Applications of Data Science

Intro to Web Scraping

Applications of Data Science - Class Bonus

Giora Simchoni

`gsimchoni@gmail.com and add #dsapps in subject`

Stat. and OR Department, TAU

2021-12-31

APPLICATIONS

OF DATA SCIENCE

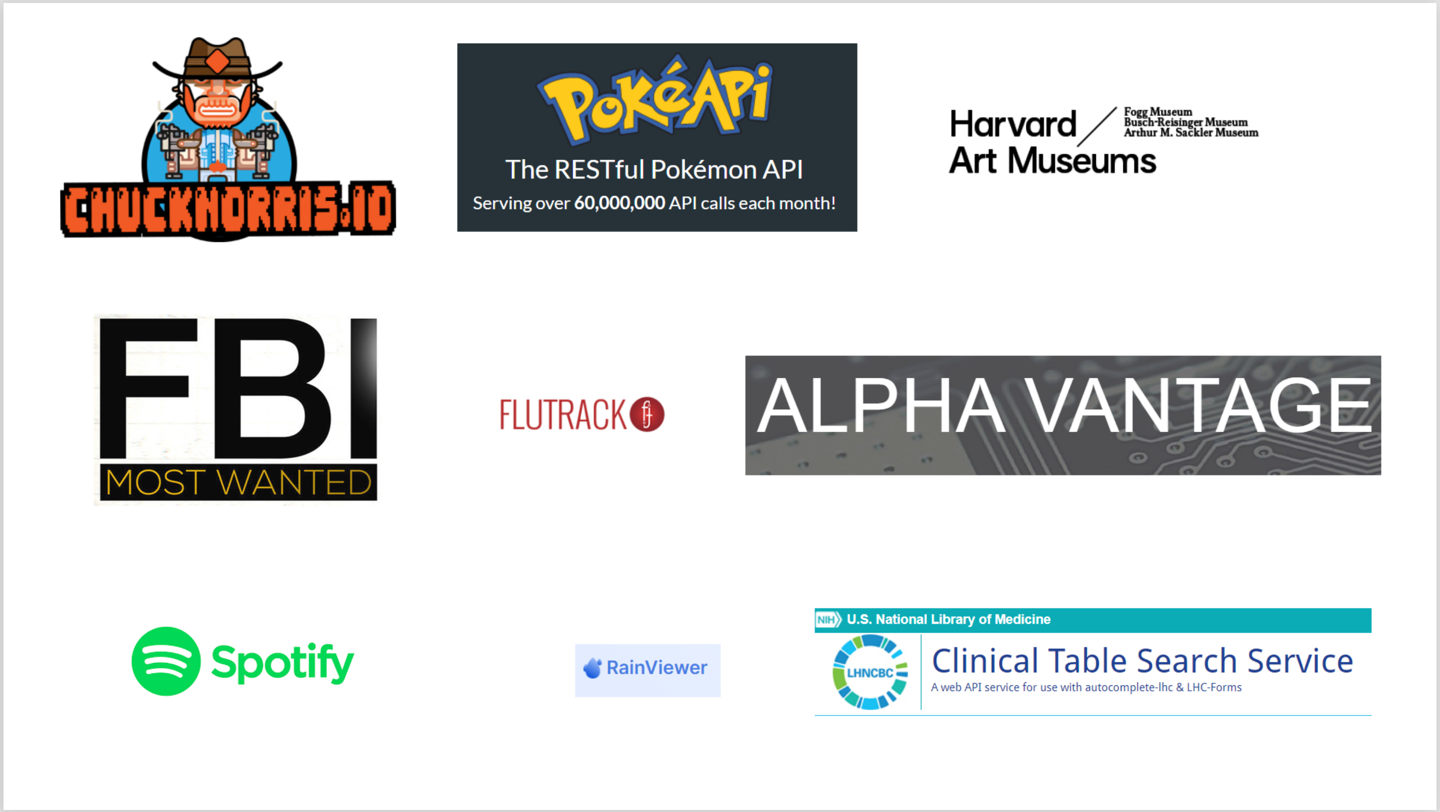
The Three Rules of Web Scraping

Applications

Of Data Science

Rule 1: Do you *really* need web scraping?

