

Lecture 22: Oct 29, 2018

# Web Scraping

- *HTML*
- *Scraping Information*
- *Advanced Scraping*
- *Resources*

James Balamuta  
STAT 385 @ UIUC



# Announcements

- **hw07** is due **Friday, Nov 2nd, 2018 at 6:00 PM**
- **Group project** member choice phase **has begun!**
  - <https://github.com/stat385-fa2018/disc/issues/75>
- **EC Opportunity:** Big Data Summit, Nov 8th
  - <https://github.com/stat385-fa2018/disc/issues/71>
- **Quiz 10** covers Week 9 contents @ [CBTF](#).
  - Window: Oct 30th - Nov 1st
  - Sign up: <https://cbtf.engr.illinois.edu/sched>
- Want to review your homework or quiz grades?  
**Schedule an appointment.**

# Last Time

- **HTTP(S)**
  - Communicating over the internet.
- **Web + APIs**
  - APIs provide structured interaction with software.
  - Web APIs are more stable than web pages
- **httr with REST**
  - Connect with a REST API using HTTP Verbs
  - Formatting of HTTP Response is: Status, Header, Body
- **JSON**
  - Semi-structured output based on key-value form
  - Common Web API response that is converted to an *R* list

# Lecture Objectives

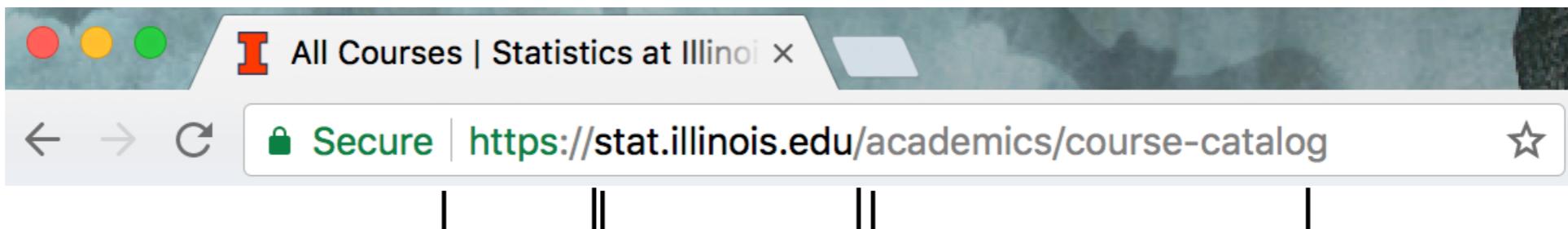
- Understand the **semi-structured form of HTML**
- **Extract** data from a Web Page
- **Describe** the difference between a static and dynamic web page.

# HTML

Previously

# HTTP(S) Request

## Hypertext Transfer Protocol (**S**ecure)



Protocol/Scheme	Domain	Path
Communication medium	Web Address	Location of Resource
for request		



### All Courses

[Collapse All Course Levels](#) [Expand All Course Levels](#)

► 100 Level Courses

### ACADEMICS

Graduate Programs

Undergraduate Program

Career Services

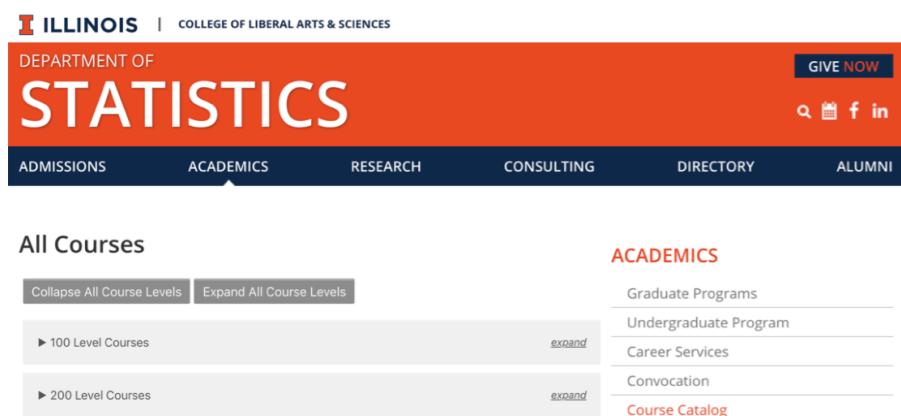
Convocation

Course Catalog

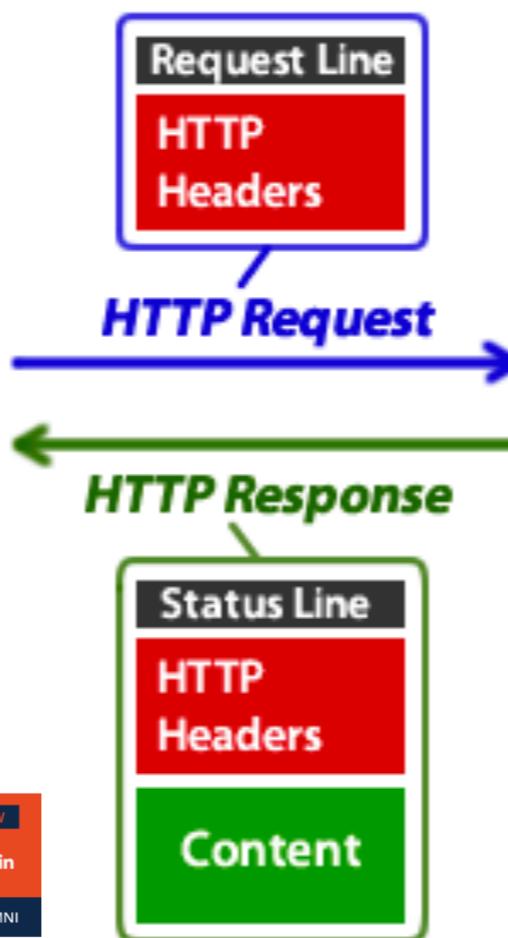
► 200 Level Courses

Previously

Hello [stat.illinois.edu](http://stat.illinois.edu),  
I'd like to see the  
**course catalog**



Hello [stat.illinois.edu](http://stat.illinois.edu),  
thank you!



Hello person's web  
browser, let me pull  
that up...

[Source](#)

Name
▼ academics
careers-statistics.htm
convocation.htm
<b>course-catalog.htm</b>
grad-programs.htm
registration.htm
statistics-seminars.htm
statistics-tutoring.htm
student-groups.htm
undergrad-programs.htm
▶ admissions
▶ alumni
▶ assets
▶ consulting
▶ directory
▶ research

Hello person's web  
browser, I found it.  
Here you go!

Previously

# Structures of Data

... how data is shaped ...

## Structured\*

Rectangular

~5 - 10%


## Semistructured\*\*

key: value

~5 - 10%

---

title: "Untitled"

author: "JJB"

date: "1/27/2018"

output: html\_document

---

## Unstructured\*\*\*

?????????

~80 - 90%

Pinky said,  
"Gee, Brain. What are we  
going to do tonight?"  
The Brain replied, "The  
same thing we do every  
night, Pinky. Try to take  
over the world."

\* Typical form for scientific experiments and company databases

\*\* RMarkdown Document Properties (YAML), JavaScript Object Notation (JSON), XML

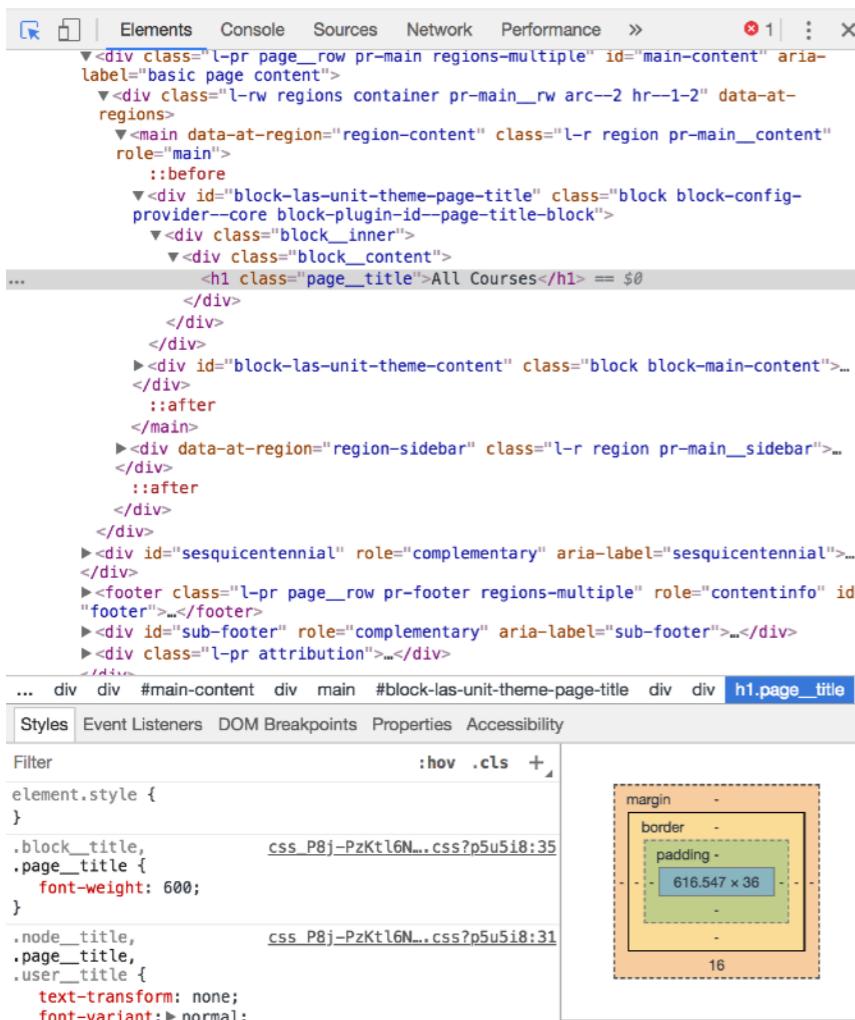
\*\*\* Pure text documents, images, social media posts, and so on. No visible relationship.

