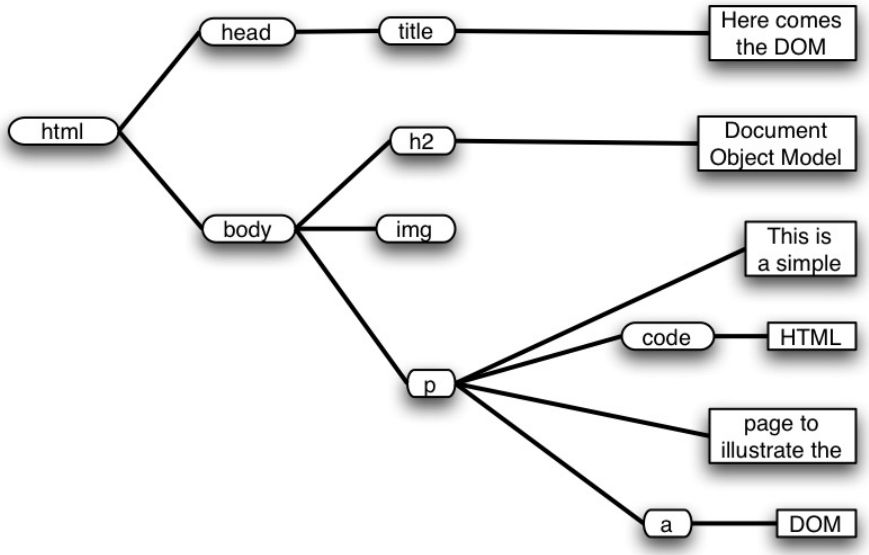
Extraction techniques

- In order to extract information from web pages, the most usual methods use **XPath**
- XPath is a W3C standard to find elements in a XML document
  - W3C is the organization that rules on the web standards
- XPath uses the hierarchical structure of nodes (and attributes) of an XML document, and therefore requires a precise document structure
- HTML documents *partly* respect a hierarchical format with XML tags
  - Parsers need to be flexible

HTML code structure

- HTML has the structure of a Document Object Model (DOM) tree
- DOM tree varies from page to page, even for the same catalog
  - Due to dynamic contents, dynamic ads, etc.

```
<html>
  <head>
    <title>Here comes the DOM</title>
  </head>
  <body>
    <h2>Document Object Model</h2>
    <img align="right" alt="dom pict" src="dom.png">
    <p>
      This is a simple
      <code>HTML</code>
      page to illustrate the
      <a href="http://www.w3.org/DOM/">DOM</a>
    </p>
  </body>
</html>
```



Xpath: example

```xml
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

<book>
  <title lang="en">Harry Potter</title>
  <price>29.99</price>
</book>

<book>
  <title lang="en">Learning XML</title>
  <price>39.95</price>
</book>

</bookstore>
```

Path Expression	Result
bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore **Note:** If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

Practice it yourself on https://www.w3schools.com/xml/xpath_examples.asp

Xpath: example

```xml
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

<book>
  <title lang="en">Harry Potter</title>
  <price>29.99</price>
</book>

<book>
  <title lang="en">Learning XML</title>
  <price>39.95</price>
</book>

</bookstore>
```

Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. **Note:** In IE 5,6,7,8,9 first node is[0], but according to W3C, it is [1]. To solve this problem in IE, set the SelectionLanguage to XPath: *In JavaScript: xml*.setProperty("SelectionLanguage","XPath");
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='en']	Selects all the title elements that have a "lang" attribute with a value of "en"
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

Technical limitations

- **Inconsistent / cahotic organization of the information**
  - Relevant information might be formatted in a different way within different websites, or even within one single website
- **Evolutions in the information structure**
  - Some data scraping programs are meant to be run on a routine, over time, to suck all freshly added info
  - As the website structure might evolve with time, the criteria that were settled initially to select relevant information might become obsolete
  - This problem arises most often for websites that use Content Management Systems *(e.g.* Wordpress) that allow to change for the graphical theme
    - As changes in the graphical theme often imply changes in the document structure

Technical limitations

- **Access restrictions**
  - Some website contents are exclusively available for authentified users
  - There are tools that can emulate authentication procedure, one has to possess a registered login/password, and provide it to the extraction tool
- **Dynamically generated contents**
  - Some pages don't load all content from the first request
  - Instead, they dynamically load the contents according to the browser's actions
    - These actions triggering queries on the website's database, *e.g.* mySQL
  - Even though some tools can emulate this type of interactions, most often, the analysis is made on the HTML contents generated from the first request

