Web Scraping
Introduction

- Gathering of data from the internet
- Also known as screen scraping or web harvesting *(or data mining in a trivial sense)*
- Practice of gathering data by writing an automated program that:
  - Queries a web server
  - Requests data (HTML and other files that comprise web pages)
  - Parses that data to extract meaningful information

Why Web Scrapers?

- Internet is awash in data, browsers are handy to “browse” through information but not to collect and sift through data
- Web scrapers are excellent at gathering and processing large amounts of data
- Data is often unstructured or semi-structured
- Data often isn’t made available as a formal data set
- Scraping opens up a new world of data to researchers

Web Scraping Data

- Publicly accessible data that doesn't come neatly packaged as a formal data set …
  - online classified ads and rental housing data
  - social media and behavioral data
  - discussion forums and sentiment analysis
  - auction sites and retail price data
- Leading to extremely practical applications
  - market forecasting
  - medical diagnostics
  - natural/machine language translation
- Is Web Scraping legal?!

Legal Considerations

- Procedurally similar to browsing and indexing: Is it publicly available information?
- Do you need to log in to access it?
- Do the terms of service explicitly forbid scraping?
- Are you using the data in a way that harms the source?
- Ethical questions
  - Can you partner with the data source's organization?
  - Proper attribution

Terminology

- **Spider** – crawls the web by following links
- **Crawler** – just another name for a spider
- **Data Scraper** – generic computer program that extracts human-readable data
- **Web Scraper** – a data scraper specifically for web pages
- **Internet bot/web robot**, or simply **bot** is a software program that runs automated tasks over the Internet

Web Scraper specifics

- Small computer program that: accesses web pages
- Finds specified data elements on the page
- Extracts them (and transforms them if necessary)
- Compiles this data into a coherent data set

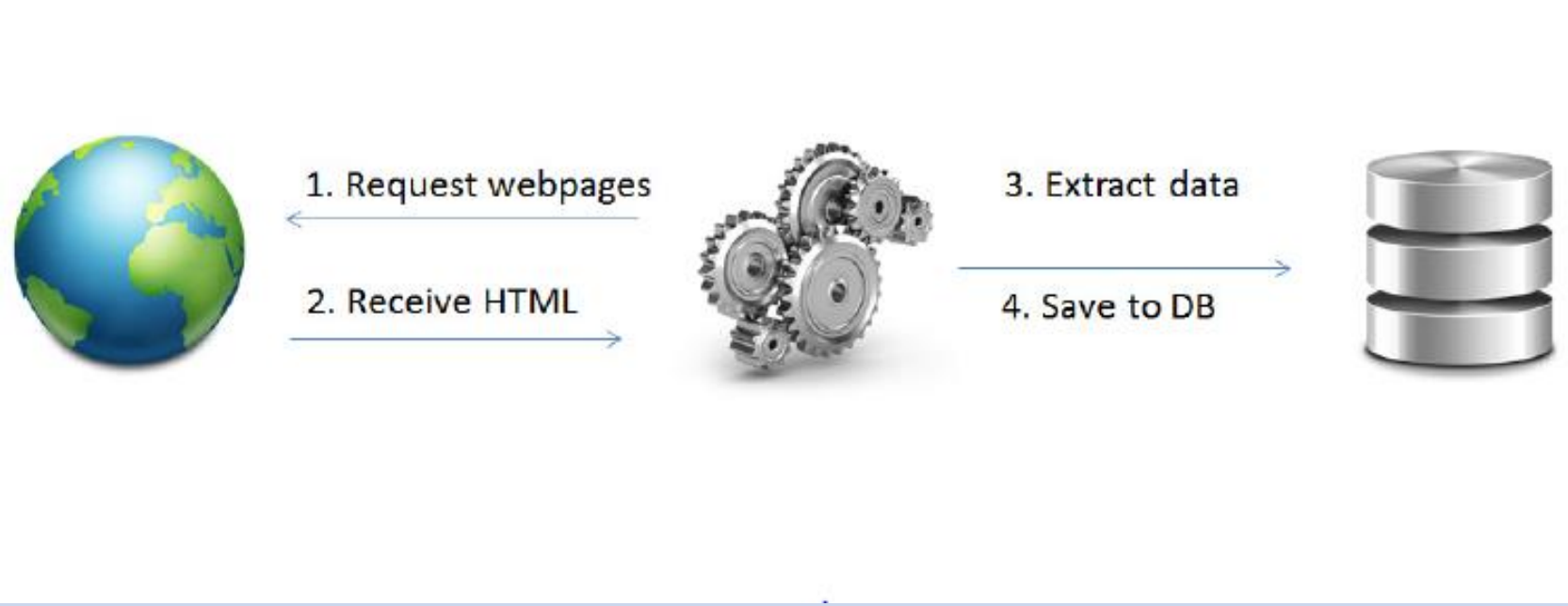
Web browser Vs Scraper

- Compare Scraper's behavior to that of a web browser
- Scraper can:
  - be run iteratively over many web pages
  - access data spread across thousands or millions of pages
  - construct large, robust data sets out of otherwise messy text that would only appear in your web browser