# HTML

## Hyper Text Markup Language

### *the language of the web*

## HTML Source



```
<div class="l-pr page__row pr-main regions-multiple" id="main-content" aria-label="basic page content">
  <div class="l-rw regions container pr-main__rw arc--2 hr--1-2" data-at-regions>
    <main data-at-region="region-content" class="l-r region pr-main__content" role="main">
      &before
      <div id="block-las-unit-theme-page-title" class="block block-config-provider--core block-plugin-id--page-title-block">
        <div class="block__inner">
          <div class="block__content">
            <h1 class="page__title">All Courses</h1> == $0
          </div>
        </div>
      </div>
    <div id="block-las-unit-theme-content" class="block block-main-content">...
      &after
    </main>
    <div data-at-region="region-sidebar" class="l-r region pr-main__sidebar">...
      &before
      &after
    </div>
    <div id="sesquicentennial" role="complementary" aria-label="sesquicentennial">...
    </div>
    <footer class="l-pr page__row pr-footer regions-multiple" role="contentinfo" id="footer">...
      <div id="sub-footer" role="complementary" aria-label="sub-footer">...
        <div class="l-pr attribution">...
    </div>
  </div>
  ... div div #main-content div main #block-las-unit-theme-page-title div div h1.page__title

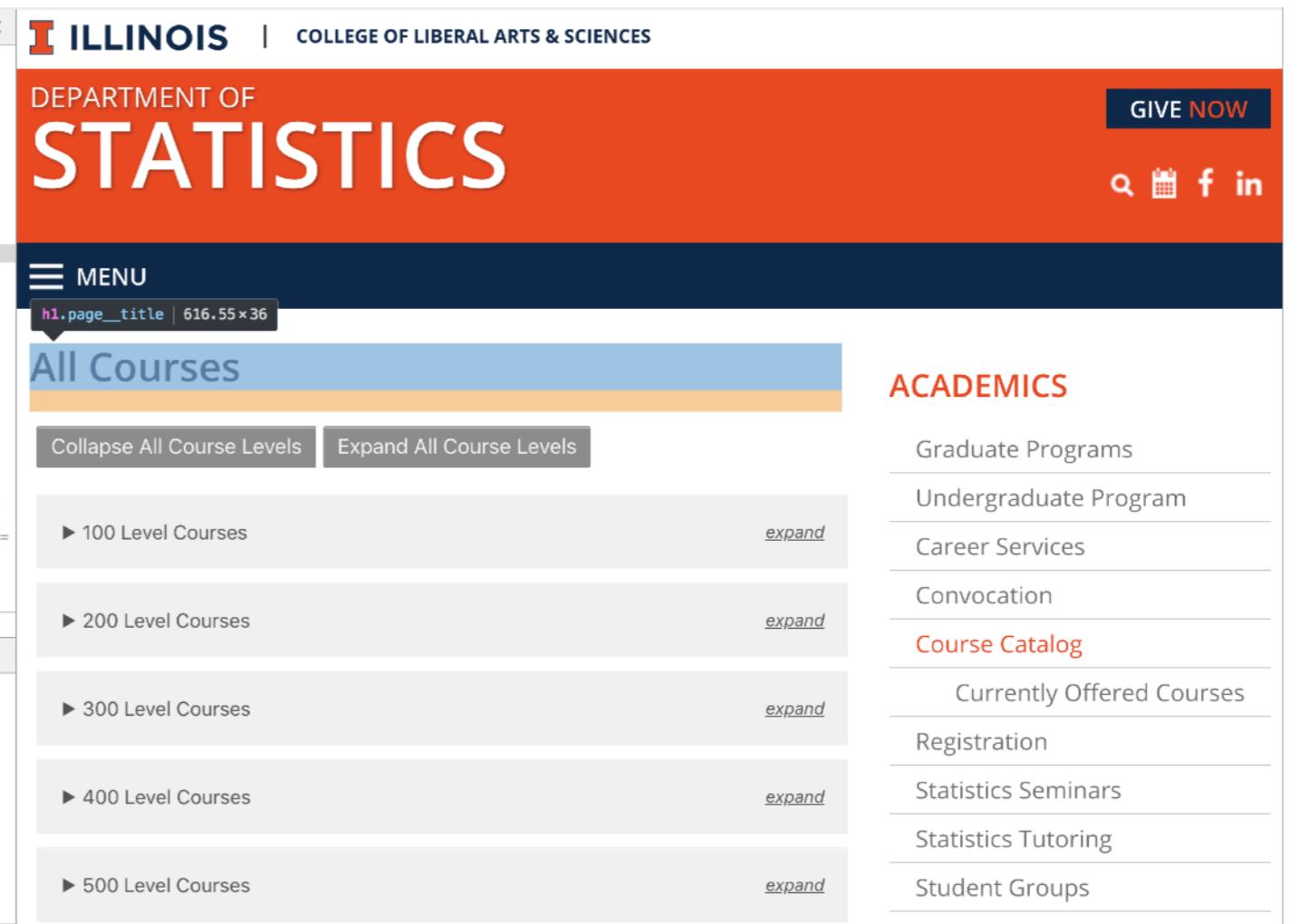
```

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

```
element.style {
}
.block_title, .page_title {
  font-weight: 600;
}
.node_title, .page_title, .user_title {
  text-transform: none;
  font-variant: normal;
}
```

margin - border - padding - 616.547 x 36 - - - 16



ILLINOIS | COLLEGE OF LIBERAL ARTS & SCIENCES

GIVE NOW

DEPARTMENT OF STATISTICS

MENU

All Courses

h1.page\_\_title | 616.55x36

Collapse All Course Levels Expand All Course Levels

- ▶ 100 Level Courses [expand](#)
- ▶ 200 Level Courses [expand](#)
- ▶ 300 Level Courses [expand](#)
- ▶ 400 Level Courses [expand](#)
- ▶ 500 Level Courses [expand](#)

ACADEMICS

Graduate Programs

Undergraduate Program

Career Services

Convocation

**Course Catalog**

Currently Offered Courses

Registration

Statistics Seminars

Statistics Tutoring

Student Groups

<https://stat.illinois.edu/academics/course-catalog>

## Browser View

# HTML Elements

... breakdown of keys and value location ...

## Generic Element

Element Name	Attribute Name	Attribute Value	Content
Individual HTML element found in the web page	Mapping to set specific behaviors	Value assigned to the attribute	Displayed on the website
<code>&lt;element attribute = "property"&gt; content &lt;/element&gt;</code>			

## Example Element for URIs

Element Name	Attribute Name	Attribute Value	Content
Individual HTML element found in the web page	Mapping to set specific behaviors	Value assigned to the attribute	Displayed on the website
<code>&lt;a href = "http://google.com"&gt; Link to Google &lt;/a&gt;</code>			

