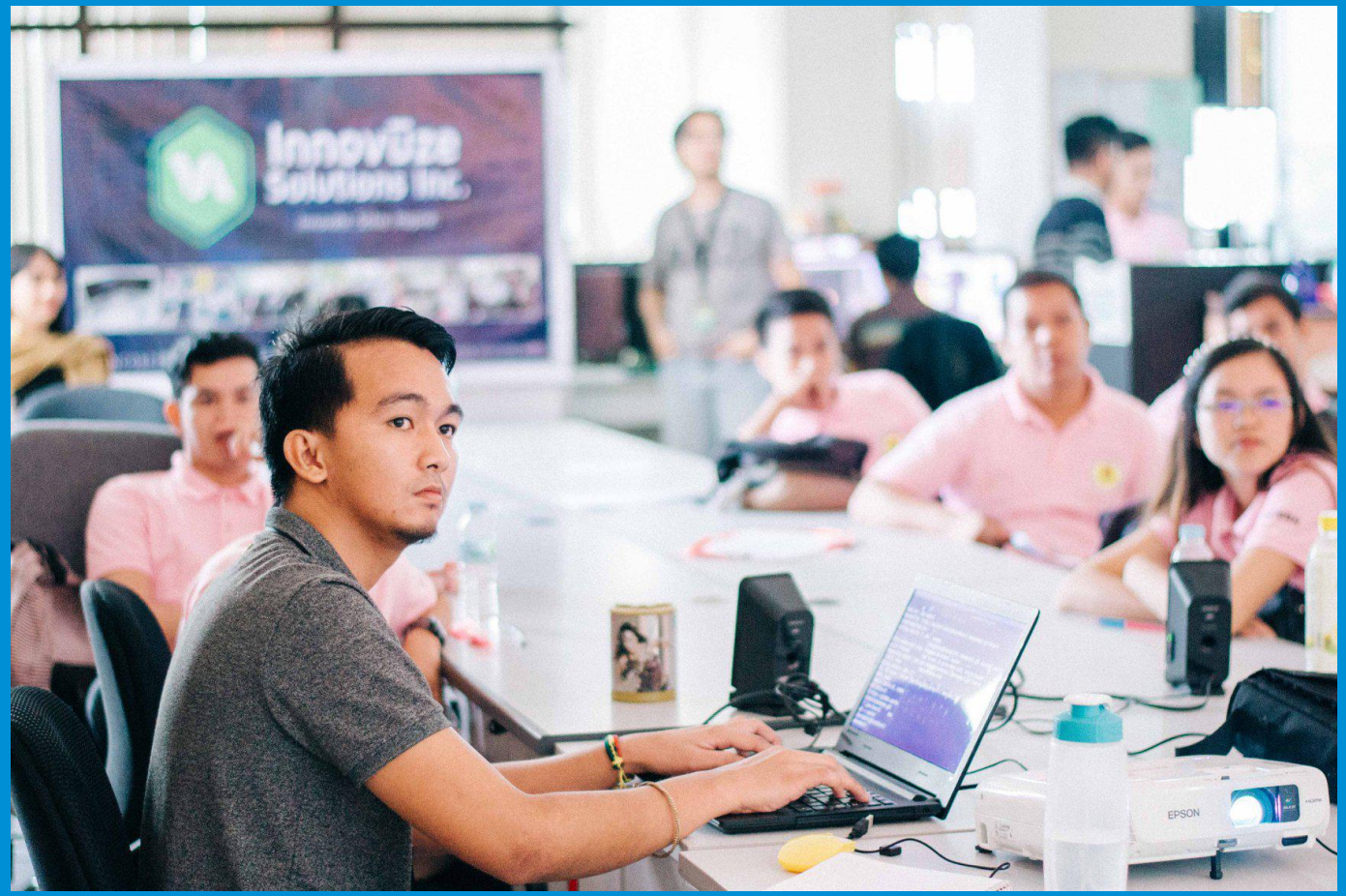
Web Scraping with Beautifulsoup

@harriebird

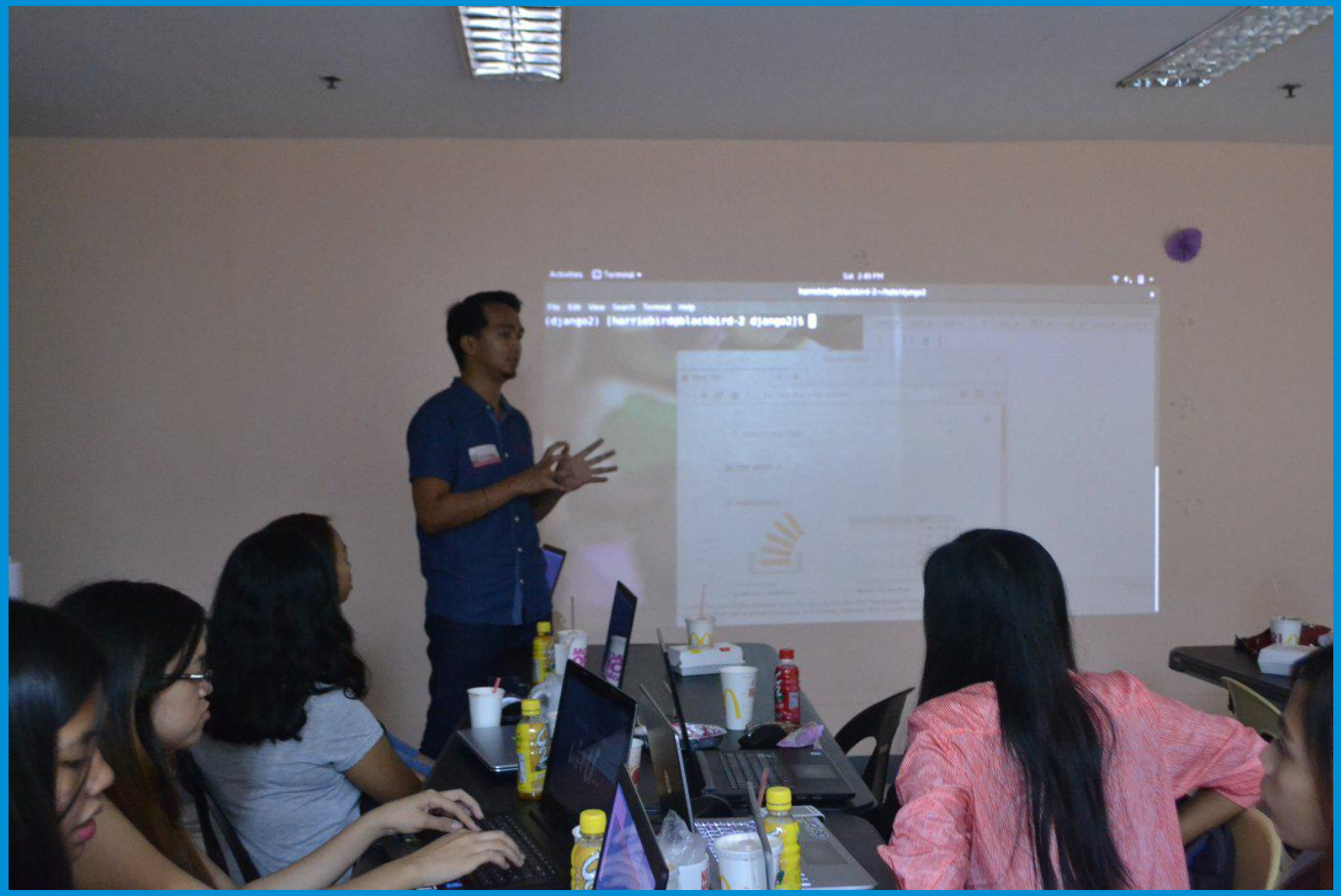
whoami && groups

- Paul Harriet Asiñero / </harriebird>
- Web Developer at Innovuze Solutions Inc.
- PineapplePy, Hackm3 member
- Open-source software loverboy

Previously...



Previously...



Previously...

Previously...

#ATM

Agenda

- Introduce Web scraping
- Introduce Beautifulsoup
- Demo Time

Needs

- Python 3
- beautifulsoup4
- background in HTML (tags, ids, classes)

Web Scraping Overview

What is Web Scraping?

- it is the practice of gathering data through any means other than a program interacting with an API

What is Web Scraping?

