LECTURE 3 MINING WEB CONTENT I

LEK HSIANG HUI

OUTLINE

Introduction to Web Content Mining

Web Basics

Techniques for performing Web Content Mining

Extracting Content from HTML Source

INTRODUCTION TO WEB MINING

Introduction to Web Content Mining

Web Basics

Techniques for performing Web Content Mining

Extracting Content from HTML Source

WEB MINING

Web mining is concerned with mining data from the web which can then be transformed into knowledge

2 main aspects we will be focus on

- Mining Web Content
- Data mining techniques which are more specific to web content (e.g. recommender systems)

MINING WEB CONTENT

Data is usually not available for us to download as a spreadsheet

We might also want to aggregate multiple data sources together

- E.g. Suppose we have the location information (address) from food review sites (yelp/dianping)
- Might want to augment this information with Google Maps/Baidu Map (which contains other info like amenities) to create a dataset suitable for doing predictive modeling
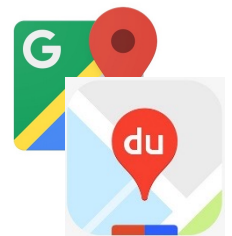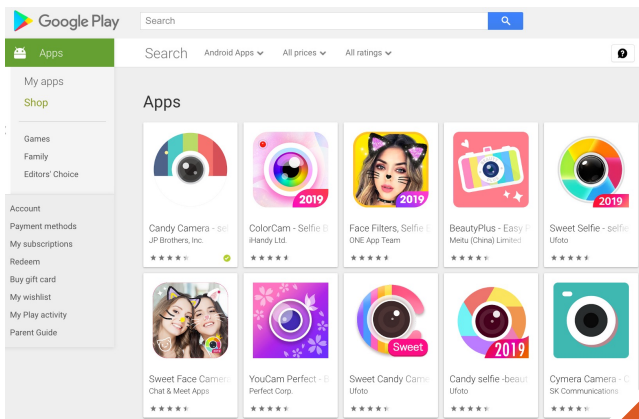
MINING WEB CONTENT

Knowledge on how to acquire data in a programmatic manner is very important

- Not realistic to manually copy and paste all the time
- Not always easy to acquire data off the web
- … but there are some techniques to do this

EXAMPLE

Mining Google Play store Apps Statistics

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	7-Jan-18	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	15-Jan-18	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	1-Aug-18	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	8-Jun-18	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	20-Jun-18	1.1	4.4 and up

APPLICATIONS

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	7-Jan-18	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	15-Jan-18	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	1-Aug-18	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	8-Jun-18	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	20-Jun-18	1.1	4.4 and up

Further manipulate the data to gain insights of the mobile app market

	Reviews	
Type	Free	Paid
Category		
ART_AND_DESIGN	1712274.0	2166.0
AUTO_AND_VEHICLES	1159503.0	4163.0
BEAUTY	396240.0	NaN
BOOKS_AND_REFERENCE	21957273.0	1796.0
BUSINESS	13907683.0	46869.0
COMICS	3383276.0	NaN
COMMUNICATION	815378046.0	84214.0
DATING	7288124.0	3154.0
EDUCATION	39561141.0	34645.0
ENTERTAINMENT	59168145.0	10009.0
...	...	...
PERSONALIZATION	88896483.0	449657.0
PHOTOGRAPHY	213285996.0	230654.0
PRODUCTIVITY	113945254.0	171721.0
SHOPPING	115040738.0	484.0
SOCIAL	621241180.0	242.0
SPORTS	70679534.0	150635.0
TOOLS	273013107.0	171937.0

Number of reviews between Free and Paid Apps for different category

	Rating	Reviews
Category		
ART_AND_DESIGN	4.358065	2.637600e+04
AUTO_AND_VEHICLES	4.190411	1.369019e+04
BEAUTY	4.278571	7.476226e+03
BOOKS_AND_REFERENCE	4.346067	9.506090e+04
BUSINESS	4.121452	3.033598e+04
COMICS	4.155172	5.638793e+04
COMMUNICATION	4.158537	2.107138e+06
DATING	3.970769	3.115931e+04
EDUCATION	4.389032	2.538191e+05
ENTERTAINMENT	4.126174	3.971688e+05
...	...	...
PERSONALIZATION	4.335987	2.279238e+05
PHOTOGRAPHY	4.192114	6.373631e+05
PRODUCTIVITY	4.211396	2.691438e+05
SHOPPING	4.259664	4.424662e+05
SOCIAL	4.255598	2.105903e+06
SPORTS	4.223511	1.844536e+05
TOOLS	4.047411	3.240629e+05

Average rating and number of reviews for different category

WEB CRAWLER

A **web crawler** (or **web spider**) is a computer program that systematically access the World Wide Web (WWW) to download web data

They are typically used for the purpose of **web indexing** (i.e. to power a search engine)