# Semi-structure of HTML

## ... **key:value** pairing ...

**Key**                    **Value**

HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Title of Page</title>
</head>
<body>
<h1 align = "center"> First order heading (large) </h1>
<p> Paragraph for text with a
    <a href = "http://www.stat.illinois.edu">link!</a>
    <h2> Top Beverages </h2>
    <ol>
        <li> Tea </li>
        <li> Coffee </li>
        <li> Milkshakes </li>
    </ol>
</p>
<table border = "1">
    <tr>
        <th> Name </th>
        <th> Salary </th>
    </tr>
    <tr>
        <td> Joshua Tree </td>
        <td> 66,666 </td>
    </tr>
    <tr>
        <td> Aaron Thomas </td>
        <td> 78,921.40 </td>
    </tr>
</table>
<!-- Comment -->
</body>
</html>
```

### Rendered in Web Browser

## First order heading (large)

Paragraph for text with a [link!](http://www.stat.illinois.edu)

## Top Beverages

1. Tea
2. Coffee
3. Milkshakes

Name	Salary
Joshua Tree	66,666
Aaron Thomas	78,921.40

# HTML DOM

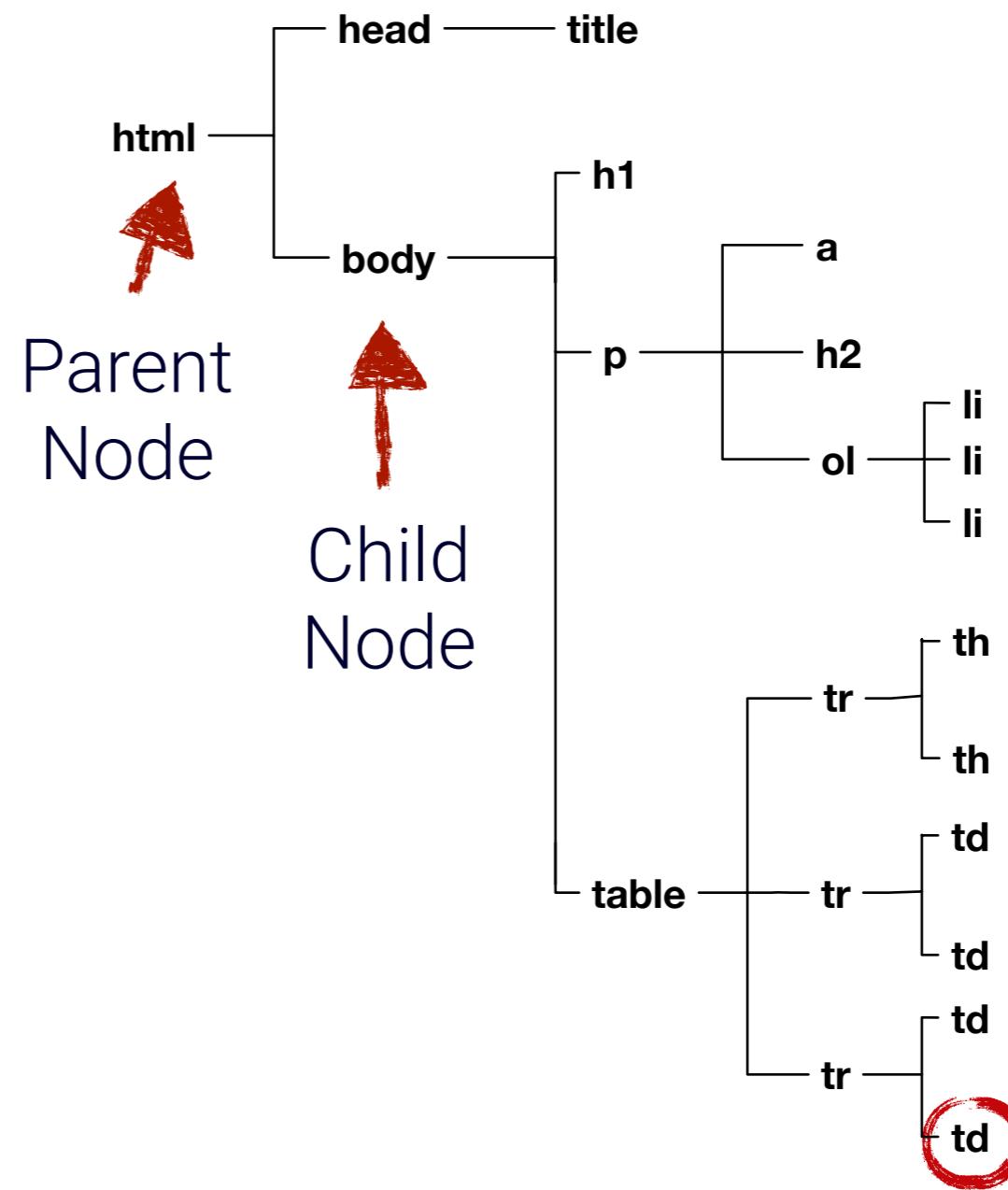
... Document Object Model Trees ...

```
<!DOCTYPE html>
<html>
<head>
<title>Title of Page</title>
</head>
<body>
<h1 align = "center"> First order heading (large) </h1>
<p> Paragraph for text with a
<a href = "http://www.stat.illinois.edu">link!</a>
<h2> Top Beverages </h2>
<ol>
<li> Tea </li>
<li> Coffee </li>
<li> Milkshakes </li>
</ol>
</p>
<table border = "1">
<tr>
<th> Name </th>
<th> Salary </th>
</tr>
<tr>
<td> Joshua Tree </td>
<td> 66,666 </td>
</tr>
<tr>
<td> Aaron Thomas </td>
<td> 78,921.40 </td>
</tr>
</table>
<!-- Comment -->
</body>
</html>
```

Key

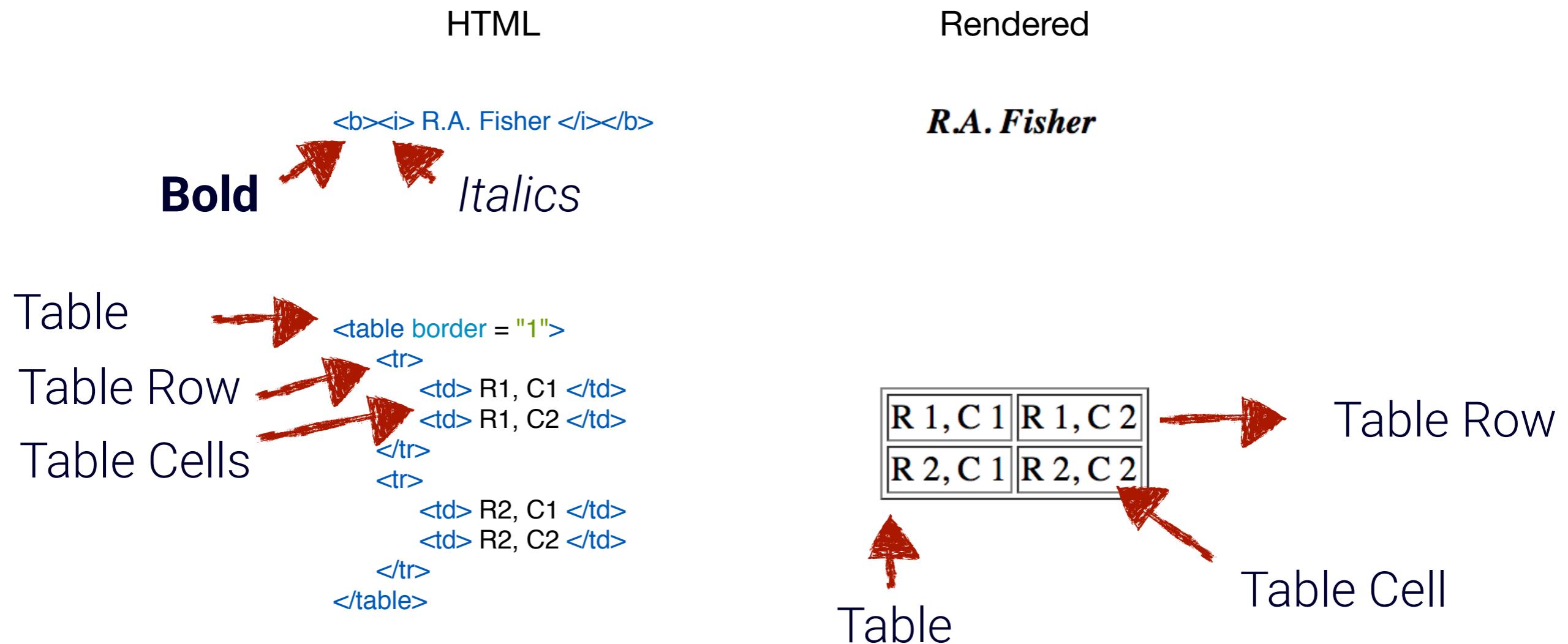
Value

DOM Tree of Web Page



# Common HTML Tags

... breakdown of keys and value location ...



# Your Turn

1. Identify all elements in the web page source.
2. What attributes are present?
3. What are the attribute properties?
4. What information can be extracted?

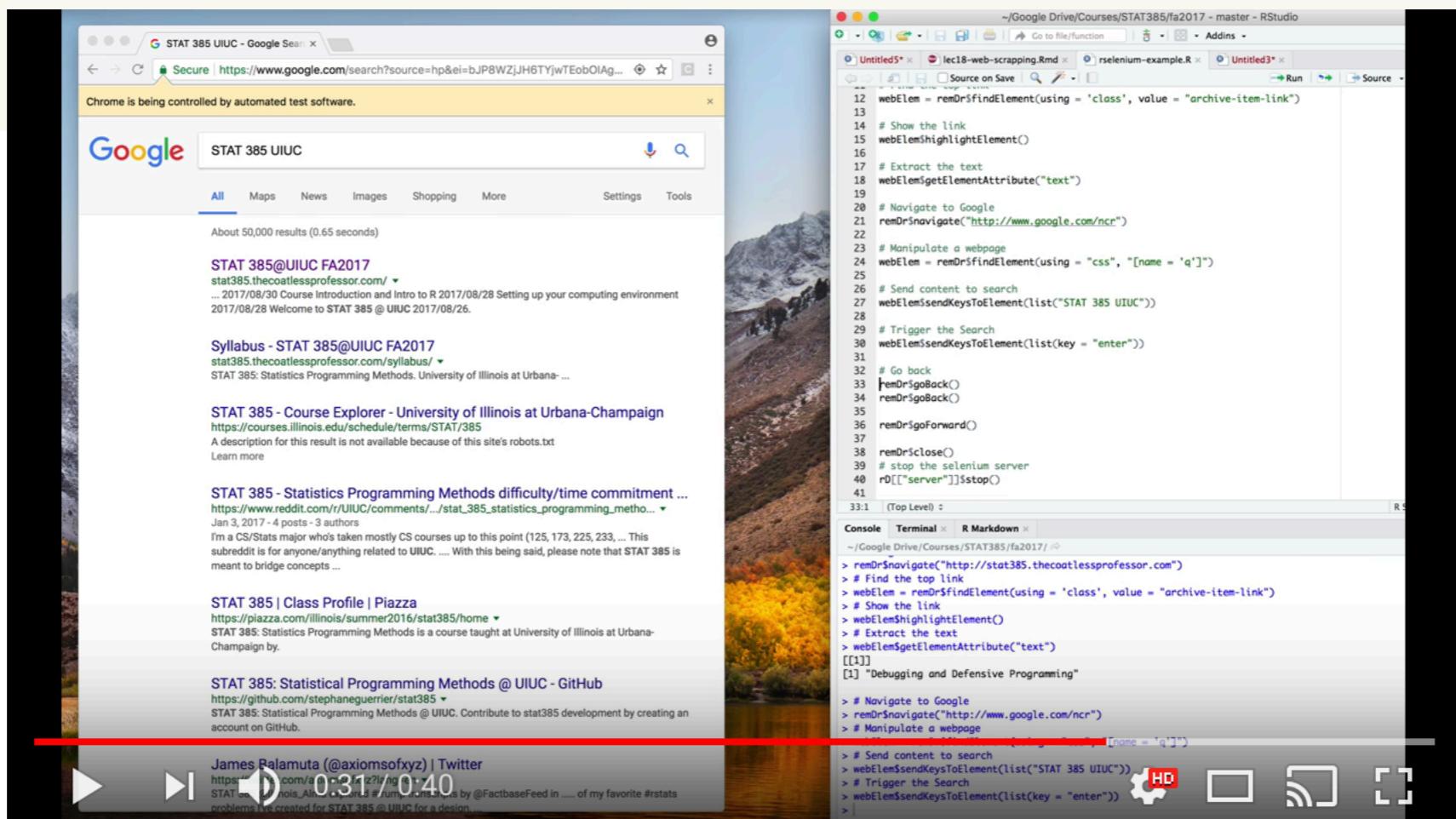
```
<!DOCTYPE html>
<html>
<head>
<title>House of Statistics</title>
</head>
<body bgcolor = "blue">
<h3 align = "left"> My Personal Website </h3>
<hr />
See my work on <a href = "http://www.github.com">GitHub!</a>
<!-- We should probably put a side bar in here... -->
<p> Check out the hypnotoad! </p>
<img src = "https://i.imgur.com/oV8yyJa.gif" />
<h2 align = "right"> Important Dates </h2>
<p>
    <i> December 27th </i> - An Unlucky Person's Birthday
</p>
<h4> Grocery List </h4>
<ul>
    <li> Yogurt </li>
    <li> Greengage Plums </li>
    <li> Saltine Crackers </li>
</ul>
</body>
</html>
```

# Scraping Information

Previously

## Definition:

Web scraping (web harvesting or web data extraction) is a computer software technique of extracting information from websites. [Source](#)



<https://www.youtube.com/watch?v=Vt6f8A35-1w>

# Breakdown of Web Scraping

... flow of the web scraping process ...

1.

**Obtain the identifier for the data to be extracted ...**

... either the html element information or css selector ...

2.

**Download and read into R the webpage ...**

... use rvest's **read\_html()** to read the web page into *R* ...

3.

**Extract the nodes ...**

... picks out the elements on the page with the identifier ...

4.

**Retrieve the content from nodes ...**

... obtain text, table, or attribute information via **html\_\***() ...



... extracting content from static web pages ...