Technical limitations

- As stated before, the website publisher might not like data scraping, because it might lead to
  - The system's overload
  - A loss of revenues generated by the ads on the webpage
  - A loss of control over the documents provided
    - Intellectual property issues
- Website administrators might use some of the following tools to prevent « non-human » traffic
  - **Time limitations for each request**
    - Limitation in the number of hits / second for instance
    - A hit is the fact of going from one page to another, from the same domain
    - Reason for using this type of limitations is that humans usually take much longer to go from one page to another than a data scraper

Technical limitations

- Website administrators might use some of the following tools to prevent « non-human » traffic
  - **Time limitations for each request**
  - **IP address-based Denial of Service (DoS)**
    - Automatically triggered when a website receives too many requests from one IP address
    - DoS is often used in case of cyber-attacks
  - **Bandwidth limitations**
  - **Verification that the user is human**
    - *e.g.* CAPTCHA, etc.

Ethical / legal aspects

- Legal aspects regarding web-content reproductions are
  - Unclear in most countries
    - Note the European regulation « General Data Protection Regulation » (GDPR)
  - Differ largely from one country to another
- Is it OK to cite contents from another site, even if one provides the reference?
  - Tough question!
  - On one hand, intellectual property is respected because credit is given to the original author
  - On the other hand, the original author looses traffic on his/her own website, which might cause a loss of revenue (ads, etc.)

Ethical / legal aspects

- A good practice is to inform the website's administrator that you have the intention to use their website's contents
  - No formal request is legally required, but the advantage of asking is twofold
    - You might obtain an agreement, thus avoiding some of the previously cited technical limitations
    - The administrator might give you info or means to access the webpage contents, *e.g. via* APIs
  - In exchange, the administrator might ask you to add his/her URL(s) on your own website (if any) to enhance their ranking by search engines
    - **Win-win** situation!

Practice

Web-scraping with *web-scraper* and Scrapy

Web scraping tools

- There are numerous web scraping tools, with different interface types:
  - Programming
    - Software libraries: **Scrapy**, Beautifulsoup (Python), PhantomJS
  - Cloud: ScrapingHub, Dexi.io...
  - Stand-alone: ParseHub, OctoParse...
  - Web-browser plugins:
    - Data Scraper - Easy Web Scraping, Instant Data Scraper, **Web Scraper**

General Features Comparison					
	Octoparse	**Parsehub**	**Mozenda**	**Dexi.io**	**Import.io**
Usability	★★★★★	★★★★☆	★★★★★	★★★★★	★★★★☆
Functionality	★★★★☆	★★★★☆	★★★★☆	★★★★★	★★★☆☆
Easy to learn	★★★★★	★★★★☆	★★★★★	★★★☆☆	★★★★★
Customer support	Email, phone, community	Email, live chat, forum	Phone, email, video chat	Email, phone, community	Email, chat bot, community
Price	$0 - $249	$149 - $499	$100/5000 page credits	$119 - $699	$299 - $9999
Trial/Free version	Free Version	Free Version	30 days trial	Trial	7 days trial
OS (Specifications)	Win	Win, Mac, Linux	Win	Win, Mac, Linux	Win, Mac, Linux
Data Export Formats	TXT, CSV, XLS, Databases	CSV, JSON	CSV, TSV, XML, XLS, JSON	CSV, XLS, XML, JSON, Zip	CSV, JSON, Google sheets
Multi-thread	✔	✔	✔	✔	✗
API	✔	✔	✔	✔	✔
Scheduling	✔	✔	✔	✔	✔

Why using Web-Scraper and Scrapy?