- Bingbot
- Googlebot
- etc

WEB SCRAPER

A web scraper is also a computer program that can be used for downloading web data

- However, they are more targeted at extracting specific data

- E.g. Web scraper for:
Google Play Store reviews, Amazon Products, etc

WEB CRAWLER VS WEB SCRAPER

Web crawlers tend to extract the entire content of a page as a whole (e.g. entire page's html)

- Lesser work needed as compared to web scrapers
- However, the data is quite raw and requires more processing (for the purpose of analytics)

Web scrapers are more specialized

- They are specially designed to extract only certain parts of the page
- Usually multiple parts in a single page

WEB CRAWLER EXAMPLE

Input:
https://sws.comp.nus.edu.sg/

Output:
Text and HTML of each page
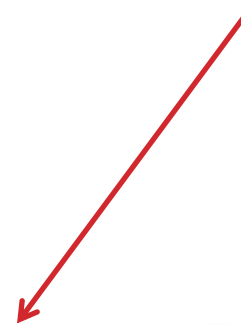
	url	text	html
0	https://sws.comp.nus.edu.sg/	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...
1	https://sws.comp.nus.edu.sg/contact.htm	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...
2	https://sws.comp.nus.edu.sg/Topics%20and%20Clu...	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...
3	https://sws.comp.nus.edu.sg/index.htm	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...
4	https://sws.comp.nus.edu.sg/Past%20Projects.htm	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...
5	https://sws.comp.nus.edu.sg/Media%20Room.htm	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...
6	https://sws.comp.nus.edu.sg/Structure.htm	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...
7	https://sws.comp.nus.edu.sg/Application.htm	CLUSTERS\n\n\n\nTOPICS\n\n\n\nSTRUCTURE\n\...	b'<!DOCTYPE HTML>\r\n<html xmlns="http://www.w...

WEB SCRAPER EXAMPLE

Input:
IMDB Scraper
(created specially for scraping IMDB content)

Output:
Each column is one piece of useful information

Scraper has to be crafted to extract these information

	title	rating
0	Avengers: Endgame(2019)	8.4
1	Captain Marvel(2019)	6.9
2	Avengers: Infinity War(2018)	8.4
3	Guardians of the Galaxy(2014)	8.0
4	The Avengers(2012)	8.0
5	Avengers: Age of Ultron(2015)	7.3

13

CHALLENGES IN WEB CONTENT MINING

Handling dynamic content (e.g. ads) which might affect extraction accuracy

14

CHALLENGES IN WEB CONTENT MINING

Jan 2017
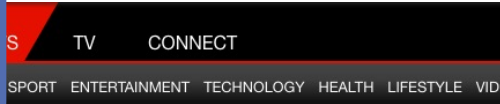
Jan 2019
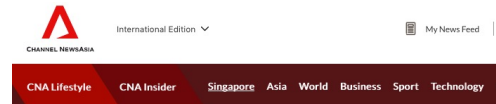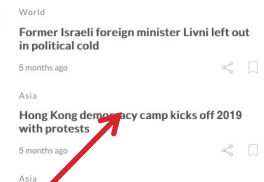
Each time when the site layout changes, potentially, the web scrapers might break