```
install.packages("rvest")  
library("rvest")
```

Function	Description
read_html(x)	Read in HTML Files
html_nodes(x, css, xpath)	Extract Nodes Matching Tag, e.g. <b>&lt;tag&gt;&lt;/tag&gt;</b>
html_node(x, css, xpath)	Extract only one node matching the tag
html_table(x, header, trim, fill, dec)	Convert HTML tables to data.frame, e.g. <b>&lt;table&gt;&lt;/table&gt;</b>
html_text(x, trim)	Extract content between tags
html_name(x)	Extract the name of HTML tag, e.g. <b>&lt;tag&gt;&lt;/tag&gt;</b> gives "tag"
html_attrs(x)	Extract all attributes on a tag, e.g. attribute = "value"
html_attr(x, name, default)	Extract values associate with specific attribute.

# Web Page as a String

... direct embedding ...

## First order heading (large)

Paragraph for text with a [link!](#)

### Top Beverages

1. Tea
2. Coffee
3. Milkshakes

Name	Salary
Joshua Tree	66,666
Aaron Thomas	78,921.40

```
#install.packages("rvest")
library("rvest")

sample_webpage = '<!DOCTYPE html>
<html>
<head>
<title>Title of Page</title>
</head>
<body>
<h1 align = "center"> First order heading (large) </h1>
<p> Paragraph for text with a
    <a href = "http://www.stat.illinois.edu">link!</a>
    <h2> Top Beverages </h2>
    <ol>
        <li> Tea </li>
        <li> Coffee </li>
        <li> Milkshakes </li>
    </ol>
</p>
<table border = "1">
    <tr>
        <th> Name </th>
        <th> Salary </th>
    </tr>
    <tr>
        <td> Joshua Tree </td>
        <td> 66,666 </td>
    </tr>
    <tr>
        <td> Aaron Thomas </td>
        <td> 78,921.40 </td>
    </tr>
</table>
<!-- Comment --&gt;
&lt;/body&gt;
&lt;/html&gt;'</pre>
```

# Read HTML

... 3 ways to read into *R* a website ...

## # Option 1: Read Web Page as a String from R

```
my_webpage = read_html(sample_webpage)
```

## # Option 2: Read an active Web Page via its URI

```
my_webpage = read_html("http://domain.com/path/to/sample_webpage.html")
```

## # Option 3: Read a Web Page Saved on the Local File System

```
my_webpage = read_html("~/Documents/sample_webpage.html")
```

```
my_webpage  
# {xml_document}  
# <html>  
# [1] <head>\n<meta http-equiv="Content-Type" content="text/#html; charset=UTF-8">\n<title>Title ...  
# [2] <body>\n<h1 align="center"> First order heading (large) </h1>\n<p> Paragraph for text with ...
```

# Previously

## Definition:

Piping is the act of taking one value and immediately placing it into another function to form a flow of results.

### Left Function

Transmitting function result  
`rnorm(10)`

### Pipe Operator

Facilitate moving left result  
to the function on right

### Right Function

Receiving function result in  
first parameter  
`abs(rnorm(10))`

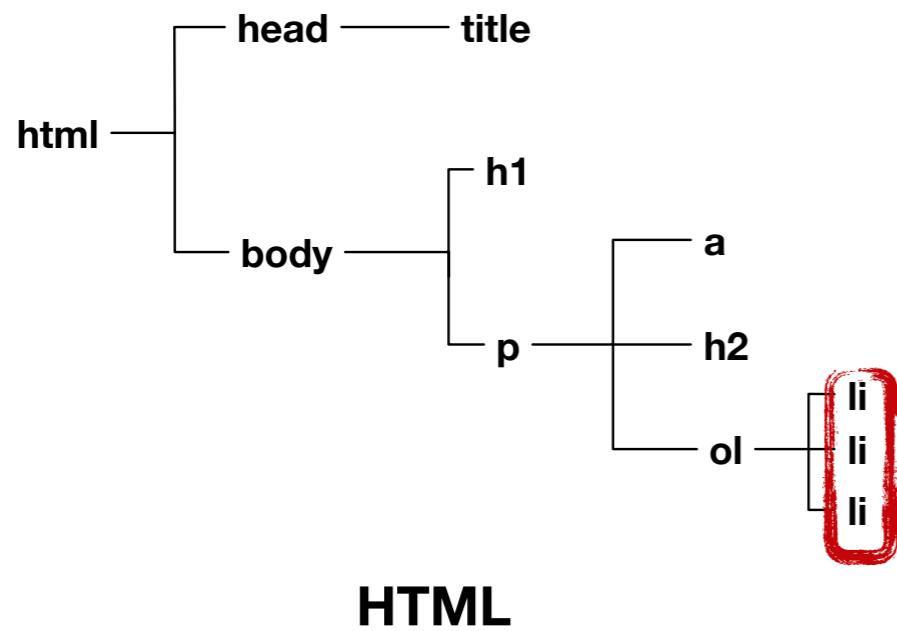
`rnorm(10) %>% abs()`



`%>%` is read as "and, then"

# Extract Nodes

... selecting HTML elements / keys ...



