

Scraping data from the web

With the move toward open data, organizations are increasingly making their data available online. But there is a good deal of variation in what, precisely, it means to publish data on the web.

Take data on crime or traffic incidents in NYC. The NYPD makes various PDF documents and Excel files for aggregate data, tabulated citywide, by borough or by precinct.

Incident-level data is rolled up to the nearest full quarter, although weekly updates are available via their CompStat 2.0 portal.

 City Wide Crime Stats - NYPD <https://www1.nyc.gov/site/nypd/stats/crime-statistics/citywide-cr...>

New York City Police Department
New York's Finest

NYPD

311 | Search all NYC.gov websites

简体中文 ▶ Translate | ▾ Text-Size

[Home](#) About Bureaus Services **Stats** Media Careers [Search](#)

Crime Statistics **Traffic Data** **Reports, Analyses**

Citywide Crime Statistics

Borough and Precinct Crime Statistics

The NYPD provides overall citywide statistics, which are updated weekly. The reports can be viewed below. The department's advanced digital version can be viewed at [CompStat 2.0](#).

- [City Wide Crime Statistics Weekly \(PDF\)](#)
[City Wide Crime Statistics Weekly \(Excel\)](#)
- [City of New York Department of Correction Crime Statistics \(PDF\)](#)
[City of New York Department of Correction Crime Statistics \(Excel\)](#)
- [New York City Housing Authority Crime Statistics Weekly \(PDF\)](#)
[New York City Housing Authority Crime Statistics Weekly \(Excel\)](#)

Incident Level Data

The NYPD releases a number of incident level datasets related to police enforcement and criminal activity. By releasing this information, the Department aims to increase transparency and foster collaboration, with a goal of continually improving police-community relationships through the use of open data.

Historic (updated annually);

- [Incident Level Arrest Data - 2013 through most recent full year](#)
- [Incident Level Summons Data - 2013 through most recent full year](#)
- [Incident Level Shooting Data - 2013 through most recent full year](#)

[!\[\]\(6cbaff651e9b7a1a7462c49d18e0be2e_img.jpg\) Share](#)
[!\[\]\(1855b11bf6aa350ebef50973960dd134_img.jpg\) Print](#)

 Borough and Precinct Crime St <https://www1.nyc.gov/site/nypd/stats/crime-statistics/borough-a...>

New York's Finest

NYPD

繁體中文 ▶ Translate | ▾ Text-Size

[Home](#) **About** Bureaus Services **Stats** Media Careers [Search](#)

Crime Statistics **Traffic Data** **Reports, Analyses**

Citywide Crime Statistics

Borough and Precinct Crime Statistics

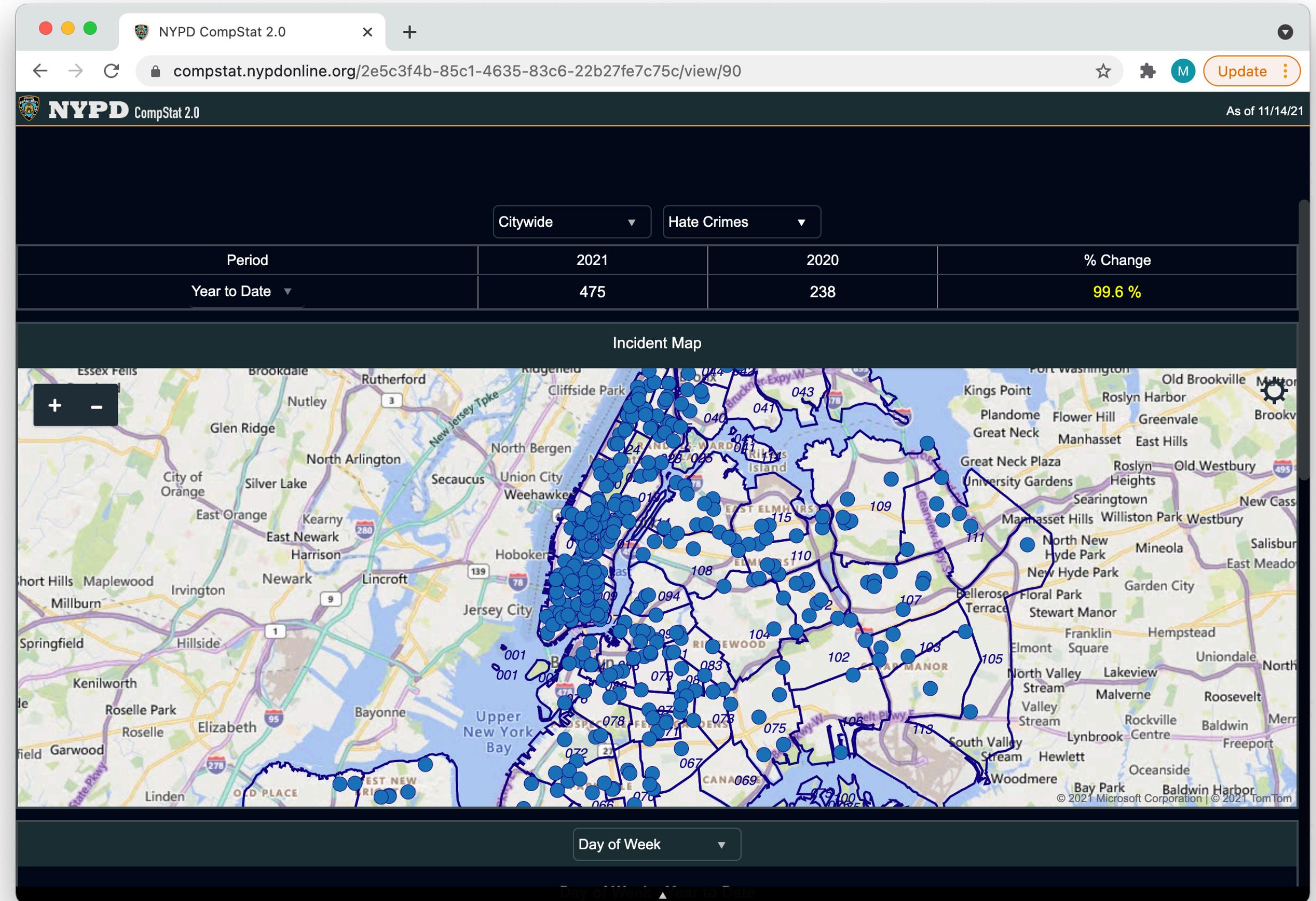
The NYPD provides statistics that are categorized by police borough and precinct. These reports are updated weekly and can be viewed below. The department's advanced digital version can be viewed at [CompStat 2.0](#).