CHALLENGES IN WEB CONTENT MINING

Data is loaded dynamically on the fly

Not just when you first access the page
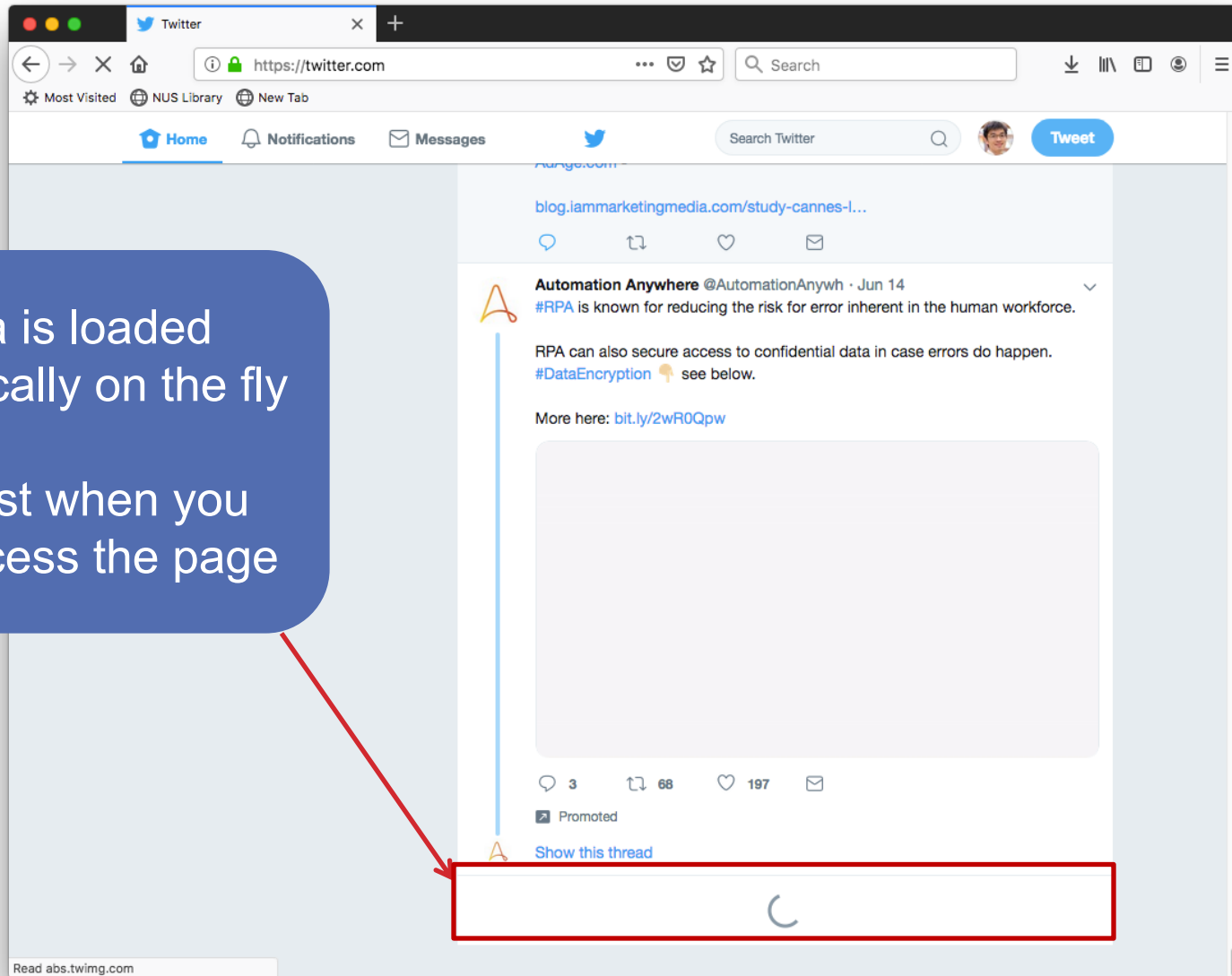
WEB BASICS

| Introduction to Web Content Mining | Web Basics | Techniques for performing Web Content Mining | Extracting Content from HTML Source |

HTML

The web is powered by HTML

- HTML = HyperText Markup Language
- Scripting language used to create webpages
- Defines the format, layout, resources in a webpage
- Used together with:
  - JavaScript: Additional features/logic
  - Cascading Style Sheets (CSS): Styling

HTML

```html
<html>

 <head>

  <title>SWS3023</title>

 </head>

 <body>

  <h1>Welcome to my webpage</h1>

 </body>

</html>
```

Browser window showing:

SWS3023 — file:///C:/Users/dislhh/Dropbc

Welcome to my webpage
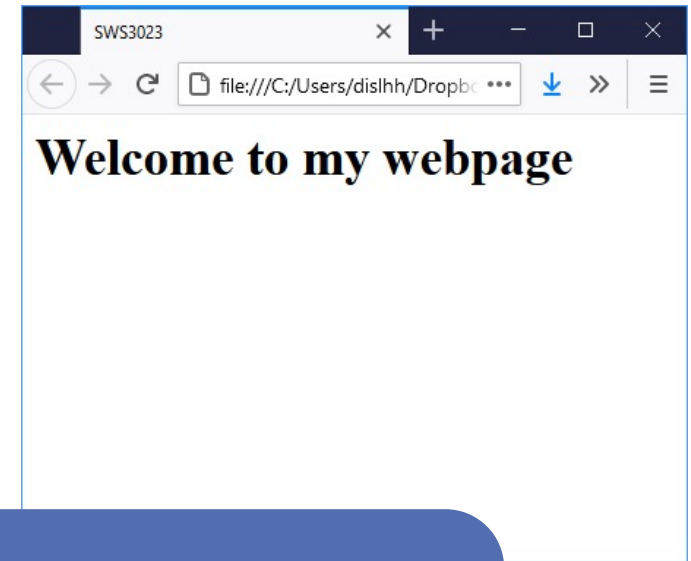
HTML is just a text document that defines both the content and the layout of the page