```
<!DOCTYPE html>
<html>
<head>
<title>Title of Page</title>
</head>
<body>
<h1 align = "center"> First order heading (large) </h1>
<p> Paragraph for text with a
    <a href = "http://www.stat.illinois.edu">link!</a>
    <h2> Top Beverages </h2>
    <ol>
        <li> Tea </li>
        <li> Coffee </li>
        <li> Milkshakes </li>
    </ol>
</p>
<!-- Comment -->
</body>
</html>
```

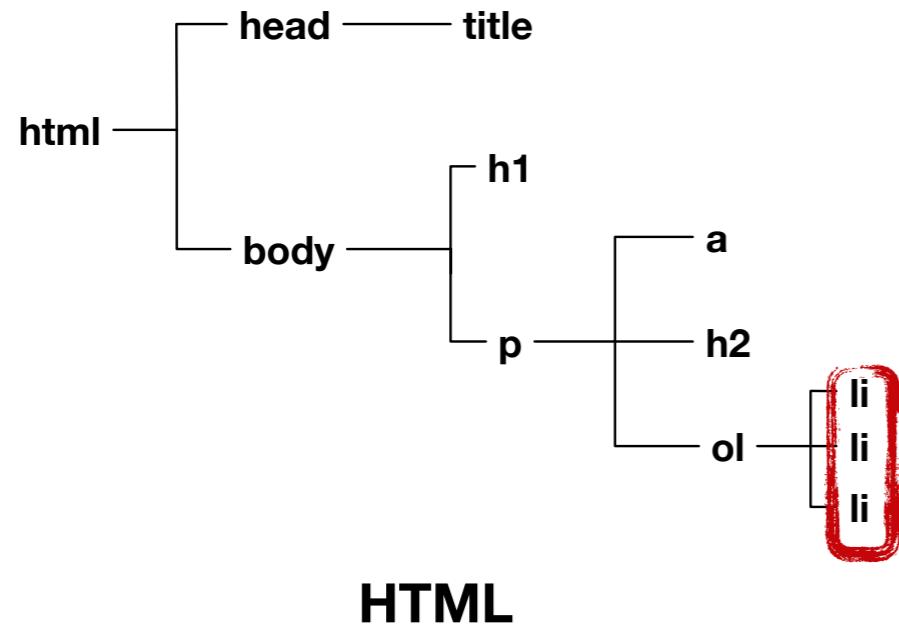
```
# Retrieve only the first
# instances of the li element
# on the page.
my_webpage %>%
    html_node("li")
# {xml_nodeset (1)}
# [1] <li> Tea </li>
```

```
# Retrieve all instances
# of the li element on page.
my_webpage %>%
    html_nodes("li")
# {xml_nodeset (3)}
# [1] <li> Tea </li>
# [2] <li> Coffee </li>
# [3] <li> Milkshakes </li>
```

# Which should you prefer?

# Retrieving Content

... obtaining values as text ...

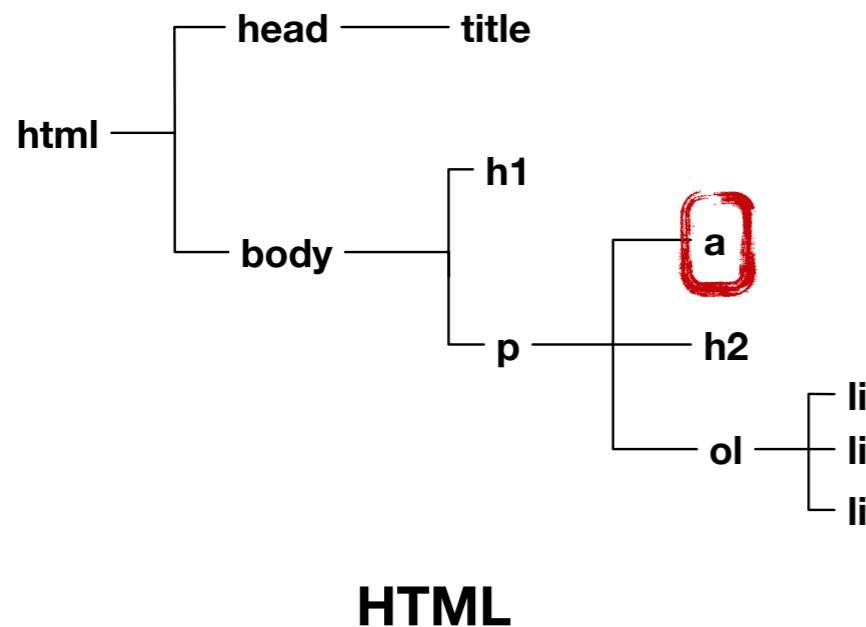


```
<!DOCTYPE html>
<html>
<head>
<title>Title of Page</title>
</head>
<body>
<h1 align = "center"> First order heading (large) </h1>
<p> Paragraph for text with a
<a href = "http://www.stat.illinois.edu">link!</a>
<h2> Top Beverages </h2>
<ol>
<li> Tea </li>
<li> Coffee </li>
<li> Milkshakes </li>
</ol>
</p>
<!-- Comment -->
</body>
</html>
```

```
# Retrieve the text content
# in between all instances
# of the li element.
my_webpage %>%
  html_nodes("li") %>%
  html_text()
# [1] "Tea"
# [2] "Coffee"
# [3] "Milkshakes"
```

# Retrieving Attribute Content

... obtaining the text values ...



```
<!DOCTYPE html>
<html>
<head>
<title>Title of Page</title>
</head>
<body>
<h1 align = "center"> First order heading (large) </h1>
<p> Paragraph for text with a
<a href = "http://www.stat.illinois.edu">link!</a>
<h2> Top Beverages </h2>
<ol>
<li> Tea </li>
<li> Coffee </li>
<li> Milkshakes </li>
</ol>
</p>
<!-- Comment -->
</body>
</html>
```

```
# Extract all a elements and, then...
```

```
# Retrieve only the attribute
# value for href.
```

```
my_webpage %>%
  html_nodes("a") %>%
  html_attr("href")
```

```
# [1]
```

```
# "http://www.stat.illinois.edu"
```

```
# Retrieve all the attributes
# on all a elements.
```

```
my_webpage %>%
  html_nodes("a") %>%
  html_attrs()
```

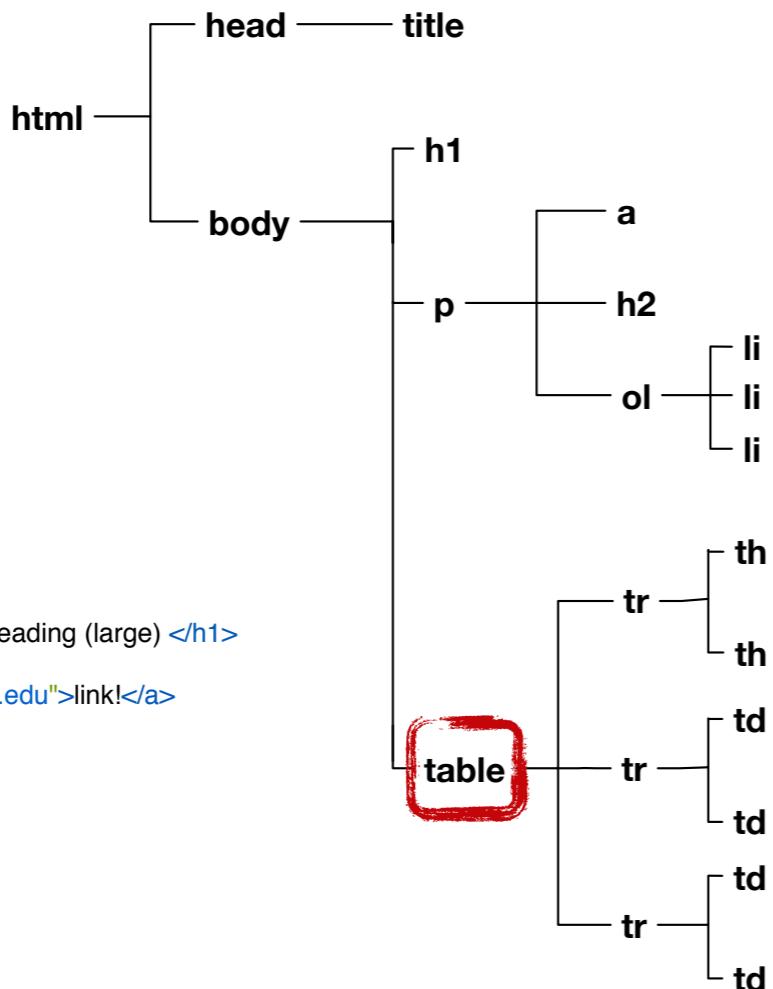
```
# [[1]]
```

```
# href
```

```
# "http://www.stat.illinois.edu"
```

# Retrieving Table Content

... obtaining the cell information ...