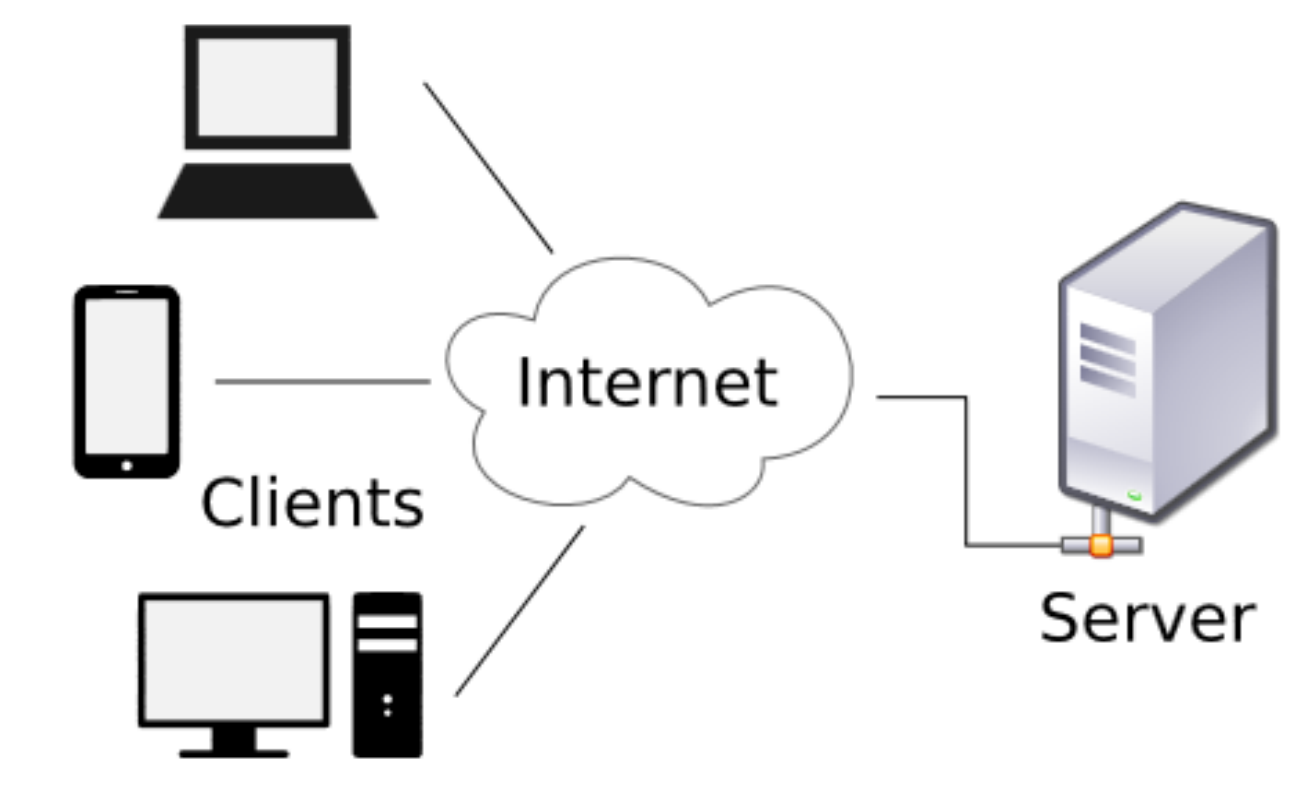
Web Scraping basics

- Traditional data transfer of unambiguous structured data
  - not very human-readable
- APIs application programming interfaces
  - provides external access to some software, data, or service
  - not all applications have APIs
  - Web scraper can be written as a custom API for some web site of interest
- Web pages
  - unstructured or semi-structured data
  - very human-readable
  - Challenge is to convert semi-structured data on web pages into a structured data set on your hard drive