19

HOW TO GET THE HTML SOURCE OF A PAGE?
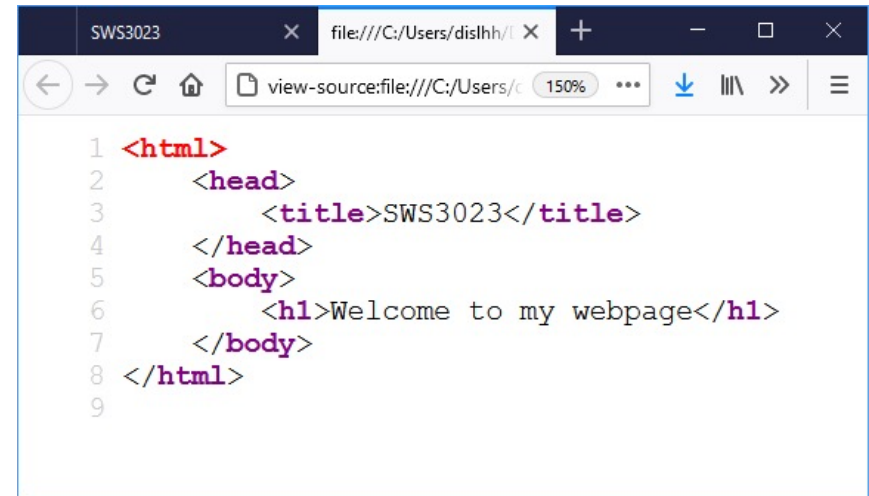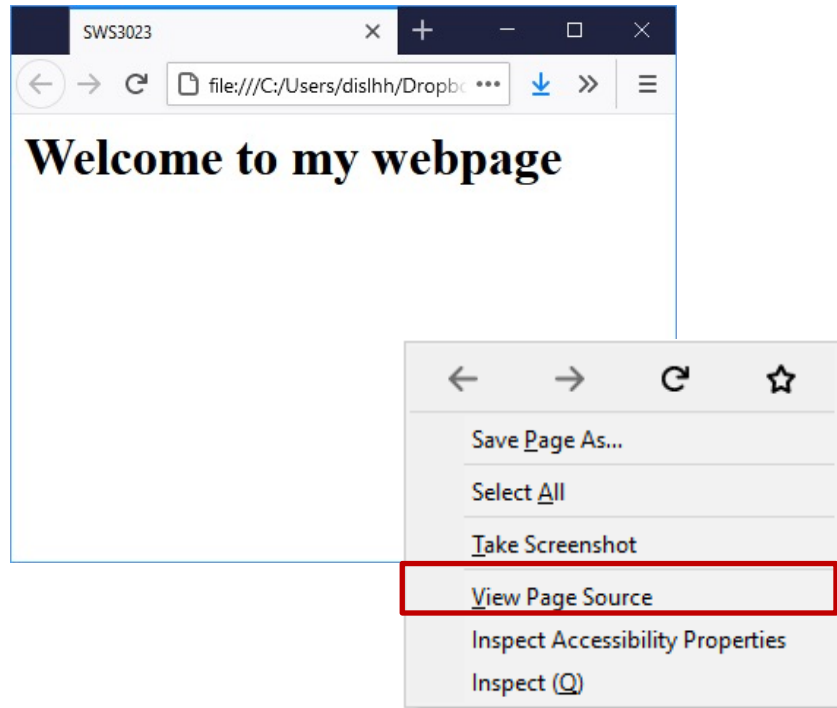
Right click >>
View Page Source

HTML

```html
<html>

  <head>

    <title>SWS3023</title>

  </head>

  <body>

    <h1>Welcome to my webpage</h1>

  </body>

</html>
```
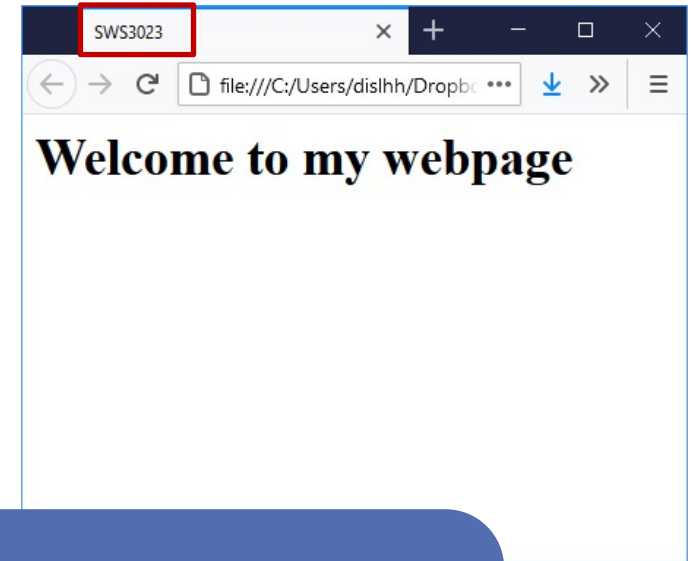
`<title>…</title>`
defines the title of the page

HTML

```
<html>

 <head>

  <title>SWS3023</title>

 </head>

 <body>

  <h1>Welcome to my webpage</h1>

 </body>

</html>
```