There are data APIs for just about anything, you know...

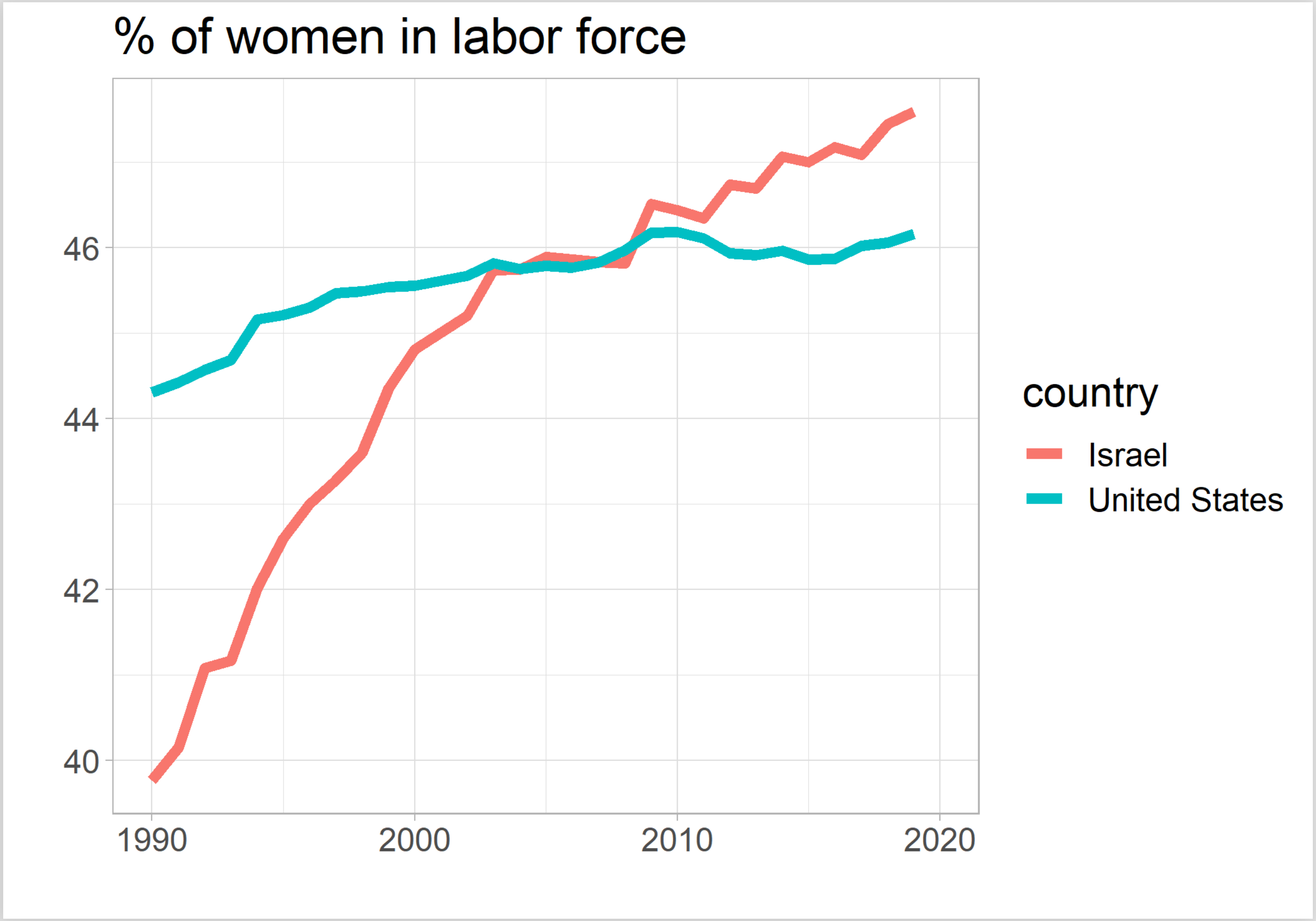


R API Packages

Many of them already accessible with a R/Python package...

```r
library(wbstats)

female_labor <- wb_data(
  indicator = c("women_lab_share" = "SL.TLF.TOTL.FE.ZS"),
  start_date = 1990,
  end_date = 2020
)

female_labor %>%
  filter(country %in% c("Israel", "United States")) %>%
  ggplot(aes(date, women_lab_share, color = country)) +
  geom_line(lwd = 2) +
  labs(title = "Share of women in labor force") +
  theme_light() +
  theme(text = element_text(size=16))
```