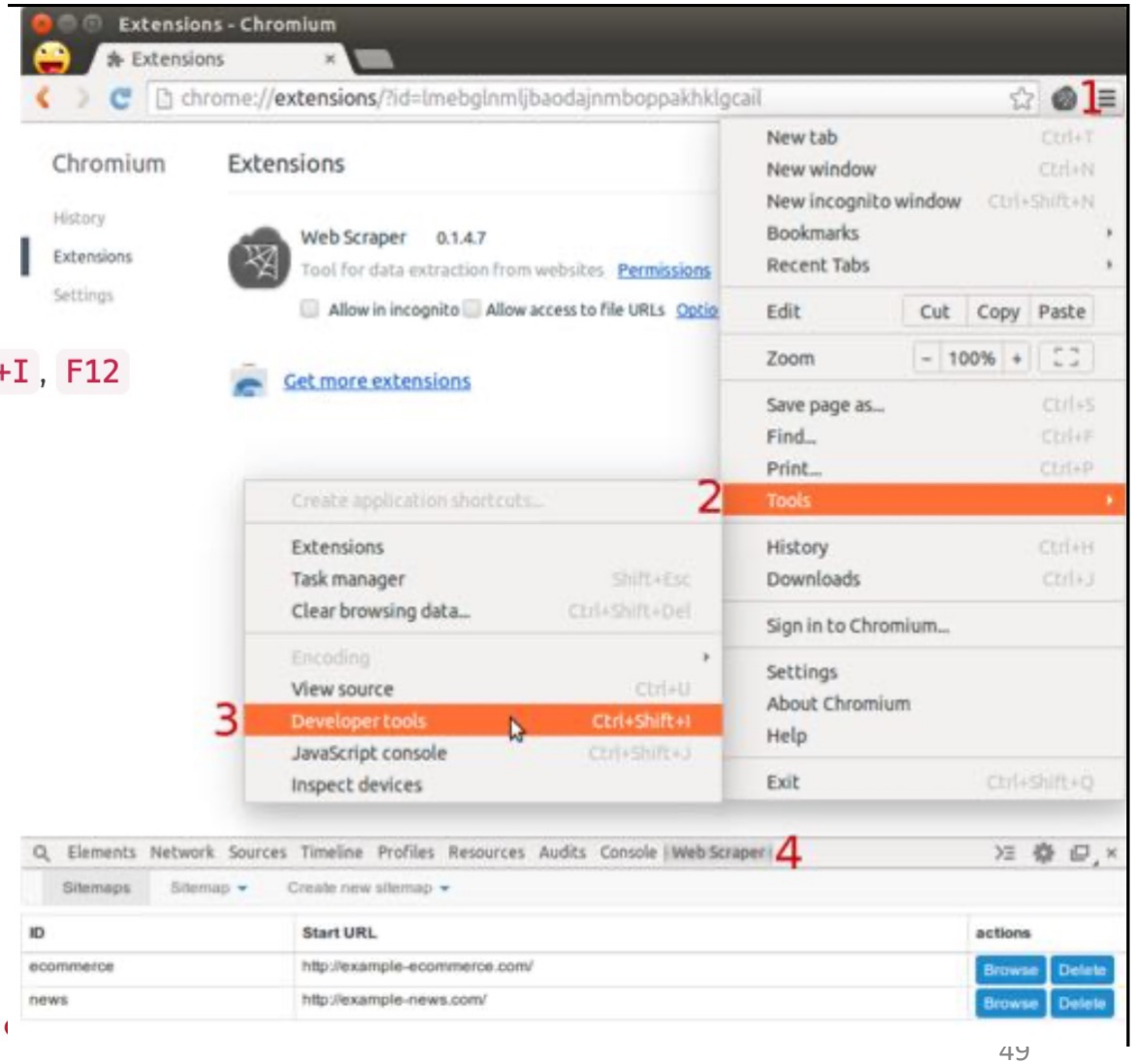
- Web-Scraper
  - Simple to install and to use
  - Easy to understand
  - Free for non-cloud usage
- Scrapy
  - Example of an effective Python library
  - Open-source web scraping framework
  - More refined than Web-Scraper

Practice

Web-scraper

Opening web-scraper

- Web Scraper is integrated into Chrome Developer tools
- You can install the extension from Chrome store or Firefox browser Add-ons.
- Keyboard shortcuts to open Developer tools:
  - Windows, Linux: `Ctrl+Shift+I`, `F12`
  - Mac `Cmd+Opt+I`
- After opening Developer tools, open *Web Scraper* tab