SWS3023

file:///C:/Users/dislhh/Dropbc

Welcome to my webpage
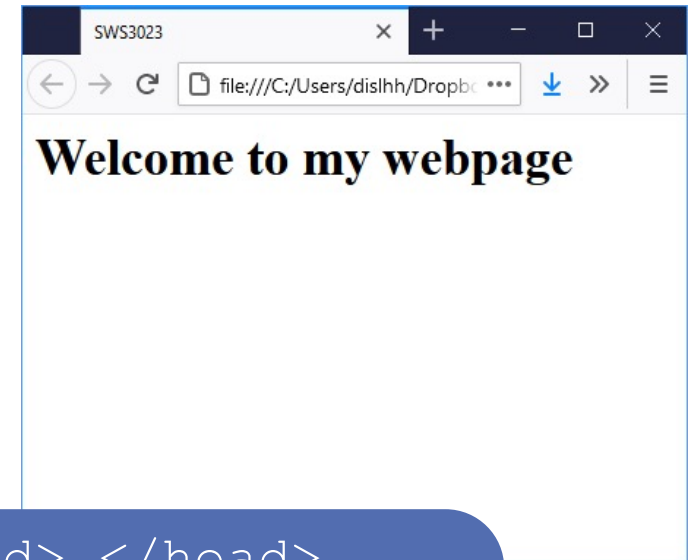
<head>…</head>
section contains information
about the page that is usually
not visible (with the exception
of <title>…</title>

HTML

```
<html>

  <head>

    <title>SWS3023</title>

  </head>

  <body>

    <h1>Welcome to my webpage</h1>

  </body>

</html>
```
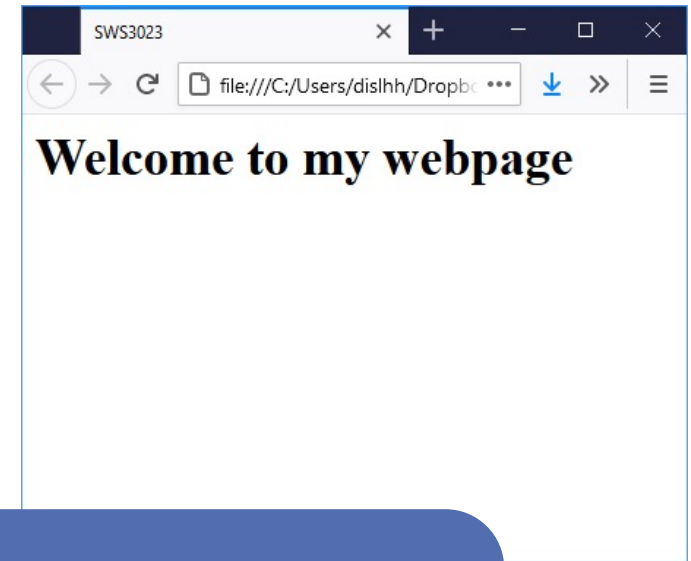
<body>…</body>
section contains presentational content for the page

HTML

```html
<html>

  <head>

    <title>SWS3023</title>

  </head>

  <body>

    <h1>Welcome to my webpage</h1>

  </body>

</html>
```

SWS3023

file:///C:/Users/dislhh/Dropbo ...

Welcome to my webpage
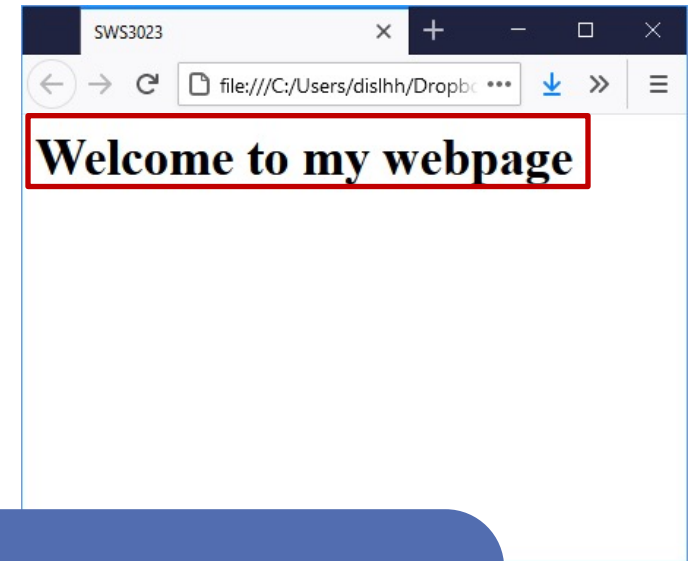
<h1>...</h1>
allows you to add a heading
text (similar to Microsoft Word)