```
# Extract the table element.
# Retrieve the contents as
# a data.frame.
my_webpage %>%
  html_node("table") %>%
  html_table()
```

```
# Make sure the node is
# a table before trying to
# use html_table()!
```

# Live Web Page

... web scraping [illinoiselectiondata.com](http://illinoiselectiondata.com) ...

We want to obtain all tables

The screenshot shows a live web page from [illinoiselectiondata.com](http://illinoiselectiondata.com). The top navigation bar includes links for About, Elections, Maps, Analysis, Budgets, Redistricting, FAQ, Subscribers, Login, and a user icon. The main content area is titled "Illinois Profile - General Elections" and features a section titled "Vote Share - Where the Vote Comes From". Below this is a detailed description of how vote location varies over time and shifts between presidential and non-presidential election years. Two tables are presented: "General Election - Vote Share by Region - Traditional Collar Counties" and "General Election - Vote Share by Region - Expanded Collar Counties". Both tables compare 2014 Governor, 2012 President, and 2010 US Senate election results across four regions: Chicago, Cook County Suburbs, Collar Counties (5), and Downstate (96). A red arrow points from the text "We want to obtain all tables" towards the top of the first table.

**General Election - Vote Share by Region - Traditional Collar Counties**

Region	2014 Governor	2012 President	2010 US Senate
Chicago	18.10%	19.37%	18.59%
Cook County Suburbs	18.97%	18.99%	19.12%
Collar Counties (5)	24.89%	24.45%	24.28%
Downstate (96)	38.03%	37.18%	38.00%
<b>TOTAL:</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>

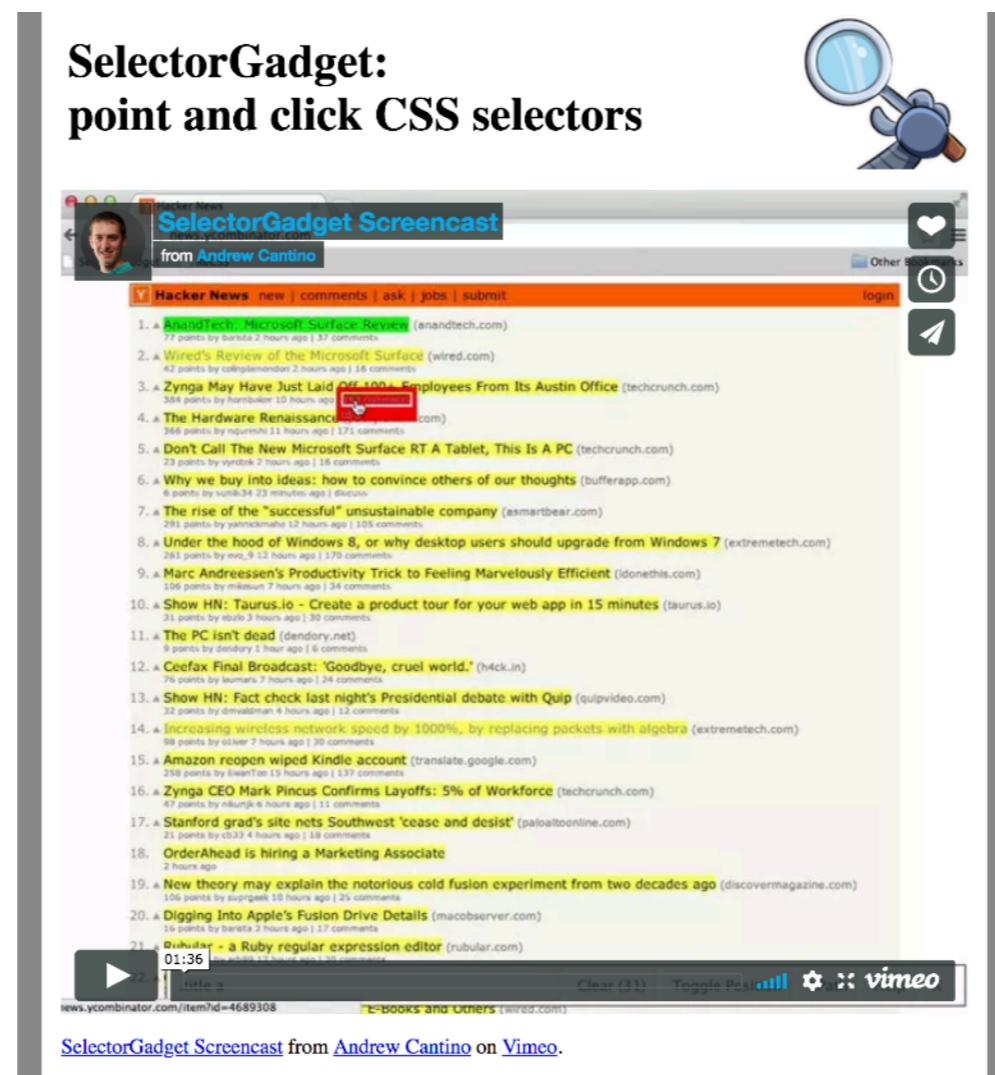
**General Election - Vote Share by Region - Expanded Collar Counties**

Region	2014 Governor	2012 President	2010 US Senate
Chicago	18.10%	19.37%	18.59%
Cook County Suburbs	18.97%	18.99%	19.12%
Collar Counties (11)	29.19%	28.55%	28.39%
Downstate (90)	33.73%	33.07%	33.89%
<b>TOTAL:</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>

How can we  
retrieve elements uniquely?

# SelectorGadget

... obtaining unique identifiers for page elements ...



<http://selectorgadget.com/>  
rvest: SelectorGadget Vignette

# Install SelectorGadget

## ... Bookmark vs. Browser ...

Slashdot Apple W Wikipedia Outlook Web App SelectorGadget LON-CAPA

Created by [Andrew Cantino](#) and [Kyle Maxwell](#). You can find the [current version on GitHub](#), and please feel free to leave comments below.

Try our new [Chrome Extension!](#)



### Hold the click and drag

Or drag this link to your bookmark bar: [SelectorGadget \(updated August 7, 2013\)](#)

Or use the development version: [SelectorGadget Unstable \(updated August 7, 2013\)](#)

Bookmark link on page  
(contains JavaScript)

 SelectorGadget offered by [selectorgadget.com](#)

★★★★★ (67) | [Developer Tools](#) | 74,143 users

[OVERVIEW](#) [REVIEWS](#) [SUPPORT](#) [RELATED](#)

Hacker News [100] | threads | comments | ask | jobs | submit

tectonic (2931) | Inquit

1. [\[REDACTED\]](#) (100) | 1 hour ago | flag | 12 comments

2. [Android AOSP maintainer quits](#) ([plus.google.com](#)) (72 points) by [wizid](#) 2 hours ago | flag | 44 comments

3. [Steps To \\$5,000 In Monthly Recurring Revenue](#) ([statuspage.io](#)) (54 points) by [joshua](#) 2 hours ago | flag | 13 comments

4. [Hacker News founder Rob Malda on why there won't be another Hacker News](#) ([washingtonpost.com](#)) (54 points) by [liberateur](#) 2 hours ago | flag | 130 comments

5. [Thoughts on Twitter's new Two-Factor Authentication](#) ([wuth.com](#)) (74 points) by [endre](#) 3 hours ago | flag | 26 comments

6. [Starving at the Sun: Dalvik vs. Asm.js vs. Native](#) ([mozilla.org](#)) (130 points) by [bengard](#) 3 hours ago | flag | 130 comments

7. [MaxComputeWIKI \(YC W12\) Needs a Graphic/Game Designer](#) ([techcrunch.com](#)) (130 points) by [bengard](#) 3 hours ago | flag | 26 comments

8. [EFFP "Parallel construction" is really intelligence laundering](#) ([eff.org](#)) (99 points) by [bengard](#) 3 hours ago | flag | 72 comments

9. [Programming languages to watch: LiveScript, Julia, Elixir](#) ([adambard.com](#)) (143 points) by [phipps](#) 6 hours ago | flag | 79 comments

10. [Envelopes - Python email for humans](#) ([homeworkjck.github.io](#)) (130 points) by [homeworkjck](#) 6 hours ago | flag | 11 comments

11. [App.net Response to Brennan Novak, part II](#) ([zapp.net](#)) (130 points) by [aravind](#) 1 hour ago | flag | 12 comments

12. [Pixel-perfect timing attacks with HTML5](#) ([contentful.co.uk](#)) (149 points) by [hettner](#) 7 hours ago | flag | 72 comments

13. [Restoring Trust in Government and the Internet](#) ([schneier.com](#)) (149 points) by [hettner](#) 7 hours ago | flag | 72 comments

14. [Jupyter Notebook for Julia](#) ([github.com](#)) (52 points) by [bengard](#) 8 hours ago | flag | 5 comments

15. [Dulia: IPython Notebook for Julia](#) ([github.com](#)) (52 points) by [bengard](#) 8 hours ago | flag | 5 comments

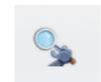
+ ADD TO CHROME 

Compatible with your device

Easy, powerful CSS Selector generation.

Selector Gadget is an open source Chrome Extension that makes CSS selector generation and discovery on complicated sites a breeze.

After having installed the extension, go to any page and launch it. A box will open in the bottom right of the website. Click on a page element that you would like your selector to match (it will turn green). SelectorGadget will then generate a minimal CSS selector for that element, and will highlight (outline in green) any element that is matched.



### Chrome Extension

# Example Selectors

... possible ways to select HTML ...

Selector	Example	Description
.class	.greentext	Find all elements with class="greentext"
#id	#newsblock	Find elements with id="newsblock"
tag	p	Find all <p> elements
tag tag	p span	Find all <span> and <p> elements
tag > tag	p > span	Find all <span> elements with <p> as a parent
[attribute]	[href]	Find all elements that have a href attribute
[attribute=value]	[href="http://google.com/"]	Find all elements with href="https://google.com/"

# SelectorGadget in Action

... web scraping [illinoiselectiondata.com](http://illinoiselectiondata.com) ... \*

ILELECTIONDATA About Elections Maps Analysis Budgets Redistricting FAQ Subscribers Login ↑

## Illinois Profile - General Elections

### Vote Share - Where the Vote Comes From

The location of the vote varies from year to year and has evolved over time as the population has shifted. Further, the noted difference in turnout between Presidential election years and non-presidential election years has an effect on the location of the vote. Below is a table that shows the vote share by region for the top ballot races in the last two cycles. The regions are defined as City of Chicago, the Cook County suburbs outside of Chicago, the five traditional collar counties (Lake, McHenry, DuPage, Kane & Will) and the downstate 96 counties.

#### General Election - Vote Share by Region - Traditional Collar Counties

Region	2014 Governor	2012 President	2010 US Senate
Chicago	18.10%	19.37%	18.59%
Cook County Suburbs	18.97%	18.99%	19.12%
Collar Counties (5)	24.89%	24.45%	24.28%
Downstate (96)	38.03%	37.18%	38.00%
<b>TOTAL:</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>

[Vote Share by Region - Traditional Collars](#)

#### General Election - Vote Share by Region - Expanded Collar Counties

Region	2014 Governor	2012 President	2010 US Senate
Chicago	18.10%	19.37%	18.59%
Cook County Suburbs	18.97%	18.99%	19.12%
Collar Counties (11)	29.19%	28.55%	28.39%
Downstate (90)	33.73%	33.07%	33.89%
<b>TOTAL:</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>

[Vote Share by Region - Expanded Collars](#)

Selected

Remove element selection

Also included, click to deselect

.myTable-blue| Clear (10) Toggle Position XPath Help X

Generated Element Selector

# Using the Selector

... Retrieving Tables of Voting Information...