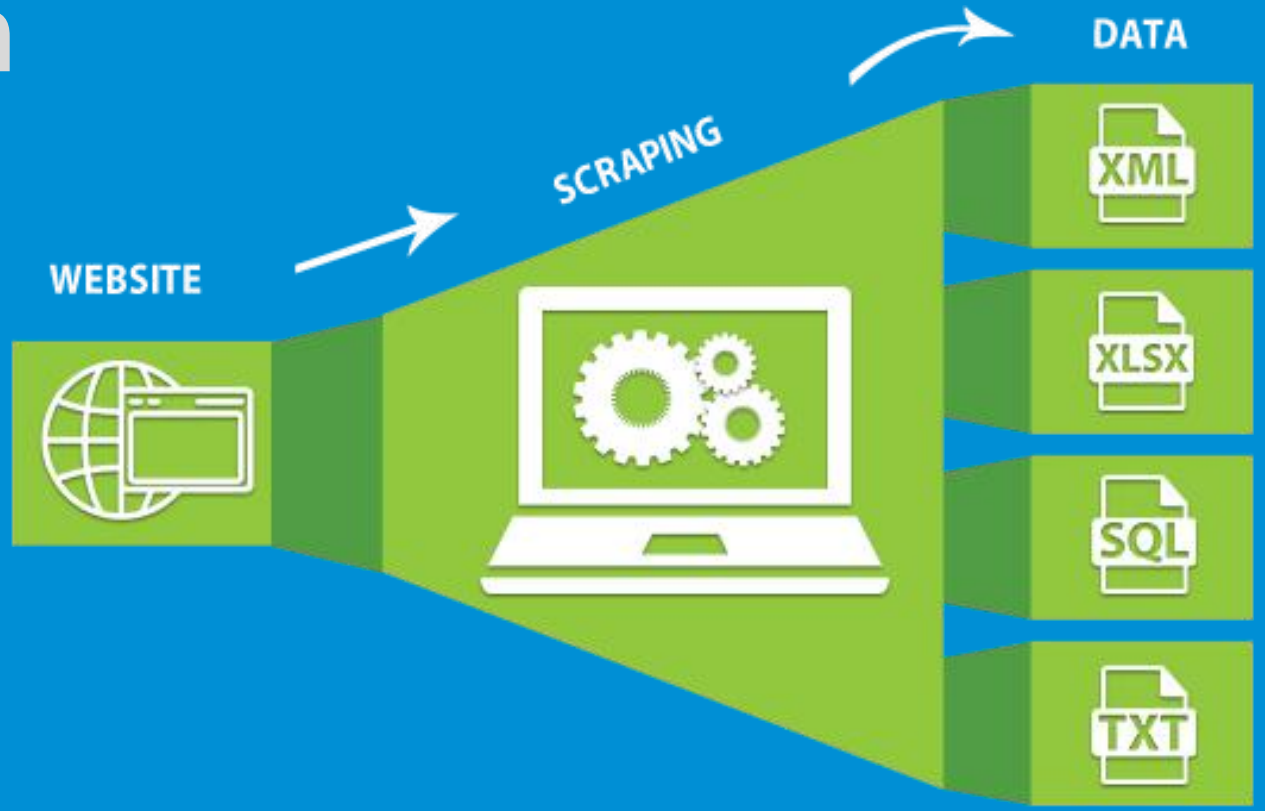
- Commonly accomplished by writing an automated program that queries the web server, requests data, and then parses that data to extract needed information

Why Scrape the Web?

- valuable data is coming from a website
- the website does not provide any API
- the website has an API but with limitations

Web Scraping Workflow

- Retrieve HTML data from a web site
- Parse the data for target information
- Store the target information



BeautifulSoup

- is a Python library for pulling out data out of HTML and XML files
- simplifies web scraping

Using Beautifulsoup

Initializing Beautifulsoup

- from bs4 import BeautifulSoup
- soup = BeautifulSoup(html,'html.parser')

Navigating the parse tree

- find()
  - searches the soup and returns one result
- findAll()
  - searches the soup and returns all the results matching to the criteria
- select()
  - same with findAll() but uses CSS selectors in searching

Navigating the parse tree

- soup.body.p
  - tag names can be used to navigate
- getText()
  - removes the tags enclosing the text and returns it
- div.a['href']
  - returns the value of the attribute specified

Request data across the web

- request from urllib
  - can be helpful in making web requests
- urlopen(url)
  - opens the URL/Request object passed and returns the response contents
- Request()
  - used to create an advanced request to website like adding headers