24

HTML

```html
<html>

  <head>

    <title>SWS3023</title>

  </head>

  <body>

    <h1>Welcome to my webpage</h1>

  </body>

</html>
```
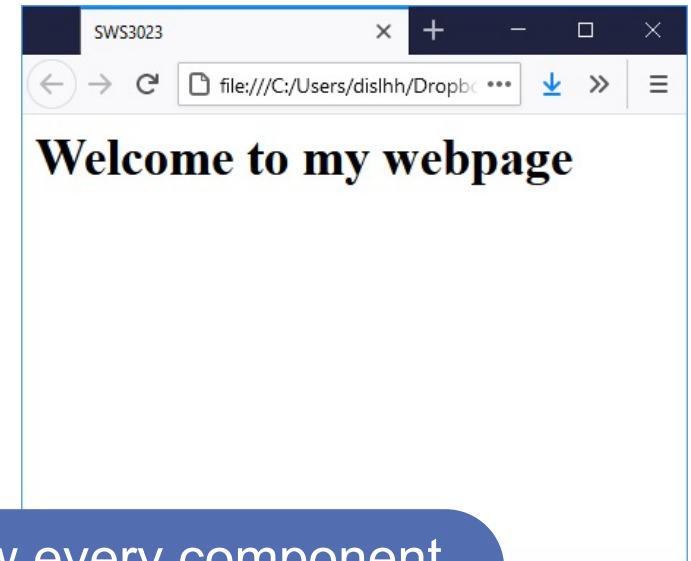
Notice how every component of a webpage is defined with a starting tag (e.g. `<title>`) and an ending tag (e.g. `</title>`)

25

HTML

Welcome to my webpage

```html
<html>

 <head>

  <title>SWS3023</title>

 </head>

 <body>

  <h1>Welcome to my webpage</h1>

  <img src="hello.jpg" width="250"/>
```

Some elements have a single self-closing tags instead (notice the `/>` at the end)

`src` and `width` are attributes of `<img>` tag
Can use `"` or `'` to enclose attribute values (as long as they are matching quotes)

26

BODY SECTION

Many other types of tags can be added to the body section:

- Paragraph (`<p>...</p>`)
- Heading (`<h1>...</h1>, h1,h2,...h6`)
- Formatting tags
    - Bold (`<b>...</b>`)
    - Italics (`<i>...</i>`)
    - Underline (`<u>...</u>`)
- Links
- Images
- List
- Table
- Div, Span

CHARACTERISTICS OF HTML

HTML is case insensitive:

- `<HTML>` is the same as `<html>`

Web browsers do not recognize line breaks (or multiple empty spaces)

- i.e.
  ```
  I want to have
  another line
  ```
  will be seen on the browser as a single line:
  ```
  I want to have another line
  ```

Line break tags (`<br>`) must be used to denote line breaks

TECHNIQUES FOR PERFORMING WEB CONTENT MINING

Introduction to Web Content Mining

Web Basics

Techniques for performing Web Content Mining

Extracting Content from HTML Source

HOW DOES A WEB CRAWLER WORK?
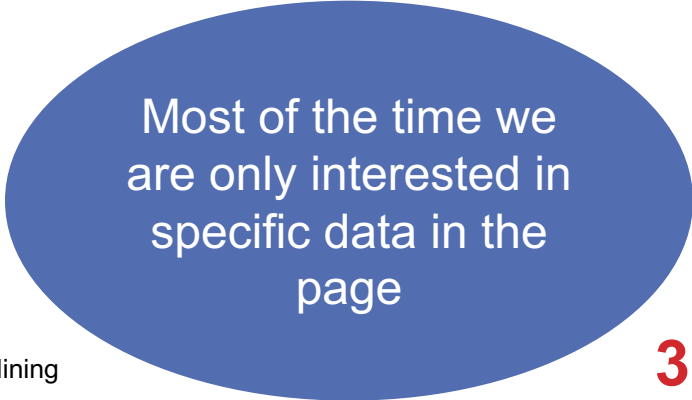
General Approach:

- Maintain a queue (or stack)
- The queue is first populated with a list of URLs (seed) to start crawling from
- while (there are items in the queue)
    - Remove the first item (URL)
    - Extract a list of hyperlinks on the page and add these URLs to the queue
    - Download the page (if the page is "needed")

PERFORMING WEB SCRAPING

General Approach:

- Maintain a queue (or stack)
- The queue is first populated with a list of URLs (seed) to start crawling from
- while (there are items in the queue)
    - …
    - Extract data into a database ("web scraping")

Most of the time we are only interested in specific data in the page

PERFORMING WEB SCRAPING

Queue contains the front page of a website as a seed URL

PERFORMING WEB SCRAPING

Extract the list of hyperlinks and add them back to the queue

PERFORMING WEB SCRAPING

Extract useful content from it such as text or images

SPORT

Paralympics: Singapore's Theresa Goh wins bronze in 100m breaststroke SB4 final

By Philip Goh, Sports Editor Posted 12 Sep 2016 07:21 Updated 12 Sep 2016 10:28

Theresa Goh with her bronze medal at the 2016 Rio Paralympic Games. mediacorp

CAPTION

816   f   Email   More   A A

SINGAPORE: After more than a decade of trying, Singapore's Theresa Goh has finally made it to the podium at the Paralympics.