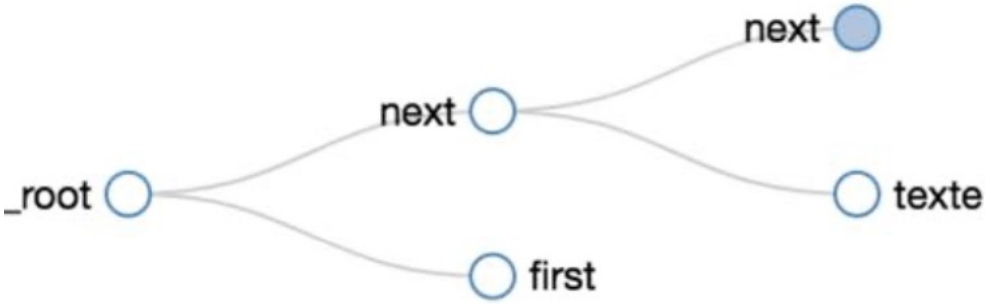


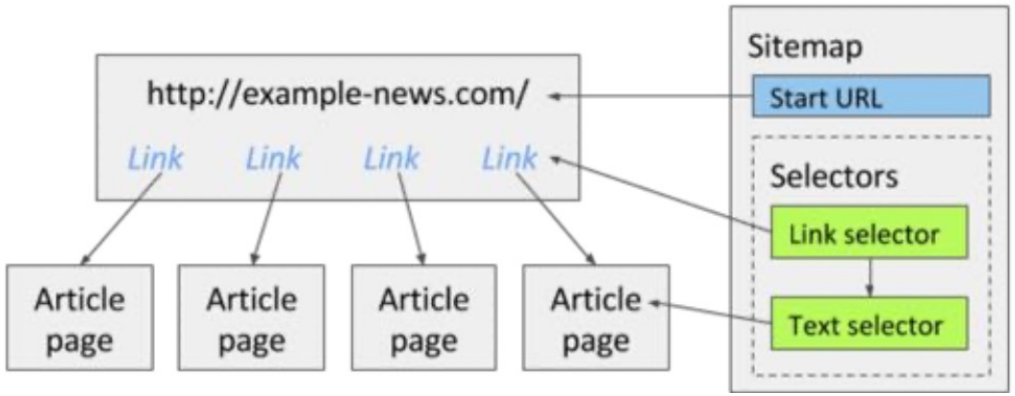
Web-scraper: sitemap

- To start with, the user must specify the URL from which information should be extracted
- The URL can be parameterized so as to scrap multiple pages, *e.g.* by specifying the page numbers
  - Example
    - **http://example.com/page/[0-20:10]** corresponds the the URLs:
      - http://example.com/page/0
      - http://example.com/page/10
      - http://example.com/page/20

Web-scraper: selectors hierarchy

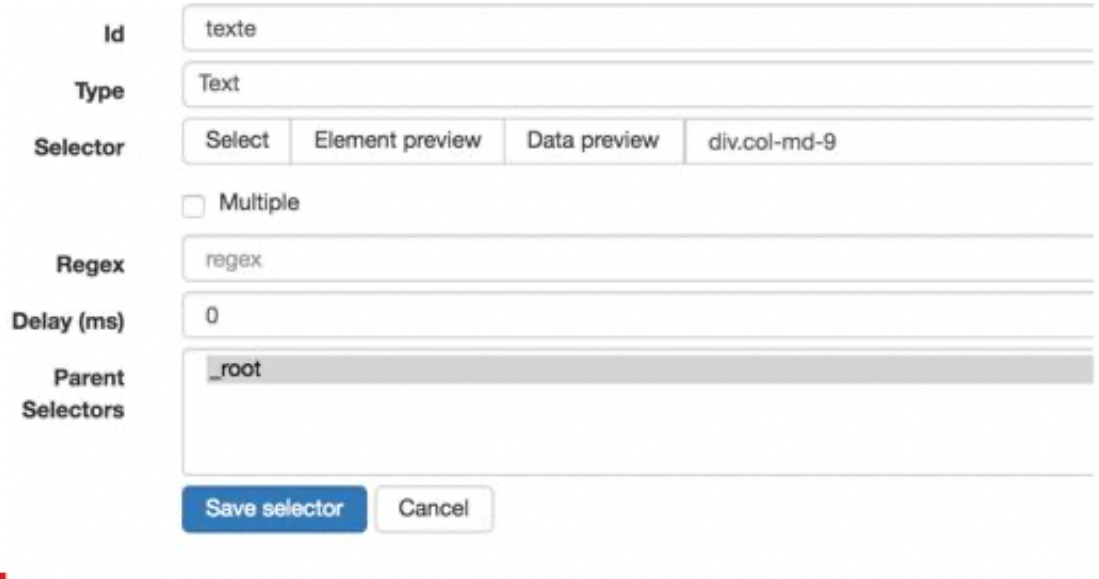
- Web-Scraper uses a hiararchy of **selectors**
  - Data extraction selectors
  - Link selectors for site navigation
  - Element selectors
- A selector references an element in the XPath hierarchy
- From any element, one can select its descendants
- Information lie in the tree leaves
  - Text
  - Image
  - Table