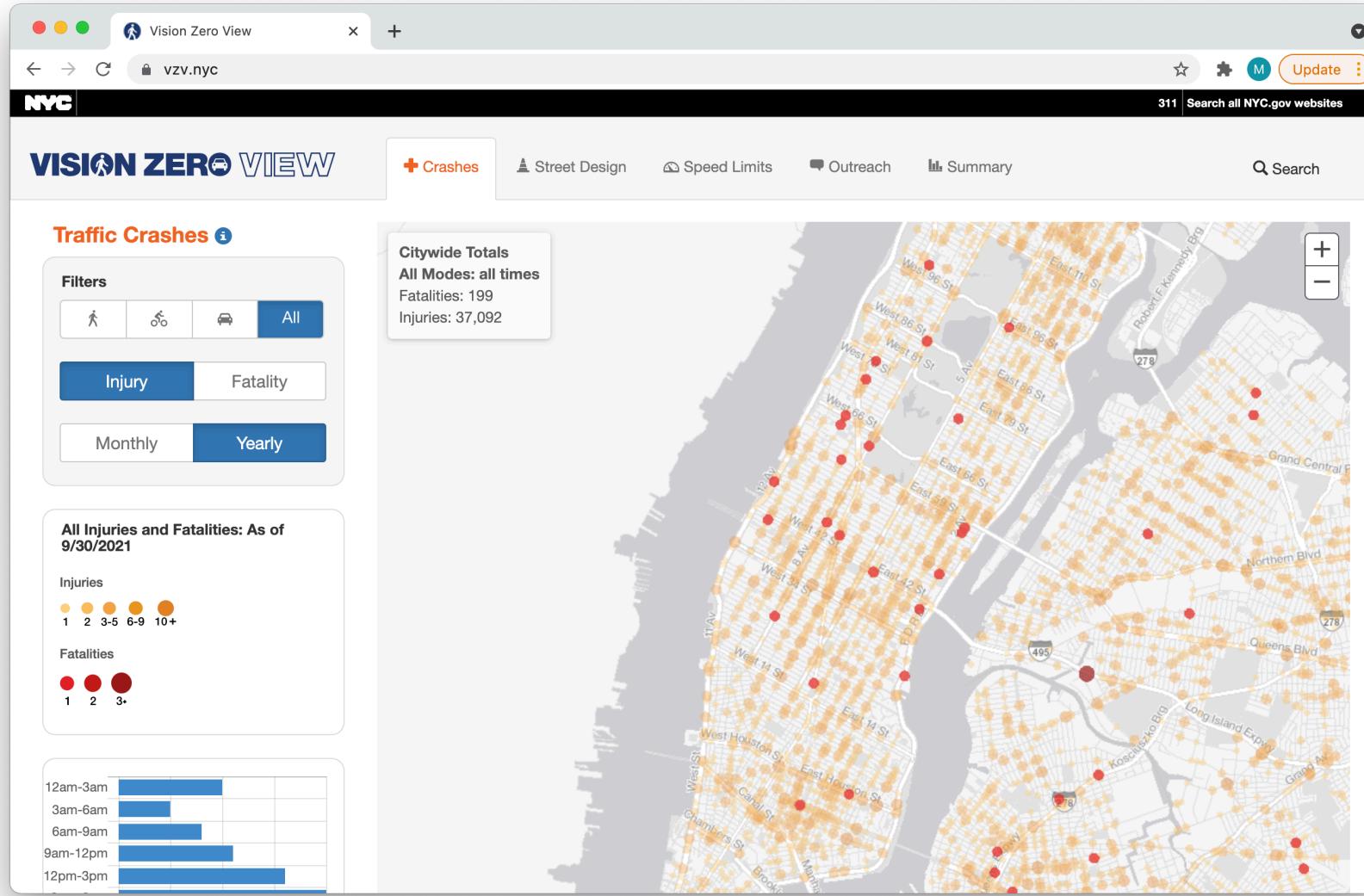
- [Bronx](#)
- [Brooklyn](#)
- [Manhattan](#)
- [Queens](#)
- [Staten Island](#)

Bronx

Patrol Borough Bronx

- [Bronx \(PDF\)](#)
[Bronx \(Excel\)](#)
- [40th Precinct \(PDF\)](#)
[40th Precinct \(Excel\)](#)
- [41st Precinct \(PDF\)](#)
[41st Precinct \(Excel\)](#)





NYC OpenData

Motor Vehicle Collisions - Crashes

The Motor Vehicle Collisions crash table contains details on the crash event. Each row ▾

Find in this Dataset

More Views **Filter** **Visualize** **Export** **Discuss** **Embed** **About**

CRASH DATE	CRASH TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	OR
02/26/2021	14:50	BRONX	10461	40.843464	-73.836	(40.843464°, -73.836°)	
04/15/2021	13:30	BRONX	10461	40.857365	-73.84657	(40.857365°, -73.846...)	
04/16/2021	17:40	BRONX	10474	40.815	-73.89402	(40.815°, -73.89402°)	GA
04/16/2021	16:35	BRONX	10475	40.890076	-73.819855	(40.890076°, -73.819...)	BO
04/14/2021	21:08	BRONX	10451	40.817696	-73.922615	(40.817696°, -73.922...)	
04/15/2021	12:16	BRONX	10460	40.841087	-73.86447	(40.841087°, -73.864...)	EA'
04/15/2021	20:13	BRONX	10457	40.84744	-73.89968	(40.84744°, -73.8996...)	EA'
04/15/2021	20:15	BRONX	10461	40.854485	-73.854645	(40.854485°, -73.854...)	
04/15/2021	17:25	BRONX	10475	40.861687	-73.82435	(40.861687°, -73.824...)	
04/10/2021	12:00	BRONX	10453	40.856014	-73.91213	(40.856014°, -73.912...)	WE
04/14/2021	9:33	BRONX	10472	40.826424	-73.85868	(40.826424°, -73.858...)	BR
04/13/2021	9:00	BRONX	10475	40.870056	-73.83222	(40.870056°, -73.832...)	
04/15/2021	11:41	BRONX	10468	40.85968	-73.90427	(40.85968°, -73.9042...)	

< Previous **Next** >

Showing Motor Vehicle Collisions 1 to 100 out of 1,841,953

[Privacy Policy](#) [Terms of Use](#) [Contact Us](#) [FAQ](#)

© 2021 City of New York. All Rights Reserved. NYC is a trademark and service mark of the City of New York.

NYC Open Data - opendata.cityofnewyork.us

NYC | OpenData 311 | Search all NYC.gov websites

Home Data About Learn Contact Us Blog Sign In

NYC OpenData

Open Data for All New Yorkers

Open Data is free public data published by New York City agencies and other partners. [Share your work during Open Data Week 2021](#) or [sign up for the NYC Open Data mailing list](#) to learn about training opportunities and upcoming events.

Search Open Data for things like 311, Buildings, Crime

Open Data for All 2021 Progress Report

Acknowledgments About Contact

1 Introduction 2 Strategic Plan Update 3 NYC Open Data Timeline 4 2021 Dataset Highlights 5 Open Data By The Numbers 6 2021 Compliance Plan

Learn about the next decade of NYC Open Data, and read our 2021 Report

Translate »