On Monday (Sep 12) morning, Singapore time, the 29-year-old Goh touched home in third place in the women's SB4 100m breaststroke final in in Rio de Janeiro.

Her time of 1min 55.55s was a second off the Asian record she had set in the morning heats when she qualified second fastest behind world record holder Sarah Louise Rung of Norway.

HAPPENIN

Latest New

12:10 BUSINESS

12:05 ENTERTAINMENT   20

11:45 ENTERTAINMENT   Box Offic million US ope

11:44 ASIA PACIFIC   China, Russia to stage military drills in South China Sea

<SPONSORED> Knight Rider alert: How Audi light technology saves lives

<ADV> One Day in Taiwan? Don't Miss the Night Markets

11:06 SINGAPORE   StarHub releases iPhone 7 prices

Manulife Educate

Manulife

11:05 ASIA PACIFIC   South Korea's Yonhap says US delays sending B-1 bomber to Korean peninsula due to weather

10:40 ASIA PACIFIC   Japan's Abe says North Korea nuclear tests "absolutely unacceptable"

10:34 SINGAPORE   PM Lee wishes Muslims Selamat Hari Raya Haji

34

TECHNIQUES FOR WEB SCRAPING

The following are some of the techniques for doing web scraping:

- Extracting content from HTML source
- Extracting content using a HTML parser
- Web Scraping using APIs
- Scraping using an actual browser/headless browser

EXTRACTING CONTENT FROM HTML SOURCE

| Introduction to Web Content Mining | Web Basics | Techniques for performing Web Content Mining | Extracting Content from HTML Source |

STRING MATCHING

One possible way to do web scraping is to download the HTML source and perform string-based matching

- This is often achieved through to use of regular expressions

- A regular expression is a sequence of characters which defines a search pattern for matching string or for extracting string

STRING MATCHING WEB SCRAPER

```python
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
  with mining data from the web the web which can then be
  transformed into knowledge</body></html>"

pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"

#group 1
title = re.search(pattern, html).group(1)
title

#group 2
description = re.search(pattern, html).group(2)
description
```

STRING MATCHING WEB SCRAPER

```python
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
  with mining data from the web the web which can then be
  transformed into knowledge</body></html>"

pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"

#group 1
title = re.search(pattern, html).group(1)
title

#group 2
description = re.search(pattern, html).group(2)
description
```

Import the regular expression library (`re`)

STRING MATCHING WEB SCRAPER

```python
import re

html = "<html><body>Web Mining
  with mining data from the web
  transformed into knowledge</body></html>"


pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"


#group 1
title = re.search(pattern, html).group(1)
title

#group 2
description = re.search(pattern, html).group(2)
description
```
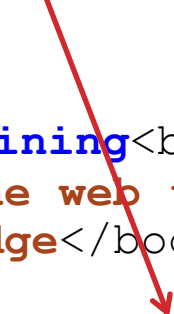
search() takes 2 parameters :
pattern, **string**

STRING MATCHING WEB SCRAPER

```
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
  with mining data from the web the web which can then be
  transformed into knowledge</body></html>"


pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"
```

Notice some special characters: ^ (.*) $

^ indicates the start of the string
( ) defines a new capturing group
. indicates any character
* indicates 0 or more times
.* indicates a sequence of any character occurring 0 or more times
$ indicates the end of the string

41

STRING MATCHING WEB SCRAPER

```python
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
  with mining data from the web the web which can then be
  transformed into knowledge</body></html>"


pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"

#group 1
title = re.search(pattern, html).group(1)
title

#group
des
des
```

^ indicates the start of the string
$ indicates the end of the string

By using ^ and $, there will either be one match or no match (there won't be multiple matches)

42

STRING MATCHING WEB SCRAPER

```python
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
  with mining data from the web the web which can then be
  transformed into knowledge</body></html>"


pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"


#group 1
title = re.search(pattern, html).group(1)
title


#group 2
description = re.search(pattern, html).group(2)
description
```
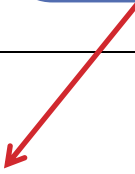
STRING MATCHING WEB

group(1) will retrieve the contents of the first capturing group

```python
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
  with mining data from the web the web which can then be
  transformed into knowledge</body></html>"


pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"


#group 1
title = re.search(pattern, html).group(1)
title


#group 2
description = re.search(pattern, html).group(2)
description
```