Web-scraper: text selector

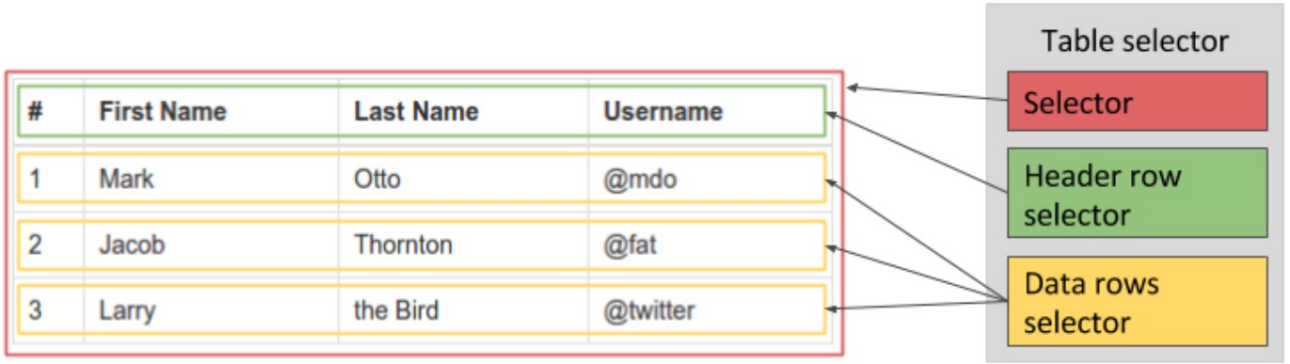
- Text selector is the most simple selector
  - Used for text selection
- Extract text from the selected element **and** from all its child elements.
- HTML will be stripped and only text will be returned (ignoring <script> and <style> tags). New line
 tags will be replaced with newline characters.
- You can additionally apply a regular expression to resulting data.



- Element Preview
  - Highlighting the selected text
- Data Preview
  - Result data
- The right part contains the Xpath selector