From: https://cfss.uchicago.edu/notes/application-program-interface/

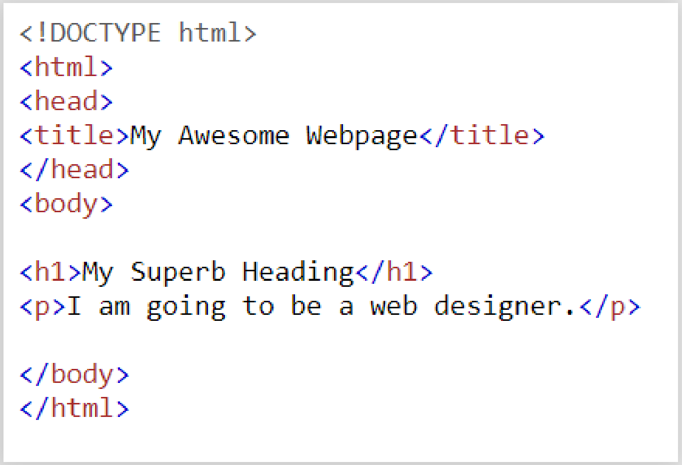

% of women in labor force
46
44
42
40
1990
2000
2010
2020
country
Israel
United States

The `datapasta` package

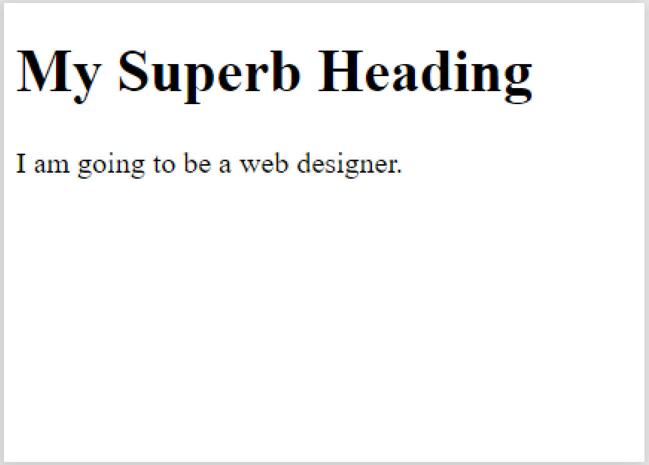
My gift to you.

Rule 2: Learn some HTML first!

HTML is a set (or tree) of *elements*, marked by *HTML tags*:



```
<!DOCTYPE html>
<html>
<head>
<title>My Awesome Webpage</title>
</head>
<body>

<h1>My Superb Heading</h1>
<p>I am going to be a web designer.</p>

</body>
</html>
```



My Superb Heading

I am going to be a web designer.

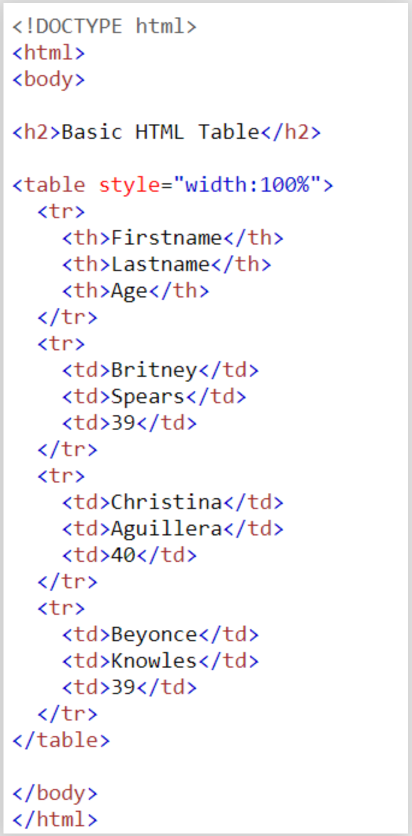
- First children in the tree: `header` and `body`
- View any page's HTML (on chrome) with right-click and "View page source" (or Ctrl + U)