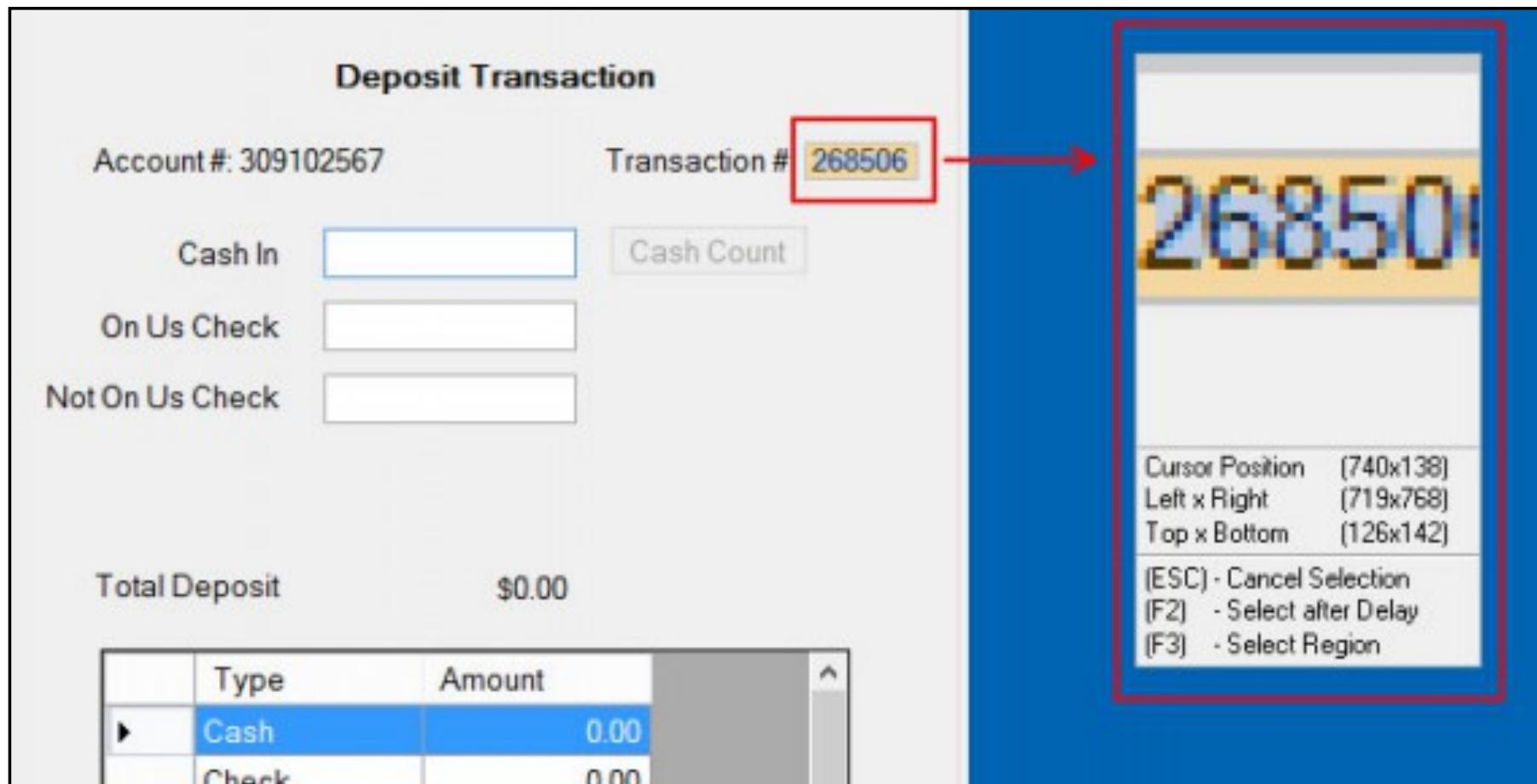
Scraping data from the web

Whenever possible, we would like to find data published for us to use in a sensible format. With luck, data are in an easily accessible portal or, if we ask nicely, someone will email us a copy of the data we are after (email or Dropbox or SecureDrop or whatever seems appropriate given sensitivities around the data).

That is, the verb “scrape” implies we expend some amount of effort, or apply a degree of force. All things being equal, we’d prefer not to work that hard if we don’t have to. **So the first rule of scraping is that we don’t want to do it** — we should always first make sure data are not available through simpler means.

Scraping data from the web

So what do we mean by “scraping” or more generally “data scraping”? Back in the day, we might read data from a program output that was being shown on a screen. A “screen scraper” would read data from the user interface, perhaps simulating user input to collect a multiple fields of information.



Scraping data from the web

In a similar way, “web scraping” involves taking data that was formatted for one purpose and translating it for another. Let’s take a simple example. Look at the page on crime statistics by city on Wikipedia.

You might be interested in the table that appears toward the bottom of the article describing the crime rates for the largest metropolitan areas in the US.

The screenshot shows a Wikipedia article titled "Crime in the United States". The sidebar on the left contains a table titled "United States" with data on crime rates for 2016. The table is divided into "Violent crimes" and "Property crimes" sections. The "Violent crimes" section includes Homicide (5.3), Robbery (102.8), Aggravated assault (248.5), and Total violent crime (386.3). The "Property crimes" section includes Burglary (468.9), Larceny-theft (1,745.0), Motor vehicle theft (236.9), and Total property crime (2,450.7).

United States	
Crime rates* (2016)	
Violent crimes	
Homicide	5.3
Robbery	102.8
Aggravated assault	248.5
Total violent crime	386.3
Property crimes	
Burglary	468.9
Larceny-theft	1,745.0
Motor vehicle theft	236.9
Total property crime	2,450.7

Scraping data from the web

In this format, on the web page, there is a limit to what you can do with these data aside from simply reading them. And that's basically the point of a web page — information is formatted as a “document” that is meant to be read by a human.

But what if we wanted to compute an average rate for these areas or wanted to compare the violent and property crime rates? OK silly actions for these data but the point is that on the web page they are somewhat “inert”.

The screenshot shows a web browser window with the URL https://en.wikipedia.org/wiki/Crime_in_the_United_States. The page content includes a section titled "Metropolitan areas" with a table titled "Crime in ten largest metropolitan areas (2011)".

Metropolitan statistical area	Violent crime rate	Property crime rate
New York-Northern New Jersey-Long Island, NY-NJ-PA MSA	406.0	1744.1
Los Angeles-Long Beach-Santa Ana, CA MSA	405.4	2232.7
Chicago-Joliet-Naperville, IL-IN-WI MSA	357.2	2791.5
Dallas-Fort Worth-Arlington, TX MSA	358.4	3498.5
Houston-Sugar Land-Baytown, TX MSA	550.8	3576.9
Philadelphia-Camden-Wilmington, PA-NJ-DE-MD MSA	532.3	2747.3
Washington-Arlington-Alexandria, DC-VA-MD-WV MSA	334.6	2386.0
Miami-Fort Lauderdale-Pompano Beach, FL MSA	596.7	4193.3
Atlanta-Sandy Springs-Marietta, GA MSA	400.9	3552.0
Boston-Cambridge-Quincy, MA-NH MSA	374.7	2109.0

Below the table, there is a section titled "Number and growth of criminal laws" which discusses the number of federal crimes and the growth of criminal laws over time.