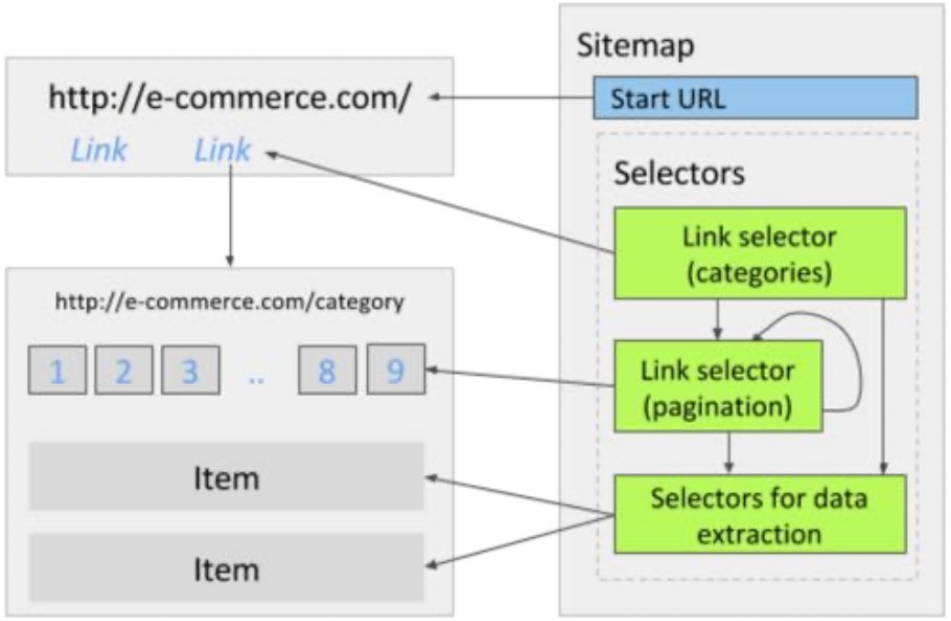
Web-scraper: table selector

- Composite selector
- Check « Multiple » to select all rows



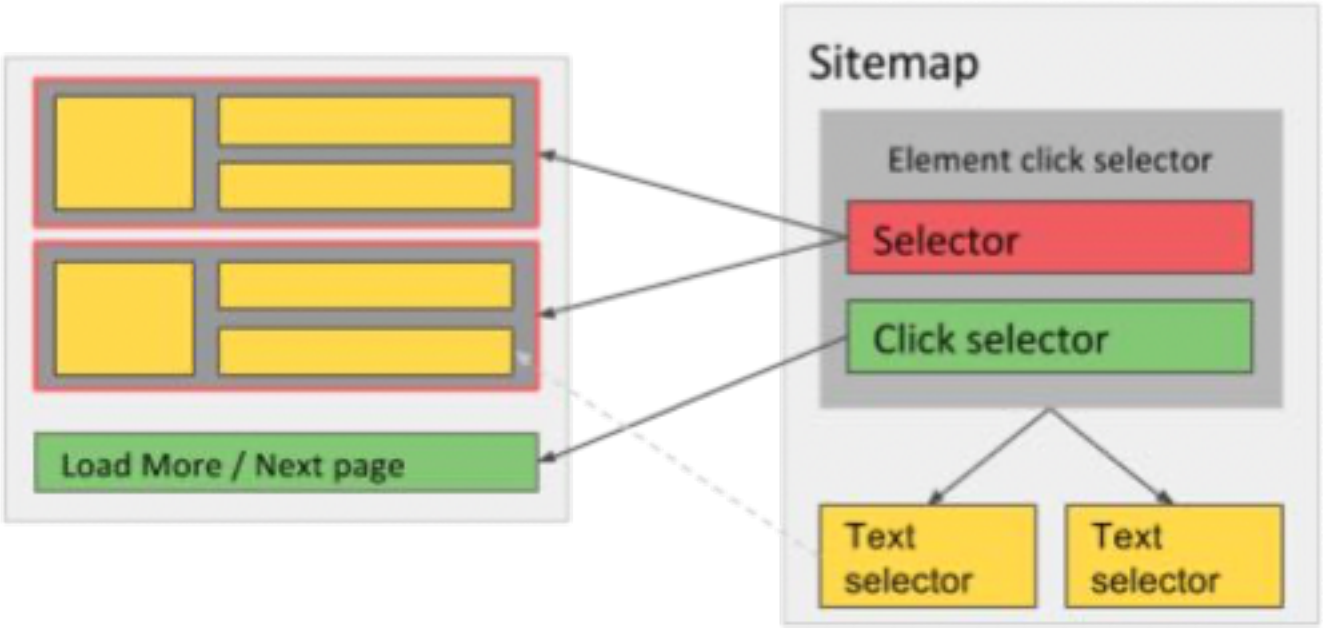
Web-scraper: link selector

- Link selector is used for link selection and website navigation.
  - If you use *Link selector* without any child selectors then it will extract the link and the href attribute of the link
  - If you add child selectors to *Link selector* then these child selectors will be used in the page that this link was leading to
  - Link selectors can be recursive