Useful elements and attributes to know

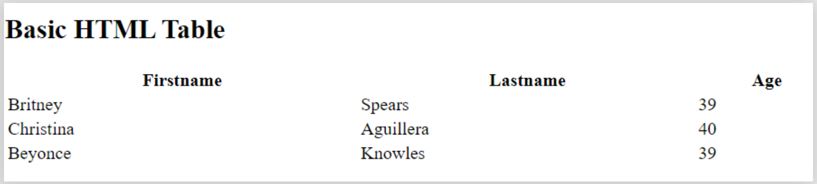
- `<p>` for paragraph `</p>`
- `<h1>` for headings `</h1>`
- `<br>`, `<hr>` for breaks
- `<a href = "http://www.google.com>` for links `</a>`
- `<b><i>` For bold, italic etc. `</i></b>`
- `<img src="img_name.jpg" alt="Alternative text">`
- `<p style="color:DodgerBlue;">` for font color `</p>`

HTML Tables

A big thing when it comes to data as you can imagine...



```
<!DOCTYPE html>
<html>
<body>

<h2>Basic HTML Table</h2>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Britney</td>
    <td>Spears</td>
    <td>39</td>
  </tr>
  <tr>
    <td>Christina</td>
    <td>Aguillera</td>
    <td>40</td>
  </tr>
  <tr>
    <td>Beyonce</td>
    <td>Knowles</td>
    <td>39</td>
  </tr>
</table>

</body>
</html>
```