Metropolitan areas [edit]
Further information: [United States cities by crime rate](#)
Crime in metropolitan statistical areas tends to be above the national average; however, wide variance exists among and within metropolitan areas.^[68] Some responding jurisdictions report very low crime rates, while others have considerably higher rates; these variations are due to many factors beyond population.^[68] FBI crime statistics publications strongly caution against comparison rankings of cities, counties, metropolitan statistical areas, and other reporting units without considering factors other than simply population.^[68] For 2011, the metropolitan statistical area with the highest violent crime rate was the [Memphis metropolitan area](#), with a rate of 980.4 per 100,000 residents, while the metropolitan statistical area with the lowest violent crime rate was [Logan metropolitan area](#), with a rate of 47.7.^{[69][70]}
It is quite common for crime in American cities to be highly concentrated in a few, often economically disadvantaged areas. For example, [San Mateo County, California](#) had a population of approximately 707,000 and 17 homicides in 2001. Six of these 17 homicides took place in poor [East Palo Alto](#), which had a population of roughly 30,000. So, while East Palo Alto accounted for a mere 4.2% of the population, about one-third of the homicides took place there.^[71]
Crime in ten largest metropolitan areas (2011)^{[69][70]}

Number and growth of criminal laws [edit]
There are conflicting opinions on the number of [federal crimes](#),^{[72][73]} but many have argued that there has been explosive growth and it has become overwhelming.^{[74][75][76]} In 1982, the [U.S. Justice Department](#) could not come up with a number, but estimated 3,000 crimes in the [United States Code](#).^{[72][73][77]} In 1998, the [American Bar Association](#) (ABA) said that it was likely much higher than 3,000, but didn't give a specific estimate.^{[72][73]} In 2008, the [Heritage Foundation](#) published a report that put the number at a minimum of 4,450.^[73] When staff for a task force of the [U.S. House Judiciary Committee](#) asked the [Congressional Research Service](#) (CRS) to update its 2008 calculation of criminal offenses in the [United States Code](#) in 2013, the CRS responded that they lack the manpower and resources to accomplish the task.^[78]

See also [edit]

Scraping data from the web

Here we open a new sheet and in the A1 cell we use the `IMPORTHTML` function. Specifically, you call it using the command

```
=IMPORTHTML( url, item, number)
```

where `url` is the address of the web page we want to scrape a table from, `item` specifies either a “table” or a “list” (more on the latter shortly), and `number` specifies which table on the page we want (where we start counting from 1)

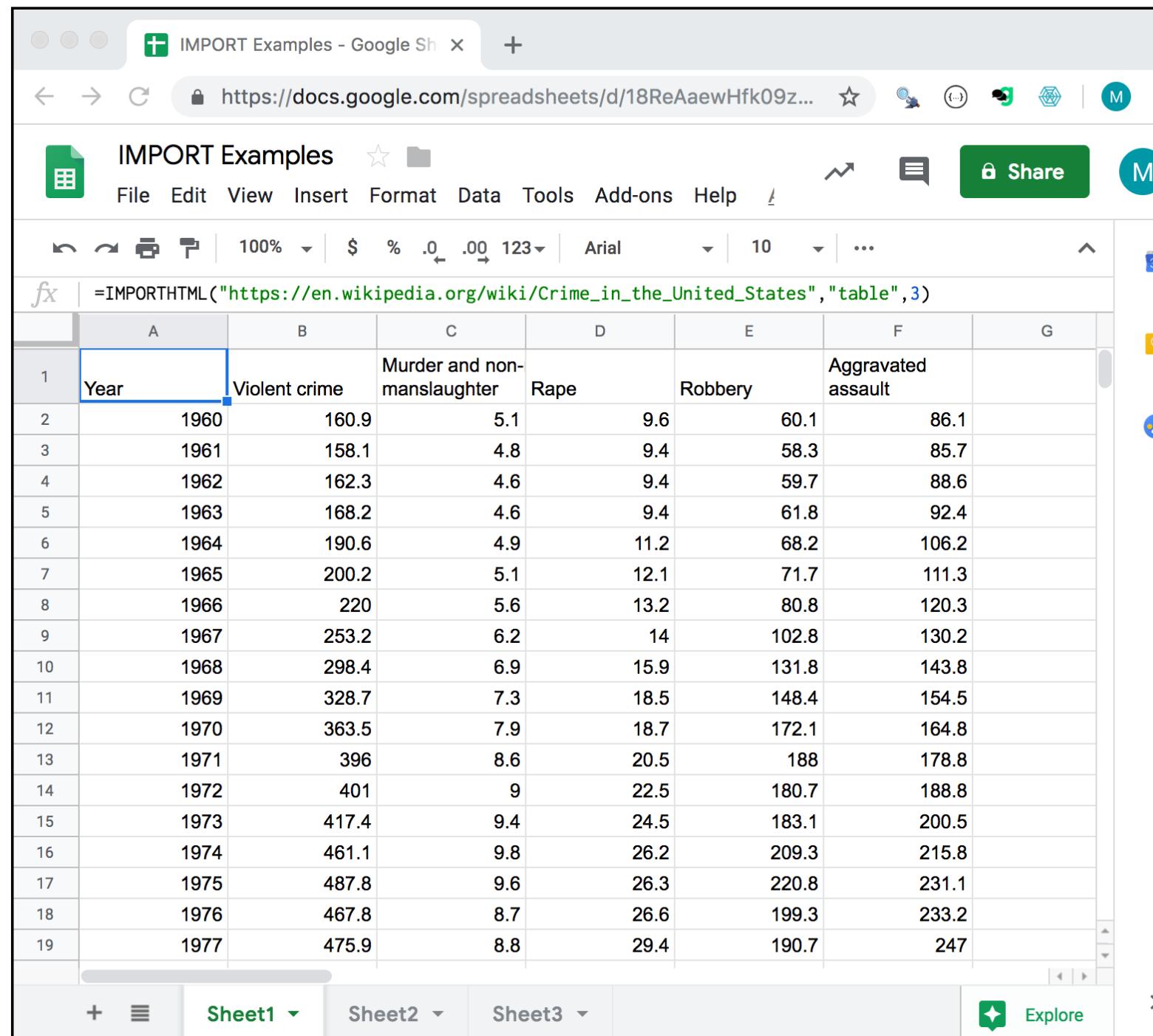
The table with crime rates for large metropolitan areas can be scraped into Google Sheets using the command below in cell A1 of a blank table.

```
=IMPORTHTML("https://en.wikipedia.org/wiki/Crime_in_the_United_States","table",8)
```

A screenshot of a Google Sheets spreadsheet titled "IMPORT Examples". The URL in the browser bar is <https://docs.google.com/spreadsheets/d/18ReAaewHfk09z...>. The formula in cell A1 is =IMPORTHTML("https://en.wikipedia.org/wiki/Crime_in_the_United_States","table",9). The table contains data for 11 metropolitan areas, with columns for Metropolitan stat, Violent crime rate, and Property crime rate. The data is as follows:

	A	B	C
1	Metropolitan stat	Violent crime rate	Property crime rate
2	New York-Northeast	406	1744.1
3	Los Angeles-Los Angeles-Long Beach	405.4	2232.7
4	Chicago-Joliet-Naperville	357.2	2791.5
5	Dallas-Fort Worth-Arlington	358.4	3498.5
6	Houston-Sugar Land-Baytown	550.8	3576.9
7	Philadelphia-Camden-Wilmington	532.3	2747.3
8	Washington-Arlington-Alexandria	334.6	2386
9	Miami-Fort Lauderdale-West Palm Beach	596.7	4193.3
10	Atlanta-Sandy Springs-Marietta	400.9	3552
11	Boston-Cambridge-Dorchester	374.7	2109
12			
13			
14			
15			
16			
17			
18			
19			
20			

```
=IMPORTHTML("https://en.wikipedia.org/wiki/Crime_in_the_United_States","table",2)
```



The screenshot shows a Google Sheets spreadsheet titled "IMPORT Examples". The formula bar contains the formula `=IMPORTHTML("https://en.wikipedia.org/wiki/Crime_in_the_United_States","table",2)`. The main table on the sheet displays data for violent crime in the United States from 1960 to 1977. The columns represent the year (A), violent crime rate (B), murder and non-manslaughter (C), rape (D), robbery (E), and aggravated assault (F). The data is as follows:

	A	B	C	D	E	F	G
1	Year	Violent crime	Murder and non-manslaughter	Rape	Robbery	Aggravated assault	
2	1960	160.9	5.1	9.6	60.1	86.1	
3	1961	158.1	4.8	9.4	58.3	85.7	
4	1962	162.3	4.6	9.4	59.7	88.6	
5	1963	168.2	4.6	9.4	61.8	92.4	
6	1964	190.6	4.9	11.2	68.2	106.2	
7	1965	200.2	5.1	12.1	71.7	111.3	
8	1966	220	5.6	13.2	80.8	120.3	
9	1967	253.2	6.2	14	102.8	130.2	
10	1968	298.4	6.9	15.9	131.8	143.8	
11	1969	328.7	7.3	18.5	148.4	154.5	
12	1970	363.5	7.9	18.7	172.1	164.8	
13	1971	396	8.6	20.5	188	178.8	
14	1972	401	9	22.5	180.7	188.8	
15	1973	417.4	9.4	24.5	183.1	200.5	
16	1974	461.1	9.8	26.2	209.3	215.8	
17	1975	487.8	9.6	26.3	220.8	231.1	
18	1976	467.8	8.7	26.6	199.3	233.2	
19	1977	475.9	8.8	29.4	190.7	247	

Note that table 2 is not visible on the page, but has to be expanded.

```
=IMPORTHTML("https://en.wikipedia.org/wiki/Crime_in_the_United_States","list",11)
```

The screenshot shows a Google Sheets spreadsheet with the title "IMPORT Examples". The formula `=IMPORTHTML("https://en.wikipedia.org/wiki/Crime_in_the_United_States","list",11)` is entered in cell A1. The sheet contains the following data:

	A	B	C	D	E	F	G
1	Gangs in the United States						
2	Incarceration in the United States						
3	Mass shootings in the United States						
4	Race and crime in the United States						
5	National Crime Information Center Interstate Identification Index						
6	United States cities by crime rate						
7	List of U.S. states by homicide rate						
8	Strict liability (criminal) § United States						
9	Contempt of court § United States						
10	List of criminal enterprises, gangs and syndicates § United States						
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

The sheet also includes a sidebar with icons for file operations, sharing, and other Google services.

Specifying "list" instead of "table" gets you access to the lists on the page. Here is the "See Also" list.

Scraping data from the web

Now, the Wikipedia page being displayed in your browser is written in HTML (the Hypertext Markup Language). This is a format that allows you to describe “documents.”

HTML is rich in “tags” that format information for display in a browser. There are tags to indicate titles (headers), paragraphs, lists, tables, and links. You can find [a complete list here](#).

At the right, I’ve put a (slightly simplified) version of the table of crime rates on the Wikipedia page. You can see the content in your browser by “right clicking” on the table and choosing “inspect” — it will open the “Developer Tools” pane in Chrome and let you see the HTML behind this element.

```
<table>
  <tbody>
    <tr>
      <th>Metropolitan statistical area</th>
      <th>Violent crime rate</th>
      <th>Property crime rate</th>
    </tr>
    <tr>
      <td>New York-Northern New Jersey-Long Island</td>
      <td>406.0</td>
      <td>1744.1</td>
    </tr>
    <tr>
      <td>Los Angeles-Long Beach-Santa Ana, CA MSA</td>
      <td>405.4</td>
      <td>2232.7</td>
    </tr>
    <tr>
      <td>Chicago-Joliet-Naperville, IL-IN-WI MSA</td>
      <td>357.2</td>
      <td>2791.5</td>
    </tr>
    <tr>
      <td>Dallas-Fort Worth-Arlington, TX MSA</td>
      <td>358.4</td>
      <td>3498.5</td>
    </tr>
    ...
  </tbody>
</table>
```

Scraping data from the web

This should give you a good sense of the difference between data stored (or marked up) for a web page and data in a Google Sheet (well, any spreadsheet).

In general, the act of web scraping takes data in one perhaps unusable format and translates it into something we can compute with.

In the case of lists or tables, this translation is fairly direct (assuming people have entered data in nicely). But often, the elements we want to pull are not so direct, making scraping real work.

```
<ul>
  <li>
    <a href="/wiki/Gangs in the United States" title="Gangs in the United States">Gangs in the United States</a>
  </li>
  <li>
    <a href="/wiki/Incarceration in the United States" title="Incarceration in the United States">Incarceration in the United States</a>
  </li>
  <li>
    <a href="/wiki/Mass shootings in the United States" title="Mass shootings in the United States">Mass shootings in the United States</a>
  </li>
  <li>
    <a href="/wiki/Race and crime in the United States" title="Race and crime in the United States">Race and crime in the United States</a>
  </li>
  ...
  <li>
    <a href="/wiki/Contempt of court#United States" title="Contempt of court §160;United States">Contempt of court §160;United States</a>
  </li>
  <li>
    <a href="/wiki/List of criminal enterprises, gangs and syndicates#United States" title="List of criminal enterprises, gangs and syndicates">List of criminal enterprises, gangs and syndicates §160;United States</a>
  </li>
</ul>
```

Clemency Statistics | PARDON

<https://www.justice.gov/pardon/clemency-statistics#obama>

The screenshot shows the official website of the United States Department of Justice. At the top, the seal of the Department of Justice is visible next to the text "THE UNITED STATES DEPARTMENT OF JUSTICE". A search bar with the placeholder "Search this site" and a magnifying glass icon is positioned on the right. Below the header, a navigation menu includes links for "ABOUT", "OUR AGENCY", "PRIORITIES", "NEWS", "RESOURCES", "CAREERS", and "CONTACT". A breadcrumb trail "Home » Office of the Pardon Attorney" is located on the left, along with a "SHARE" button. The main content area features a large black banner with the text "CLEMENCY STATISTICS" in white. Below the banner, a list of presidents is provided, including William McKinley, Theodore Roosevelt, William H. Taft, Woodrow Wilson, Warren Harding, Calvin Coolidge, Herbert Hoover, Franklin D. Roosevelt, Harry S. Truman, Dwight D. Eisenhower, John F. Kennedy, Lyndon B. Johnson, Richard M. Nixon, Gerald R. Ford, Jimmy Carter, Ronald Reagan, George H.W. Bush, William J. Clinton, George W. Bush, Barack Obama, and Donald J. Trump. Two tables are displayed below this list, showing clemency statistics for William McKinley and Theodore Roosevelt.

William McKinley | Theodore Roosevelt | William H. Taft | Woodrow Wilson | Warren Harding | Calvin Coolidge |
 Herbert Hoover | Franklin D. Roosevelt | Harry S. Truman | Dwight D. Eisenhower | John F. Kennedy | Lyndon B. Johnson | Richard M.
 Nixon | Gerald R. Ford | Jimmy Carter | Ronald Reagan | George H.W. Bush | William J. Clinton | George W. Bush | Barack Obama |
 Donald J. Trump

WILLIAM MCKINLEY								
Fiscal Year	Petitions Pending	Petitions Received	Petitions Granted				Petitions Denied	Closed Without Presidential Action
			Pardon	Commutation	Respite	Remission		
1900	68	677	129	73	4	14	131	332
1901	45	796	162	50	2	12	117	448

THEODORE ROOSEVELT								
Fiscal Year	Petitions Pending	Petitions Received	Petitions Granted			Petitions Denied	Closed Without Presidential Action	
			Pardon	Commutation	Respite			
1901	45	796	162	50	2	12	117	448

Another example

Here are a series of tables with clemency statistics for presidents going back to McKinley.