STRING MATCHING WEB

> group(2) will retrieve the contents of the second capturing group

```python
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
  with mining data from the web the web which can then be
  transformed into knowledge</body></html>"


pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"


#group 1
title = re.search(pattern, html).group(1)
title


#group 2
description = re.search(pattern, html).group(2)
description
```

STRING MATCHING WEB

```
import re

html = "<html><body>Web Mining<br/>Web mining is concerned
with mining data from the web the web which can then be
transformed into knowledge</body></html>"


pattern = "^<html><body>(.*)<br/>(.*)</body></html>$"


#group 1
title = re.search(pattern, html).group(1)
title


#group 2
description = re.search(pattern, html).group(2)
description
```

REGULAR EXPRESSION (REGEX) REFERENCES

https://www.regular-expressions.info/python.html

https://www.dataquest.io/wp-content/uploads/2019/03/python-regular-expressions-cheat-sheet.pdf

EXTRACTING MATCHING PATTERNS
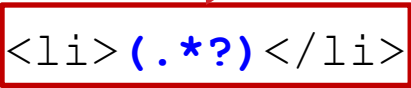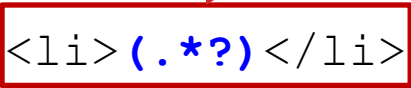
```python
import re

html2 = "<html><body>
  <ul><li>johndoe@gmail.com</li><li>janedoe@gmail.com</li>
  <li>Robin</li></ul></body></html>"

pattern2 = "<li>(.*?)</li>"

list_values = re.findall(pattern2, html2)
list_values
```

EXTRACTING MATCHING PATTERNS

```python
import re

html2 = "<html><body>
  <ul><li>johndoe@gmail.com</li><li>janedoe@gmail.com</li>
  <li>Robin</li></ul></body></html>"

pattern2 = "<li>(.*?)</li>"

list_values = re.findall(pattern2, html2)
list_values
```

Pattern to match any string that starts with `<li>` followed by (zero or more characters) and ends with `</li>`

EXTRACTING MATCHING PATTERNS

```python
import re

html2 = "<html><body>
  <ul><li>johndoe@gmail.com</li><li>janedoe@gmail.com</li>
  <li>Robin</li></ul></body></html>"


pattern2 = "<li>(.*?)</li>"


list_values = re.findall(pattern2, html2)
list_values
```

Notice that there is a `?` which indicates that we are doing a **non-greedy** matching (3 matches in total)

.* will match greedily (i.e. highlighted section)

EXTRACTING MA...

```python
import re

html2 = "<html><body>
  <ul><li>johndoe@gmail.com</li><li>janedoe@gmail.com</li>
  <li>Robin</li></ul></body></html>"


email_pattern = "<li>(.*?)</li>"


emails = re.findall(email_pattern, html2)
emails
```

Another example (Real website)

Suppose we want to extract the contents under the INTRODUCTION

INTRODUCTION

The National University of Singapore (NUS) School of Computing (SOC) Summer Workshop 2021 is the third edition of the Summer Workshop organized by NUS, School of Computing.

This year's workshop will mark the return of the popular programme as 2020's workshop was cancelled amid the COVID-19 pandemic. It will be held **online** entirely in view of the on-going global pandemic.

Our highly-interactive programme aims to integrate theory into practice. In addition to providing a deep focus in high-demand computing fields (e.g. Analytics, Artificial Intelligence, FinTech, Media and IoT), it highlights the real-life relevance of these specialised knowledge through hands-on application.

Students gain valuable industry insights and knowledge through lectures and have opportunities to hone and flex their computing skills through projects.

You may also check out the previous runs of the Summer Workshop via the links here!

Promotional Slides   Publicity Poster

ANNOUNCEMENTS**

Phase 1 Details

28 April 2021

First figure out the rough position in the page using the web browser's **Inspect Element** feature

Then find it View Source window (because the browser might clean up the page slightly such as removing unnecessary spaces)

```python
import re
import requests

page = requests.get("https://sws.comp.nus.edu.sg/")

html3 = page.content.decode("utf-8")

#. by default does not match for newline characters
#re.DOTALL - will make . match even for newline
pattern3 = '<div
  id="WEcc664a539a".*?>.*?<span.*?>(.*?)</span>.*?</div>'
results = re.search(pattern3, html3, re.DOTALL)

extracted_text = results.group(1)

#remove some html
#\\1 - refers to group 1
extracted_text = re.sub("<br />", "\n", extracted_text)
extracted_text = re.sub(" ", " ", extracted_text)
extracted_text = re.sub("<.*?>(.*?)</.*>", "\\1",
  extracted_text)
```

SUMMARY

Introduction to Web Mining

- Web Scraper, Web Crawler

Web Basics

- HTML

Techniques for performing Web Content Mining

- Extracting Content from HTML Source

WHAT'S NEXT?

Predictive Analytics I

Predictive Analytics II

Mining Web Content II

Mining Web Content III

Recommender System