```
# Read in the Web Page to R  
# without saving into the server.  
il_profile = read_html("https://illinoiselectiondata.com/elections/ILprofile.php")  
  
# Select the elements that are tables.  
# Extract the contents as a data.frame.  
il_profile %>%  
  html_nodes(".myTable-blue") %>%  
  html_table()
```

# New Selections

\*  
... web scraping [news.google.com](https://news.google.com) ...

The screenshot shows the Google News homepage. At the top, there's a navigation bar with 'Google News', a search bar, and a 'Sign in' button. Below the navigation, there are tabs for 'Headlines', 'Local', 'For You', and 'U.S.' with a dropdown arrow. A gear icon for settings is also present.

**Top Stories**

- [Trump's Lawyer Raised Prospect of Pardons for Flynn and Manafort as Special Counsel Closed In](#)  
New York Times · 2h ago
- [White House says 'no discussion' of Trump pardoning Manafort or Gates](#)  
c-wiz a
- [Trump's lawyer discussed idea of him pardoning Flynn, Manafort: report](#)  
The Hill · 2h ago
- [10 legal experts on why Trump can't pardon his way out of the Russia investigation](#)  
In Depth · Vox · 2h ago

**In the News**

- Kim Jong-un
- North Korea
- China
- Amazon.com
- Xi Jinping
- Donald Trump
- Beijing
- Kim Jong-il

At the bottom, there's a search bar with the class selector '.hzdq5d' highlighted with a red oval. Other buttons in the search bar include 'Clear (227)', 'Toggle Position', 'XPath', 'Help', and 'X'. The URL in the address bar is <https://www.politico.com/story/2018/03/28/trump-pardon-manafort-gates-white-house-489838>.

- \* The class selector found for the title links are time sensitive. If you repeat this procedure, you will receive a different class. The only way to change this is to look into the link attributes.

# Time-limited Selectors

... retrieving titles for Google News ...

```
# Read in the Web Page to R without saving into the server.  
gnews = read_html("https://news.google.com")  
  
# Select story titles and retrieve the text.  
gnews %>%  
  html_nodes(".hzdq5d") %>%  
  html_text()  
# [1] "Trump's Lawyer Raised Prospect of Pardons for Flynn and Manafort as Special  
Counsel Closed In"  
# [2] "Trump's lawyer allegedly raised possibility of pardons for Manafort, Flynn last  
summer"  
# [3] "White House says 'no discussion' of Trump pardoning Manafort or Gates"
```

# Your Turn

Find the top listed stars of The Thomas Crown Affair

Extract the financial information from the  
Open Secrets campaign initiative.

<https://www.opensecrets.org/races/election?id=IL>

Hint: You will need to use `html_table(x, fill = TRUE)`

# Advanced Scraping

# Web Developer Tools

... exploring a website's source code ...

Open Chrome's Web Developer Tools



- Windows: **Ctrl + Shift + I**
- macOS: **Command + Option + I**

A screenshot of a Google News page in a web browser. The browser's address bar shows a secure connection to https://news.google.com. The main content area displays headlines about the Helsinki summit, including one from BBC News and another from The New York Times. To the right of the content, the Chrome DevTools are open, specifically the Elements tab. An arrow points from the text above to the DevTools interface. The Elements tab shows the HTML structure of the page, with the body element selected. The HTML code visible includes doctype, lang, head, and body sections with various script and style tags. The bottom of the DevTools window shows tabs for html, body.IF2Cpe.cteFme.EII Dfe.uOat3d, Styles, Event Listeners, DOM Breakpoints, Properties, and Accessibility.

# Example: IMDb

... selecting story headlines more robustly ...

1. Click on the selector
2. Mouse over element on webpage
3. Click to Select

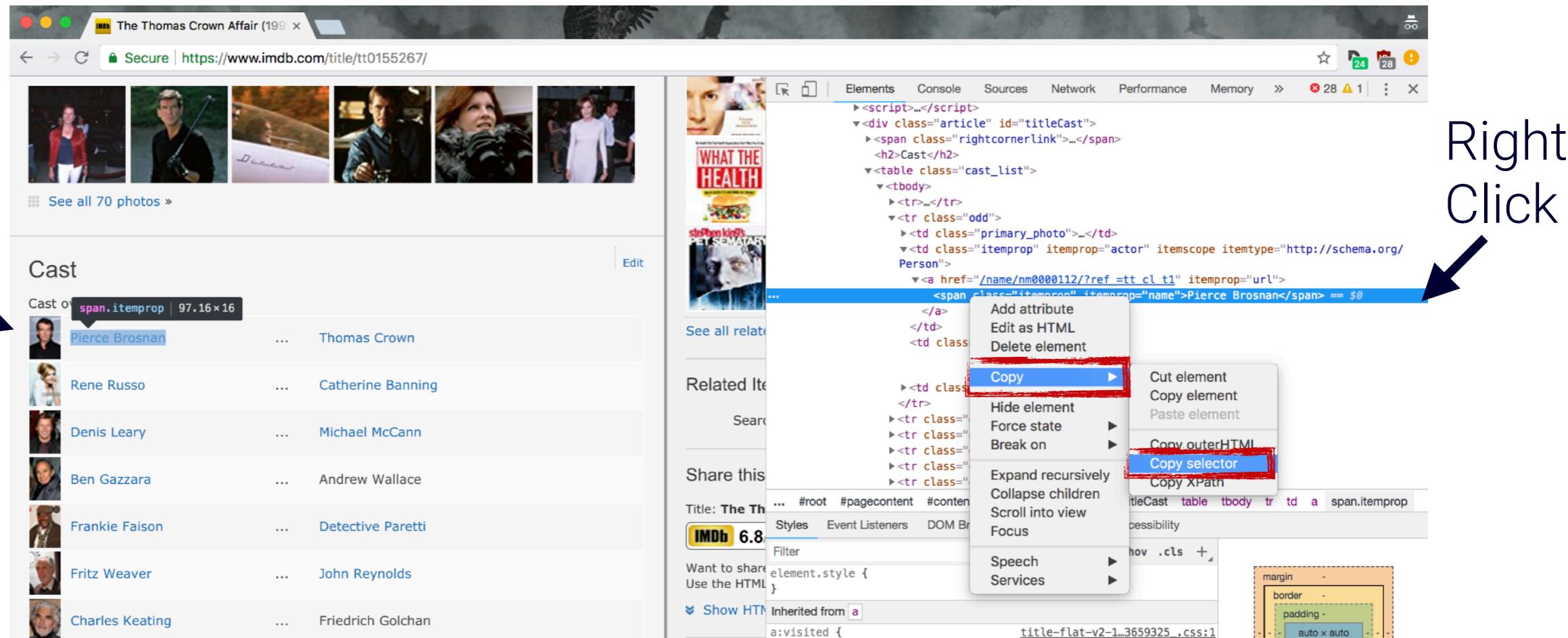
The screenshot shows a web browser window displaying the IMDb page for the movie "The Thomas Crown Affair". The browser's developer tools are open, specifically the "Elements" tab, which is highlighted with a red border. In the Elements tab, a blue arrow points from the text "span.itemprop" to a specific element in the DOM tree. This element is a span tag with the class "itemprop" and the value "Pierce Brosnan". Another blue arrow points from the same text "span.itemprop" to the "IMDb 6.8" logo located at the bottom right of the browser window.

The main content area of the browser shows the movie's cast list. The first item in the list is Pierce Brosnan, who played Thomas Crown. The developer tools also show the full HTML code for the page, with the "scriptsOn" section visible at the bottom.

Cast Member	Character
Pierce Brosnan	Thomas Crown
Rene Russo	Catherine Banning
Denis Leary	Michael McCann
Ben Gazzara	Andrew Wallace
Frankie Faison	Detective Paretti
Fritz Weaver	John Reynolds
Charles Keating	Friedrich Golchan
Mark Margolis	Heinrich Knutzhorn

# Retrieve Selector

... note this is still a unique selector ...



Selected

Right Click