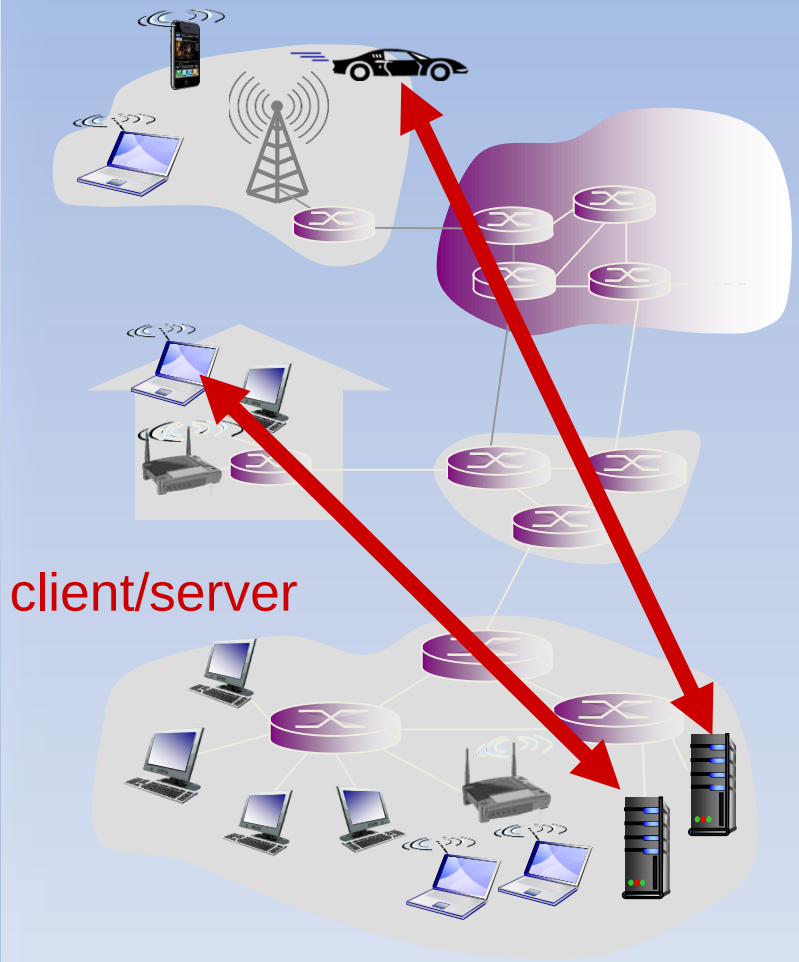
Web Scraping basics



Client Server Model



Client-server architecture



server:

- always-on host
- permanent IP address
- data centers for scaling

clients:

- communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

Web and HTTP

- *web page* consists of *objects*
- object can be HTML file, JPEG image, Java applet, audio file,…
- web page consists of *base HTML-file* which includes *several referenced objects*
- each object is addressable by a *URL,* e.g.,