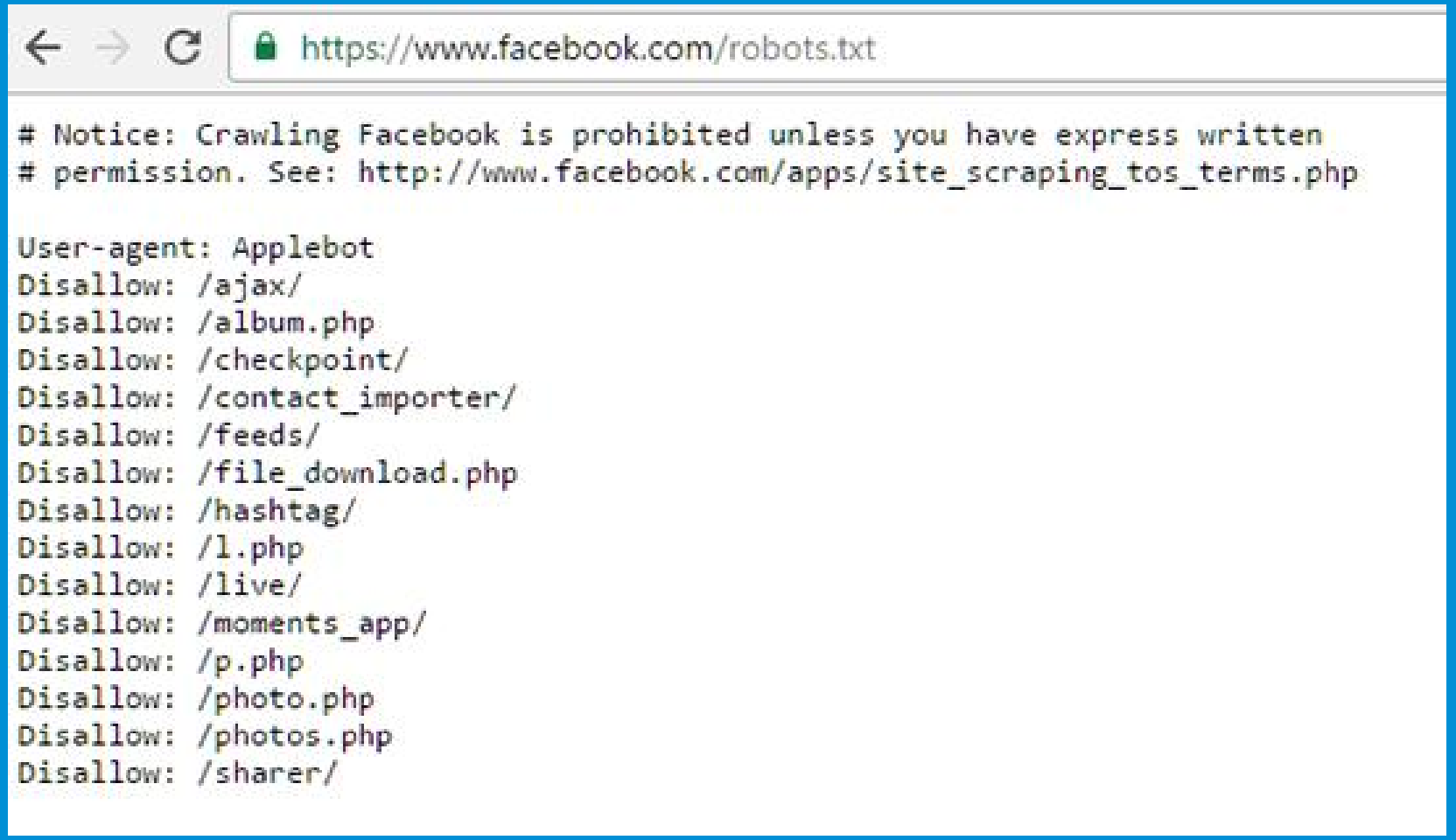
Saving data to a CSV file

- can be achieved by using the csv library
- csv.writer(fileobj)
  - specifies where the csv data will be saved
- writerow(row)
  - write the row to a file (method from csv module)

Before we dive into the Demo

Read the Terms and Conditions of the website

Check the robots.txt on the website




IPAGLABAN
MO

Some websites block scrapers.

- modification of headers could help in this case

Demo Time

PineapplePy

- facebook.com/groups/pineapplepy

Hackm3

- facebook.com/hackm3.ph