```
#titleCast > table > tbody > tr:nth-child(2) > td.itemprop > a > span
```

How can we  
**generalize to obtain all actors?**

# # Finding Commonalities in Semistructure

```
<td class="itemprop" itemprop="actor" itemscope=""  
itemtype="http://schema.org/Person">  
<a href="/name/nm0000112/?ref_=tt_cl_t1" itemprop="url">  
<span class="itemprop" itemprop="name"</a> </td>
```

The screenshot shows a web browser window displaying the cast list for the movie "The Thomas Crown Affair". The browser's developer tools are open, specifically the Elements tab, which is showing the HTML structure of the page. A context menu is open over a specific `td.itemprop` element, which is highlighted with a red box. The context menu options include "Copy" and "Copy element", also highlighted with a red box. A large blue arrow points from the text "Selected" to the highlighted `td.itemprop` element in the browser's UI. Another blue arrow points from the text "Right Click" to the context menu itself.

Selected

Right Click

```
<td class="itemprop" itemprop="actor" itemscope=""  
itemtype="http://schema.org/Person">  
<a href="/name/nm0000112/?ref_=tt_cl_t1" itemprop="url">  
<span class="itemprop" itemprop="name">Pierce Brosnan</span>  
</a> </td>
```

Secure | https://www.imdb.com/title/tt0155267/

Elements Console Sources Network Performance Memory More 30 28 1

Cast

Pierce Brosnan

Rene Russo

Denis Leary

Ben Gazzara

Frankie Faison

Fritz Weaver

Thomas Crown

Catherine Banning

Michael McCann

Andrew Wallace

Detective Paretti

John Reynolds

WHAT THE HEALTH

IMDB 6.8

Copy

Copy element

Copy outerHTML

Copy selector

Copy XPath

```
# Read in the Movie  
imdb_movie = read_html("https://www.imdb.com/title/tt0155267/")  
  
# Create a CSS selector based on two or more element attributes.  
imdb_movie %>%  
  html_nodes("td[itemprop=\"actor\"] span[itemprop=\"name\"]") %>%  
  html_text()  
# [1] "Pierce Brosnan" "Rene Russo" "Denis Leary"  
# [4] "Ben Gazzara" "Frankie Faison" "Fritz Weaver"  
# [7] "Charles Keating" "Mark Margolis" "Faye Dunaway"  
# [10] "Michael Lombard" "Bill Ambrozy" "Michael Bahr"  
# [13] "Robert D. Novak" "Joe H. Lamb" "James Saito"
```

# Creating a CSS Selector

... ensuring the right values are extracted ...

- 
- \* The use of \" is to escape the quotation character as it is already used.
  - \*\* We opt to not specify a > as there is another element nested.

## Enabled JavaScript (default)

The screenshot shows the ESPN NBA Scoreboard for June 8, 2018. The page displays game scores, player statistics, and video highlights. A green banner at the bottom indicates "JavaScript has been enabled." The browser's developer tools toolbar is visible at the top, showing various feature toggles. The "Disable JavaScript" button is highlighted with a red box.

## Disabled JavaScript

The screenshot shows the same ESPN NBA Scoreboard page, but with JavaScript disabled. Most content is missing or blank. The green banner at the bottom indicates "JavaScript has been disabled." The browser's developer tools toolbar is visible at the top, showing the "Disable JavaScript" button is now checked.

Website: <http://www.espn.com/nba/scoreboard>

# Checking for JavaScript ... hidden content error ...

Install [Web Developer](#)  
Extension for Chrome.

Disable JavaScript, reload  
the page, and watch content  
disappear.

- One solution is to use [phantomjs](#) to render the page locally and then scrape it.
- Another solution is to use [RSelenium](#) to replicate a web browser and scrape data. This isn't very stable though.

# Resources

# HTML Reference

... additional HTML help ...

w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE

HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP MORE ▾ REFERENCES ▾ EXAMPLES ▾ 🔍

HTML5 Tutorial  
HTML HOME  
HTML Introduction  
HTML Editors  
HTML Basic  
HTML Elements  
HTML Attributes  
**HTML Headings**  
HTML Paragraphs  
HTML Styles  
HTML Formatting  
HTML Quotations  
HTML Comments  
HTML Colors  
HTML CSS  
HTML Links  
HTML Images  
HTML Tables  
HTML Lists  
HTML Blocks  
HTML Classes  
HTML Id  
HTML Iframes

**HTML Headings**

◀ Previous      Next ▶

Headings

Heading 1  
Heading 2  
Heading 3  
Heading 4  
Heading 5  
Heading 6

Try it Yourself »

MDN web docs moz://a

Technologies ▾ References & Guides ▾ Feedback ▾

Sign in 🔍

HTML reference

Web technology for developers ▾  
HTML ▾ HTML reference

Related Topics

Cascading Style Sheets (CSS)  
Document Object Model (DOM)  
Guide to Web APIs  
Web API Reference

- ▶ Browser-independent tools
- ▶ Firefox developer tools
- ▶ Safari developer tools
- ▶ Chrome developer tools

HTML element reference  
This page lists all the HTML elements.

HTML attribute reference  
Elements in HTML have attributes; these are additional values that configure the elements or adjust their behavior in various ways to meet the criteria the users want.

Global attributes  
Global attributes may be specified on all HTML elements, even those not specified in the standard. That means that any non-standard elements must still permit these attributes, even though using those elements means that the document is no longer HTML5-compliant. For example, HTML5-compliant browsers hide content marked as `<foo hidden>...</foo>`, even though `<foo>` is not a valid HTML element.

Link types  
In HTML, the following link types indicate the relationship between two documents, in which one links to the other using an `<a>`, `<area>`, or `<link>` element.

View all pages tagged "HTML"...

W3Schools HTML Reference

Mozilla's HTML Reference

# SelectorGadget

... identify elements ...

## SelectorGadget: point and click CSS selectors



The screenshot shows a browser window displaying the Hacker News homepage. The SelectorGadget extension is active, with a magnifying glass icon visible in the top right corner of the page area. The page content includes several news items with titles like "AnandTech: Microsoft Surface Review", "Wired's Review of the Microsoft Surface", and "Zynga May Have Just Laid Off 100+ Employees From Its Austin Office". The extension's interface is visible at the bottom of the browser window, showing controls for play/pause, volume, and other media functions.

[SelectorGadget Screencast](#) from Andrew Cantino on [Vimeo](#).

## Selectorgadget

Hadley Wickham

2016-06-16

Selectorgadget is a javascript bookmarklet that allows you to interactively figure out what css selector you need to extract desired components from a page.

### Installation

To install it, open this page in your browser, and then drag the following link to your bookmark bar: [Selectorgadget](#).

### Use

To use it, open the page

1. Click on the element you want to select. Selectorgadget will make a first guess at what css selector you want. It's likely to be bad since it only has one example to learn from, but it's a start. Elements that match the selector will be highlighted in yellow.
2. Click on elements that shouldn't be selected. They will turn red. Click on elements that *should* be selected. They will turn green.
3. Iterate until only the elements you want are selected. Selectorgadget isn't perfect and sometimes won't be able to find a useful css selector. Sometimes starting from a different element helps.

For example, imagine we want to find the actors listed on an IMDB movie page, e.g. [The Lego Movie](#).

1. Navigate to the page and scroll to the actors list.

The screenshot shows a list of actors with their names and roles. The list includes: Will Arnett (as Batman / Bruce Wayne (voice)), Elizabeth Banks (as Wyldstyle / Lucy (voice)), Craig Berry (as Blake / Additional Voices (voice)), Alison Brie (as Unikitty (voice)), David Burnes (as Oskar Robot / Additional Voices (voice)), Anthony Daniels (as C-3PO (voice)), Charlie Day (as Benny (voice)), Amanda Faris (as Mom (voice)), and Keith Ferguson (as Han Solo (voice)). The names of the actors are highlighted in yellow, indicating they are the target elements for selection.

<http://selectorgadget.com/>

rvest: SelectorGadget Vignette

# Recap

- **HTTP(S)**
  - Way to retrieve websites.
- **HTML**
  - Semi-structured Data
  - Language of the web
- **Pipe Operator**
  - Facilitate the flow of multi-step problems
- **Scraping Information**
  - Extracts values from HTML Tags

This work is licensed under the  
Creative Commons  
Attribution-NonCommercial-  
ShareAlike 4.0 International  
License