Commutations Granted by Pres x +

https://www.justice.gov/pardon/obama-commutations



THE UNITED STATES
DEPARTMENT *of* JUSTICE

ABOUT OUR AGENCY PRIORITIES NEWS RESOURCES CAREERS CONTACT

Home » Office of the Pardon Attorney » Clemency Recipients

SHARE

COMMUTATIONS GRANTED BY PRESIDENT BARACK OBAMA (2009-2017)

Search All Pardons and Commutations

January 19, 2017 | January 17, 2017 | December 19, 2016 | November 22, 2016 | November 4, 2016 | October 27, 2016 | October 6, 2016 | August 30, 2016 | August 3, 2016 | June 3, 2016 | May 5, 2016 | March 30, 2016 | January 16, 2016 | December 18, 2015 | July 13, 2015 | March 31, 2015 | December 17, 2014 | December 15, 2014 | April 15, 2014 | December 19, 2013 | November 21, 2011

JANUARY 19, 2017

Download PDF Clemency Warrant

Abdulmuntaqim Ad-Deen

Offense: Possession with intent to distribute cocaine base

District/Date: District of Maryland; October 8, 2008

Sentence: 235 months' imprisonment; five years' supervised release

Terms of grant: Prison sentence commuted to a term of 180 months' imprisonment, conditioned upon enrollment in residential drug treatment

Lesly Alexis

Offense: Conspiracy to possess with intent to distribute more than five kilograms of cocaine powder and more than 50 grams of cocaine base

Another example

Who received a pardon or commutation is also available in a table, but a really awkward one... why?

Scraping data from the web

Here's another example, keeping with the very cheerful crime theme we have going. This is a site called ARMSLIST and was called out in the NYTimes for selling guns across state lines.

Select their “Go Inside, Take a Look” link under the “Classifieds” heading. Then select “New York” state (or any state you want really).

Here's what you get...

The screenshot shows the homepage of Armslist.com. At the top, there is a navigation bar with a search bar, a 'CREATE A LISTING' button, and account options for 'CREATE AN ACCOUNT' and 'LOGIN'. Below the header, there are sections for 'PREMIUM VENDOR SPOTLIGHT' featuring 'Phelan Gun Range' (Member Since 2018, 11 active listings), 'CLASSIFIEDS' (described as 'World's best firearm classifieds'), and 'SAFETY' (with a 'Couch Potato Drill' image). There is also an 'OPINION' section and a large 'OFFICIAL ARMSLIST.com' logo at the bottom right.

ARMSLIST FIREARMS MARKETPLACE

PREMIUM VENDOR SPOTLIGHT

Phelan Gun Range
Member Since 2018
11 active listings - see what we have!

View all vendor storefronts [now!](#)

Get featured [HERE!](#) Click for a storefront and unlimited listings!

CLASSIFIEDS

World's **best** firearm classifieds

Armslist is a **FREE**, simple, and easy to use marketplace.
No fees. No Auctions.

GO INSIDE, TAKE A LOOK!

SAFETY READ MORE

Couch Potato Drill

OPINION READ MORE

Armslist Premium Vendor Program Testimonials

OFFICIAL ARMSLIST.com

Scraping data from the web

You now get a long list of elements that each represent a firearm or ammunition for sale. But if you right click to “Inspect” the elements, you find that the data are not contained in a table.

The screenshot shows a web browser window for the Armslist website. The URL is www.armslist.com/classifieds/search?location=new-york. The page displays 15 of 1065 results in All Categories. The results are listed in a grid format:

- .44 Mag ammo for sale**
\$150
For Sale
Long Island
Sunday, 3/31 6:44 PM
- Ruger PC9 Carbine**
\$500
For Sale
New York
Sunday, 3/31 6:35 PM
- REMINGTON 1100 SKEET 12 ga SHOTGUN VERY GOOD COND**

On the right side of the page, there is a sidebar with a heading "FILTER BY LOCATION" and a map of New York state with the text "Searching in New York". Below the map are dropdown menus for "New York" and "City".

Inspecting a web page

Instead, aspects of each item are divided into different paragraphs, different headings and then a series of nested `<div>` tags (these define divisions or sections in an HTML page).

Here's what you get from inspecting the elements. I've edited it down a little so that the main structure is clearer. Have a look at this and see different tags that have been assembled to present a single item for sale.

This is now where things get interesting. When we don't have a container like `<table>` or `` to tell us where items are on the page, how are we going to find them?

```
<div class="container-fluid">
  <div class="row">
    <div class="col-md-5">
      <a href="44-mag-ammo-for-sale">
        
      </a>
    </div>
    <div class="col-md-7">
      <h4>
        <a href="44-mag-ammo-for-sale">.44 Mag ammo for sale</a>
      </h4>
      <h4> $ 150 </h4>
      <p> <small>For Sale</small> </p>
      <p> <small>Long Island</small> </p>
      <p> <small>Sunday, 3/31 6:44 PM</small> </p>
    </div>
  </div>
</div>
```

Inspecting a web page

The Chrome extension SelectorGadget helps you identify patterns in how a web page is authored to help you identify the information you'd like to extract.

Whereas all we needed with our Google Sheets scraping tool was to know which table or list we wanted, for less structured data, we need to know what tags contain the data we're after.

Let's install the extension and see what it does with the ARMSLIST page. Let's find, say, the prices of the munitions.

[Home](#) > [Extensions](#) > [SelectorGadget](#)



SelectorGadget

Offered by: [selectorgadget.com](#)

★★★★★ 72

| [Developer Tools](#)

|  100,667 users

Inspecting a web page

After you start SelectorGadget, you will see a grey bar at the bottom of your screen. It is going to compile a “description” of where the data you’ve highlighted can be found.

So, if we click on \$449 (the first price in the list of arms for New York State when I ran this) you will get a green line overlaid on the price and then a number of other things will turn yellow.

The bar at the bottom indicates that the price is included in a header tag `<h4>` (which we can verify from the snippet of the web page I included 2 slides back).

But other data are in `<h4>` tags also, including the name of the time for sale, as indicated by its yellow overlay.

ARMSLIST - New York All Cate... +

Not Secure | www.armslist.com/classifieds/search?location=new-york&category=all&page=...

CREATE AN ACCOUNT LOGIN

Search listings

CREATE A LISTING

1 - 15 of 1065 results in All Categories

Set a Search Alert Deals

Get a free SIG P365!

For Sale For Trade Want to Buy Vendors that Ship Premium Vendor Private Party

Rossi 1892 .44-40 Winchester

\$449

For Sale

Hudson Valley

Sunday, 3/31 7:38 PM

.44 Mag ammo for sale

\$150

For Sale

Long Island

Sunday, 3/31 6:44 PM

Ruger PC9 Carbine

\$500

h4

27 Vintage Sports Photos That Are Now Controversial

TieBreaker

FILTER BY

LOCATION

Searching in **New York**

New York

City

Clear (57) Toggle Position XPath ? X

Inspecting a web page

To remove the titles from our selection, we can click on one and it will turn red (see the image on the right). The subsequent names are now no longer yellow as well.

Ah but the bar is now a bit more complex. It reads h4+h4. More on that shortly.