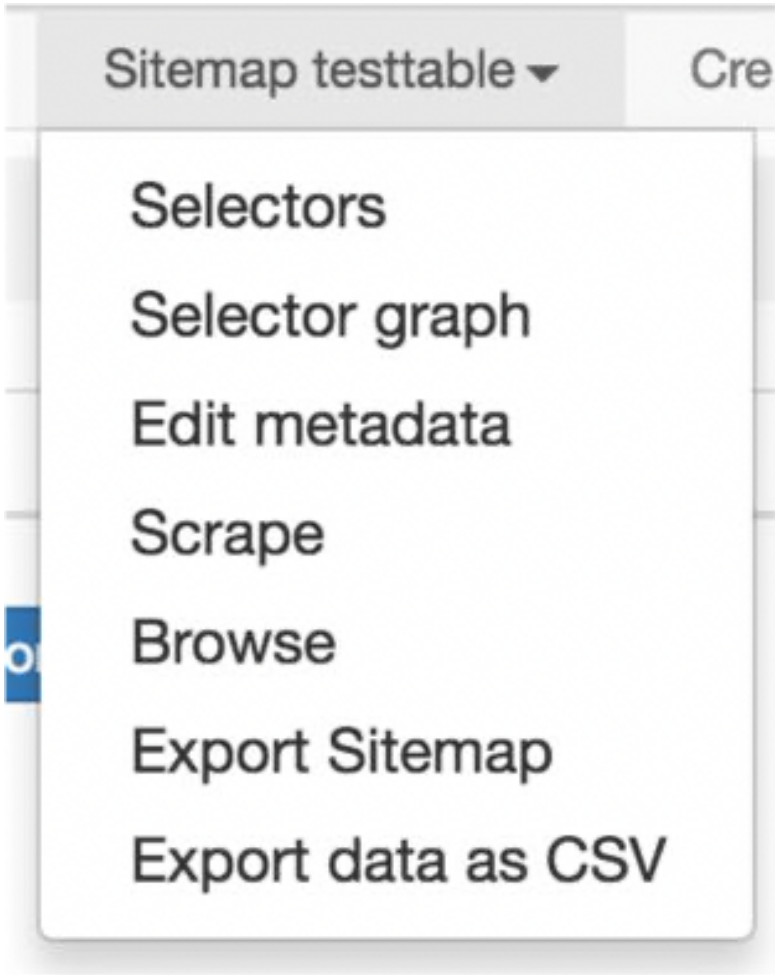
Web-scraper: click selector

- Click selectors are used when it is necessary to emulate a click



Web-scraper: data export

- To be able to export data, one needs to
  - Scrape the page(s)
    - A popup opens up on the selected URL
    - Selectors are triggered according to the selector hierarchy
    - Data is fetched and exported
- Web Scraper browser extension supports data export in CSV format
- Web Scraper Cloud supports data export in CSV, XLSX and JSON formats