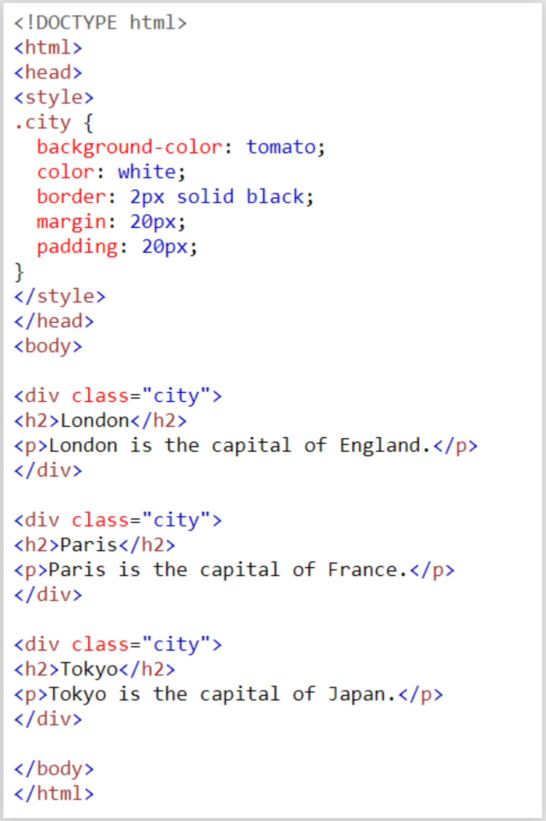


Basic HTML Table

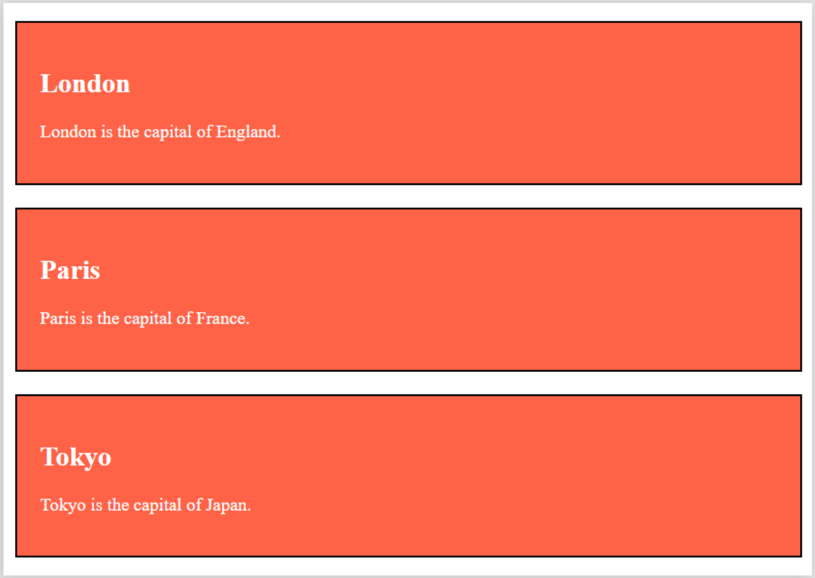
Firstname	Lastname	Age
Britney	Spears	39
Christina	Aguillera	40
Beyonce	Knowles	39

HTML Classes

A class attribute is defined in a style sheet, lets you repeat a style.



```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
<h2>London</h2>
<p>London is the capital of England.</p>
</div>

<div class="city">
<h2>Paris</h2>
<p>Paris is the capital of France.</p>
</div>

<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```



London

London is the capital of England.

Paris

Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

Rule 3: Be polite!

With great power comes great responsibility.

See e.g. the polite package.

rvest

APPLICATIONS OF DATA SCIENCE

`read_html()`

You're now a NLP expert, and you've just developed a SOTA Q&A model. How would you demonstrate your model's performance?

How about triviaquestionsnow.com?

Let's scrape a few Q&As. Politely.

```
library(rvest)

url <- "https://www.triviaquestionsnow.com/for/sports-trivia"

html_obj <- read_html(url)
```

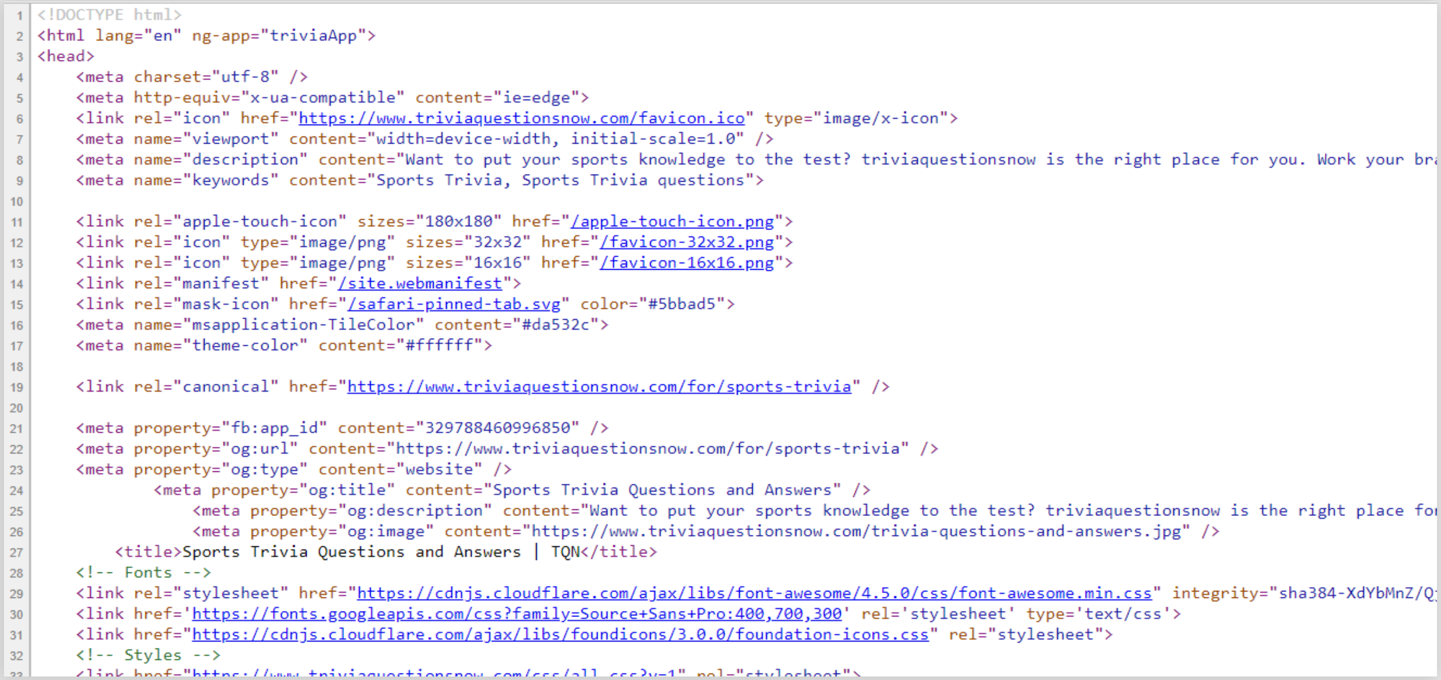
`read_html()` is usually where you'd start. What did you get?