But it looks like the only elements in yellow are the prices we are looking for! This has given us a way to target the data we want.

That is, once we understand what the descriptor coming from SelectorGadget actually means...

The screenshot shows a web browser window for the Armslist website. The URL is www.armslist.com/classifieds/search?location=new-york&category=all&page=1. The page title is "ARMSLIST - New York All Categories". The header includes a "Search listings" bar with a magnifying glass icon, a "CREATE A LISTING" button with a pencil icon, and account links for "CREATE AN ACCOUNT" and "LOGIN". Below the header, there are search filters: "For Sale" (checked), "For Trade" (checked), "Want to Buy" (checked), "Vendors that Ship" (unchecked), "Premium Vendor" (checked), and "Private Party" (checked). The main content area displays 15 of 1065 results. Each result card includes a thumbnail image, the item name, the price (\$449 or \$150), the category (For Sale), the location (Hudson Valley or Long Island), and the posting date (Sunday, 3/31). To the right of the results, there is a sidebar with a headline "27 Vintage Sports Photos That Are Now Controversial" and a photo of a man and a woman. The sidebar also features a "FILTER BY LOCATION" section with a map of New York state labeled "Searching in New York" and dropdown menus for "New York" and "City". At the bottom of the page, there is a search bar with the text "Ruger PC9 Carbine" and a status bar indicating "h4+ h4".

Inspecting a web page

To remove the titles from our selection, we can click on one and it will turn red (see the image on the right). The subsequent names are now no longer yellow as well.

Ah but the bar is now a bit more complex. It reads $h4+h4$. More on that shortly.

But it looks like the only elements in yellow are the prices we are looking for! This has given us a way to target the data we want.

That is, once we understand what the descriptor coming from SelectorGadget actually means...

The screenshot shows a web browser window displaying the Armslist website. The URL in the address bar is www.armslist.com/classifieds/search?location=new-york&category=all&page=1. The page title is "ARMSLIST - New York All Cate". The main content area displays several classified ads:

- Offer** \$300 For Sale Albany Sunday, 3/31 5:53 PM: Multiple rifles for trade, willing to trade 2 or 3 for 1.
- Offer** \$0 For Sale/Trade Buffalo Sunday, 3/31 5:49 PM: Primary Arms 2.5x Prism Scope NEW
- Offer** \$0 For Sale/Trade New York Sunday, 3/31 5:08 PM: Primary Arms 2.5x Prism Scope NEW

On the right side of the page, there is a sidebar with an American flag graphic and the text "JOIN US IN THE FIGHT TO PROTECT AND PRESERVE THE SECOND AMENDMENT". Below this is a "LATEST VIDEO" section featuring a thumbnail of a man speaking at a podium, with a YouTube logo.

At the bottom of the page, there is a search bar with the placeholder "Search listings" and a magnifying glass icon. To the right of the search bar is a "CREATE A LISTING" button with a pencil icon. Further to the right are links for "CREATE AN ACCOUNT" and "LOGIN". At the very bottom, there is a toolbar with various icons and buttons, including "h4+ h4", "Clear (15)", "Toggle Position", "XPath", "?", and "X".

h4+ h4

Clear (15)

Toggle Position

XPath

?

X

Inspecting a web page

So what do we have? SelectorGadget has 15 matches (the number of items for sale on this page) and the pattern it has identified is `h4+h4`.

If you look at the snippet of HTML 5 slides back, you see that the price is in an `<h4>` tag that is followed by an `<h4>` tag containing the listing's name.

The expression `h4+h4` is called a “CSS selector” and this particular selector identifies the content in “an `h4` tag that follows an `h4` tag.”

OK so, what is CSS? And what is a CSS selector?

CSS

CSS stands for “Cascading Style Sheets” and is a language that describes the “style” of an HTML document. With it, we can say that the background of the body of our document should be light blue, that `<h1>` headers should be white, with the text centered, our the text in a `<p>` paragraph should be rendered in Verdana with size 20 pixels.

CSS selectors are then used to find elements to apply a style to. The examples on the right are just single tags (the body, a header, a paragraph), but things can get arbitrarily more complex.

You can [read about CSS and selectors here](#).
And there's [an interesting illustration of the different CSS constructions here](#).

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

Click a selector:

```
#Lastname  
.intro, #Lastname  
h1  
h1, p  
div p  
div > p  
ul + p  
ul ~ table  
  
*[id]  
[id=my-Address]  
[id$=ess]  
[id|=my]  
[id^=L]  
[title~=beautiful]  
[id*=s]  
:checked  
:disabled  
:enabled  
:empty  
:focus  
p:first-child  
p::first-letter  
p::first-line  
p:first-of-type  
h1:hover  
input:in-range  
input:out-of-range  
input:invalid  
input:valid  
p:lang(it)  
p:last-child  
p:last-of-type  
tr:nth-child(even)
```

Selector:

ul + p

The `<p>` element that are next to each `` elements.

Result:

```
<h1> Welcome to My Homepage </h1>  
  
<div class="intro">  
  <p>My name is Donald <span id="Lastname">Duck.</span> </p>  
  
  <p id="my-Address">I live in Duckburg</p>  
  
  <p>I have many friends:</p>  
  </div>  
  
<ul id="Listfriends">  
  • <li>Goofy</li>  
  • <li>Mickey</li>  
  • <li>Daisy</li>  
  • <li>Pluto</li>  
</ul>  
  
<p>All my friends are great!<br>But I really like Daisy!!</p>  
  
<p lang="it" title="Hello beautiful">Ciao bella</p>  
  
<h3> We are all animals! </h3>  
  
<p> <b>My latest discoveries have led me to believe that we are all animals:</b> </p>  
  
<table>  
  <thead>  
    <tr>  
      <th>Name</th>  
      <th>Type of Animal</th>  
    </tr>  
  </thead>
```

CSS

As is the case with anything on the web, there are far more entertaining ways to learn about CSS selectors.

The [CSS Diner](#) is a fun (well, that depends on your definition of “fun” I guess) game that will teach you pretty painlessly what each of the selector expressions does.