Web-scraper: demo

- https://youtu.be/n7fob_XVsbY

References

- https://www.webscraper.io/tutorials
- https://www.webscraper.io/documentation

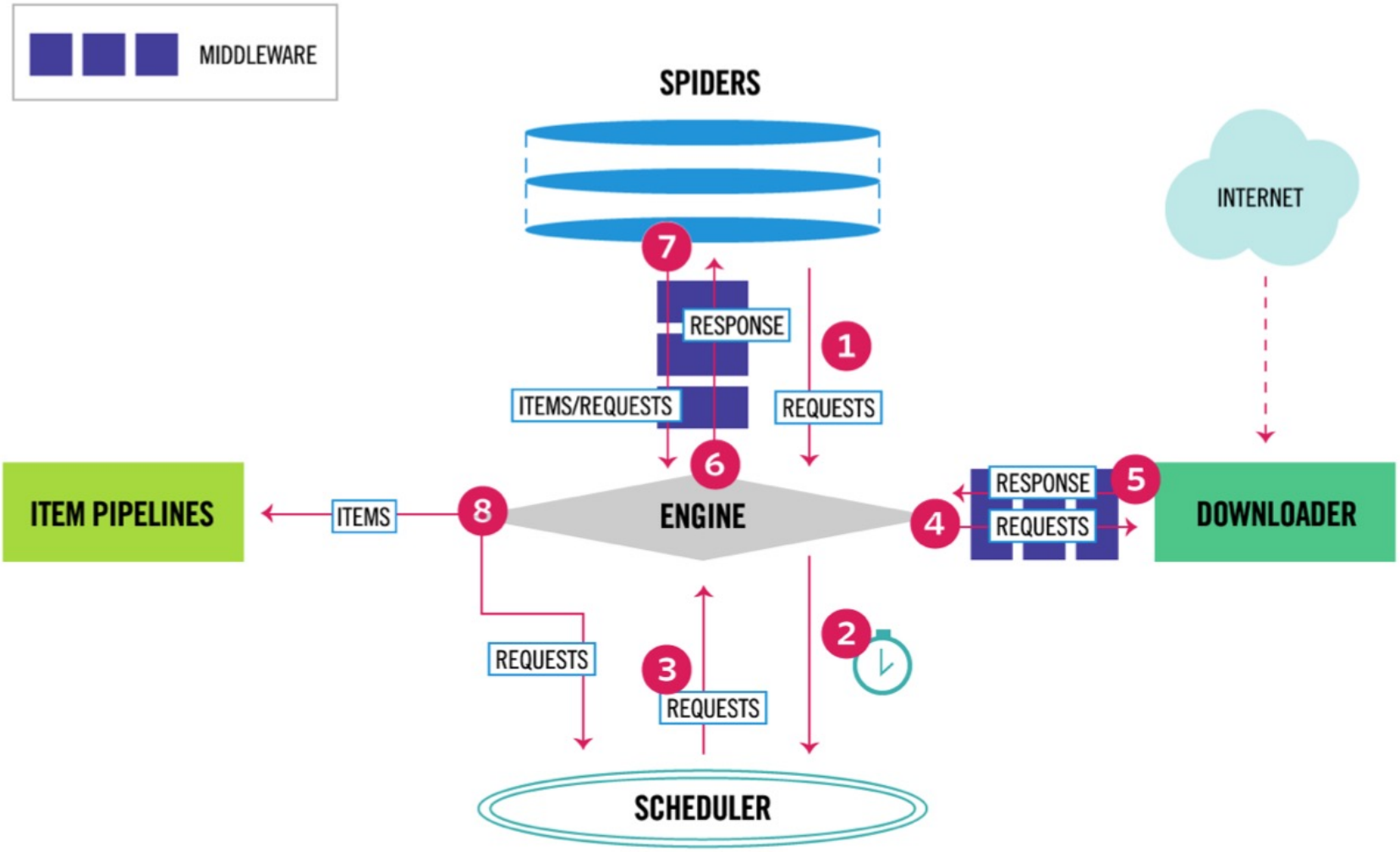
Practice

Scrapy

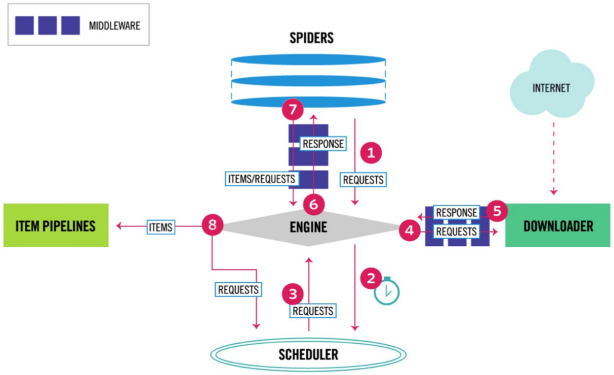
Intro

- Whereas Web Scraper is a web-browser extension enabling semi-automatic extraction…
- … Scrapy is a more complete open-source Python library
  - Scrapy can handle by itself « annoying stuff », such as
    - Throttling
    - Concurrency
    - XML sitemaps
    - Filtering duplicated URLs
    - Retry on Error

Scrapy components



Scrapy components



- Scrapy Engine
  - controlling the data flow between all components
- Scheduler
  - receives requests from the engine and enqueues them for feeding them later
- Downloader
  - Fetching web pages according to requests and feeding the responses to the engine
- Spiders
  - parse responses and extract items or additional requests to follow
- Item pipeline
  - processing the items once they have been extracted by the spiders
- Downloader middlewares
  - Process requests when they pass from the Engine to the Downloader and vice-versa
- Spider middlewares
  - specific hooks that sit between the Engine and the Spiders
  - process spider input (responses) and output (items and requests)

Exercises / homework

Exercises / homework

1. **XPath**
   - Learn it by yourself on https://www.w3schools.com/xml/xpath_examples.asp
2. **WebScraper** tutorials
   - https://www.webscraper.io/tutorials
3. **Scrapy**
   a) https://doc.scrapy.org/en/latest/intro/tutorial.html
      - Define your own data structures
      - Write spider
      - Leverage Built-in Xpath and CSS selectors to extract desired data
      - Built-in Json, csv, xml output
      - Interactive shell console

   b) https://doc.scrapy.org/en/latest/topics/media-pipeline.html
      - Downloading and processing files and images

   c) « Web Scraping in Python using Scrapy _ Codementor »: PDF in the Google Teams

Questions

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!!!