`www.someschool.edu/someDept/pic.gif`

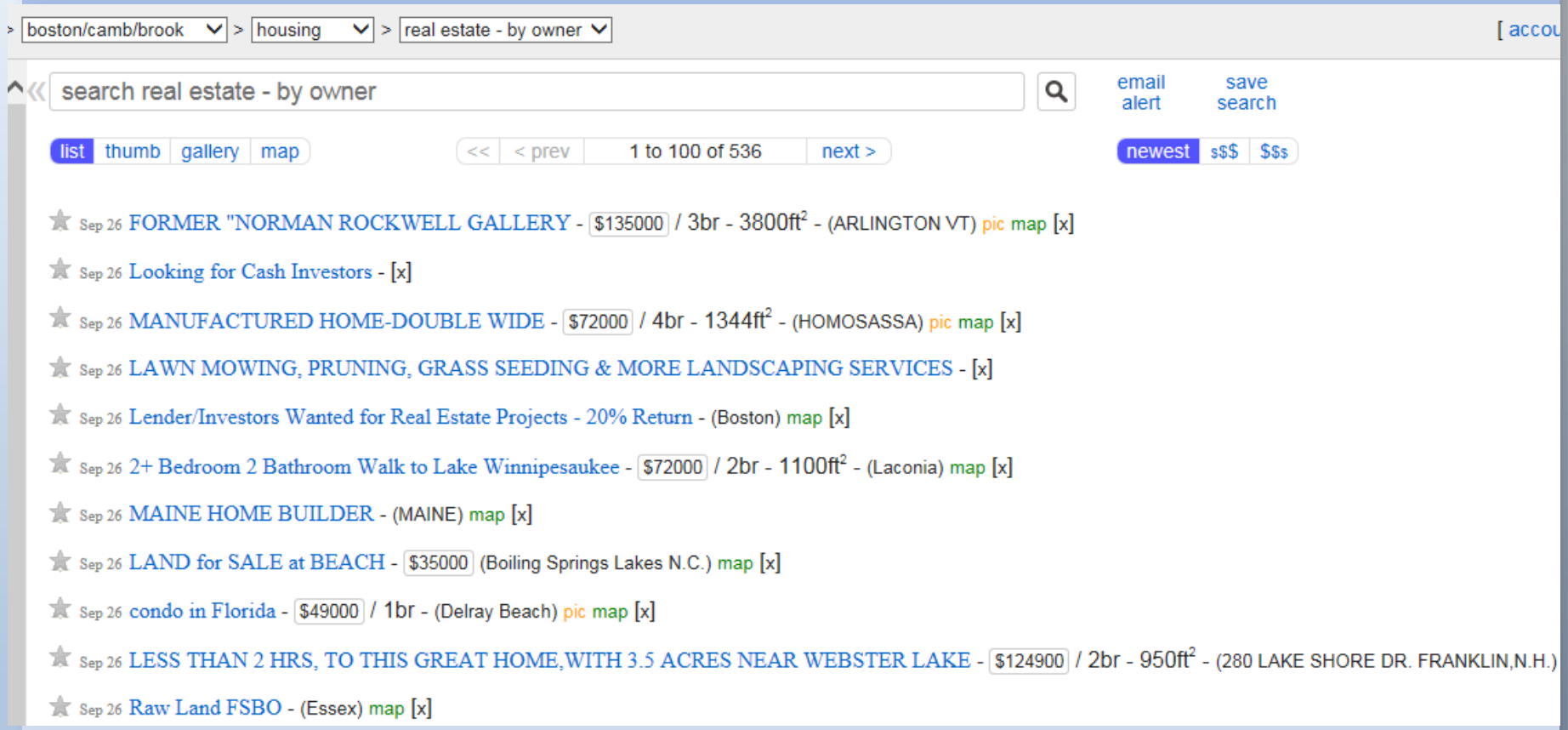
host name | path name

HTTP overview

HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client:* browser that requests, receives, (using HTTP protocol) and "displays" Web objects
  - *server:* Web server sends (using HTTP protocol) objects in response to requests

Semi-Structured data



- Human-readable listing
- How do we see or capture all of this rental market data?
- Right-click and view source…

Web page source code

```
<!DOCTYPE html>

<html class="no-js"><head>
    <title>boston real estate - by owner  - craigslist</title>

    <meta name="description" content="boston real estate - by owner  - craigslist">
    <link rel="canonical" href="https://boston.craigslist.org/search/gbs/reo">
    <link rel="alternate" type="application/rss+xml" href="https://boston.craigslist.org/search/gbs/reo?format=rss" title="RSS feed for craigslist | boston real estate - by owner  - craigslist ">

    <link rel="next" href="https://boston.craigslist.org/search/gbs/reo?s=100">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=1">
    <link type="text/css" rel="stylesheet" media="all" href="//www.craigslist.org/styles/cl.css?v=0b04254c2a971c33c5543fa6344f7dad">
    <link type="text/css" rel="stylesheet" media="all" href="//www.craigslist.org/styles/tocs.css?v=5f8a8e4c2f016a0da7b16fc9394a62e9">
    <link type="text/css" rel="stylesheet" media="all" href="//www.craigslist.org/styles/jquery-ui-1.9.2.custom.css?v=86db72e37d2cf63cab3c63485c71becb">
    <link rel="prefetch" href="//www.craigslist.org/js/postings-concat.min.js?v=2c267741ac82581ef0aff79d8144768b">

    <link type="text/css" rel="stylesheet" media="all" href="//www.craigslist.org/styles/leaflet-stock.css?v=969bf0c010aad78a472cb96ed5c1d1bc">
<link type="text/css" rel="stylesheet" media="all" href="//www.craigslist.org/styles/MarkerCluster.css?v=c9937ed03fbb57f185493cd8c283efeb">
<link type="text/css" rel="stylesheet" media="all" href="//www.craigslist.org/styles/tocsmaps.css?v=0713fba1eea156d1e29815594450b352">
<!--[if IE]>
    <link type="text/css" rel="stylesheet" media="all" href="//www.craigslist.org/styles/tocsmaps-ie.css?v=b21742b2ebe243963373e956ea115db0">
<![endif]-->

        <script type="text/javascript"><!--
            var expiredFavIDs = [];
var subarea = "gbs";
```

HTML

- HTML the markup language for web pages
  - consists of HTML elements defined by tags
  - elements can have attributes and can encapsulate text
  - creates a navigable, structured document
- Element = individual component of HTML
- Elements represent semantics
- Written with a start and end tag Tags use angle brackets
- Example: <title>this is a title</title>
- Elements can have attributes
  - Example: this is a paragraph