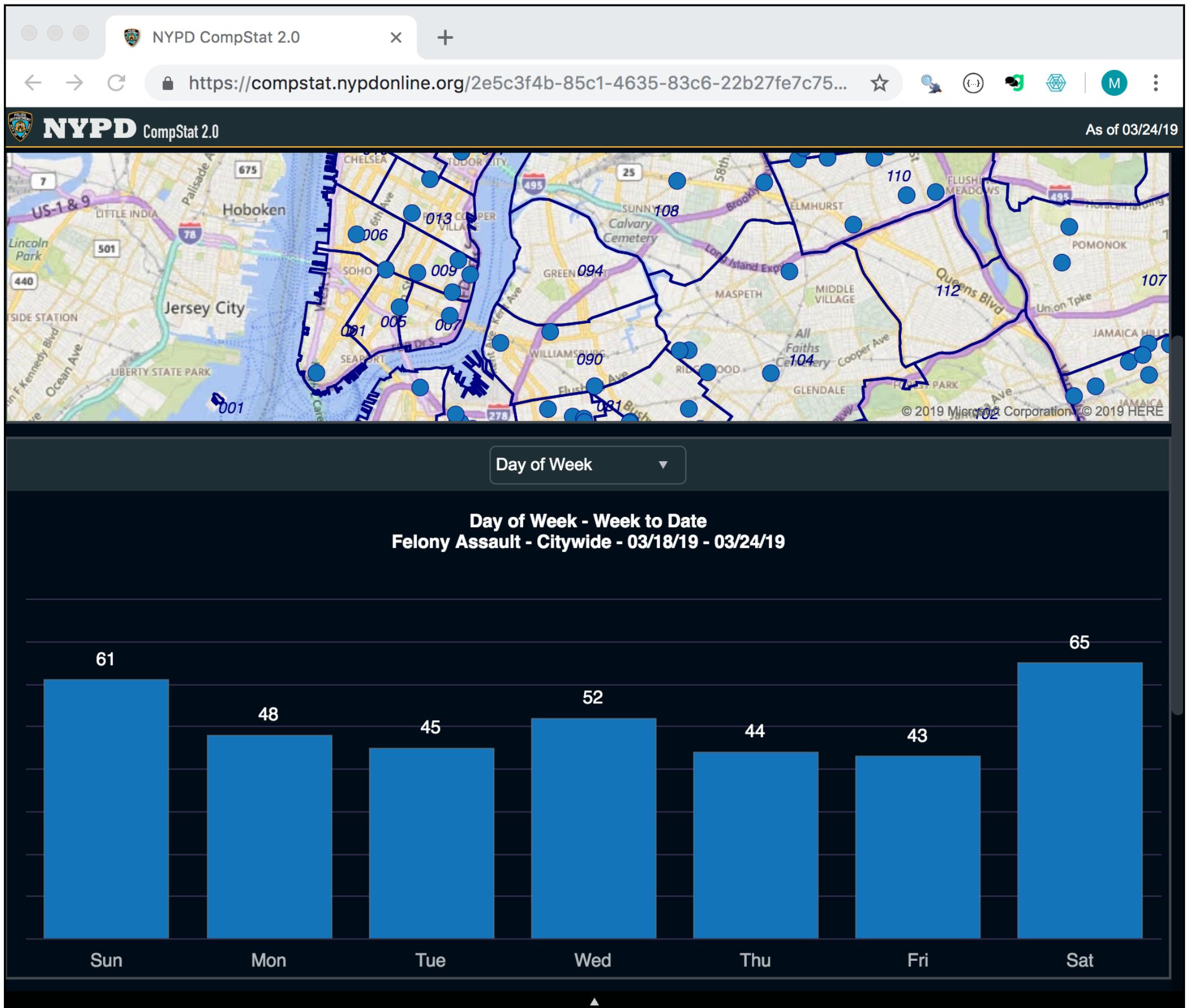
Again, the “selector” in CSS selector is meant to identify content to apply a style to (color, linewidth, size). **Notice that we are using them to find the data we want to extract!**

The screenshot shows a web browser window for "CSS Diner - Where we feast on" at <https://flukeout.github.io>. The title bar says "Level 5 of 32 ✓". The main content area displays the text "Select the pickle on the fancy plate" above an illustration of a yellow plate with a red bento box, a blue and white circle, and two green pickles. To the right, a sidebar shows the CSS selector "#id A" and the text "You can combine any selector with the descendant selector." Below this is an "Examples" section with the CSS rule "#cool span" and its description. At the bottom, a "CSS Editor" window is open with tabs for "style.css" and "HTML Viewer". The "style.css" tab contains the following code:

```
1 Type in a CSS selector enter
2 {
3 /* Styles would go here. */
4 }
5
6 /*
7 Type a number to skip to a level.
8 Ex → "5" for level 5
9 */
10
11
12
13
14
15
16
17
18
19
20
```

The "HTML Viewer" tab shows the following HTML code:

```
1 <div class="table">
2   <bento>
3     <orange />
4   </bento>
5   <plate id="fancy">
6     <pickle />
7   </plate>
8   <plate>
9     <pickle />
10    </plate>
11  </div>
12
13
14
15
16
17
18
19
20
```



The Developer Tools

The developer tools are pretty powerful on their own. You can, for example, watch the “network” activity as CompStat 2.0 calls back to its server for data as you make selections.

Whenever you see an interface like this, you should wonder where the data are coming from and if you can grab them. It's always much easier than literally scraping.

The screenshot shows the NYPD CompStat 2.0 dashboard. At the top, there are two dropdown menus: "Citywide" and "Felony Assault". Below them is a table comparing "Week to Date" data for 2019 (358) and 2018 (371), with a "% Change" of -3.5%. The main area features an "Incident Map" of New York City and surrounding areas, with numerous blue dots representing crime locations. The developer tools are open at the bottom, specifically the Network tab. The Network tab shows a timeline of requests from 1000 ms to 12000 ms. A table below lists network requests, all of which are "list" requests with a status of 200, type "xhr", initiator "libraries.345a73cc.js:15", size between 3.4 KB and 3.7 KB, and time between 54 ms and 70 ms.

Name	Status	Type	Initiator	Size	Time	Waterfall
list	200	xhr	libraries.345a73cc.js:15	3.4 KB	70 ms	
list	200	xhr	libraries.345a73cc.js:15	3.7 KB	60 ms	
list	200	xhr	libraries.345a73cc.js:15	1.1 KB	54 ms	
list	200	xhr	libraries.345a73cc.js:15	3.7 KB	69 ms	

51 requests | 536 KB transferred | 5.8 MB resources | Finish: 12.55 s

The Developer Tools

The developer tools are pretty powerful on their own. You can, for example, watch the “network” activity as CompStat 2.0 calls back to its server for data as you make selections.

Whenever you see an interface like this, you should wonder where the data are coming from and if you can grab them. It’s always much easier than literally scraping.

NYPD CompStat 2.0

<https://compstat.nypdonline.org/2e5c3f4b-85c1-4635-83c6-22b27fe7c75c/view/90>

NYPD CompStat 2.0 As of 03/24/19

Citywide ▾ Felony Assault ▾

Period	2019	2018	% Change
Week to Date ▾	358	371	-3.5 %

Incident Map

Elements Console Sources Network Performance Memory Application Security Audits Web Scraper

View: Group by frame Preserve log Disable cache Offline Online

Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms 7000 ms 8000 ms 9000 ms 10000 ms 11000 ms 12000 ms

Name	Headers	Preview	Response	Cookies	Timing
10020101121.jpg?g=782&amk=ch%20encoding=nn	▶ 0: {Value: "40.8866028,-73.8462919", Metric: 1,...}				
d_vbiawPdxB.js?version=44	▶ 1: {Value: "40.8687497,-73.8949697", Metric: 1,...}				
ping?client_id=1130699450279609&domain=compstat.n...	▶ 2: {Value: "40.8629252,-73.9277586", Metric: 1,...}				
list	▶ 3: {Value: "40.8559898,-73.9006097", Metric: 1,...}				
list	▶ 4: {Value: "40.8031134,-73.9582274", Metric: 1,...}				
	▶ 5: {Value: "40.8134700,-73.9414001", Metric: 1,...}				
	▶ 6: {Value: "40.8270501,-73.8710397", Metric: 1,...}				
	▶ 7: {Value: "40.8221602,-73.8885500", Metric: 1,...}				

51 requests | 536 KB transferred | 5.8 MB resources | Finish:...

The Developer Tools

The developer tools are pretty powerful on their own. You can, for example, watch the “network” activity as CompStat 2.0 calls back to its server for data as you make selections.

Whenever you see an interface like this, you should wonder where the data are coming from and if you can grab them. It's always much easier than literally scraping.