```
class(html_obj)
```

```
## [1] "xml_document" "xml_node"
```

View page source

With time, you'll become friendly with this weird object. Right now it is better than...



```
<!DOCTYPE html>
<html lang="en" ng-app="triviaApp">
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <link rel="icon" href="https://www.triviaquestionsnow.com/favicon.ico" type="image/x-icon">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="description" content="Want to put your sports knowledge to the test? triviaquestionsnow is the right place for you. Work your bra
    <meta name="keywords" content="Sports Trivia, Sports Trivia questions">

    <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
    <link rel="manifest" href="/site.webmanifest">
    <link rel="mask-icon" href="/safari-pinned-tab.svg" color="#5bbad5">
    <meta name="msapplication-TileColor" content="#da532c">
    <meta name="theme-color" content="#ffffff">

    <link rel="canonical" href="https://www.triviaquestionsnow.com/for/sports-trivia" />

    <meta property="fb:app_id" content="329788460996850" />
    <meta property="og:url" content="https://www.triviaquestionsnow.com/for/sports-trivia" />
    <meta property="og:type" content="website" />
            <meta property="og:title" content="Sports Trivia Questions and Answers" />
                <meta property="og:description" content="Want to put your sports knowledge to the test? triviaquestionsnow is the right place fo
                <meta property="og:image" content="https://www.triviaquestionsnow.com/trivia-questions-and-answers.jpg" />
        <title>Sports Trivia Questions and Answers | TQN</title>
    <!-- Fonts -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css" integrity="sha384-XdYbMnZ/Q
    <link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700,300' rel='stylesheet' type='text/css'>
    <link href="https://cdnjs.cloudflare.com/ajax/libs/foundicons/3.0.0/foundation-icons.css" rel="stylesheet">
    <!-- Styles -->
    <link href="https://www.triviaquestionsnow.com/css/all.css?v=1" rel="stylesheet">
```

`html_children()` and `html_node()`

Our tree has two children: `head` and `body`

```
html_obj %>% html_children()
```

```
## {xml_nodeset (2)}
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=
## [2] <body>\n    <div class="title-bar" data-responsive-toggle="nav" data-
```

Again notice the object returned might not be familiar ("`xml_nodeset`")

And each of the children has children of its own:

```
html_obj %>% html_node("body") %>% html_children()
```

```
## {xml_nodeset (6)}
## [1] <div class="title-bar" data-responsive-toggle="nav" data-hide-for="m
## [2] <div id="nav" class="top-bar">\n        <div class="row">\n
## [3] <div class="wrap bg-grey-light t-pad-20 b-pad-20">\n          <div
## [4] <script src="https://www.triviaquestionsnow.com/js/all.js?v=1"></scr
```

html_nodes()

Usually we'd figure out a rule and want a list of all relevant nodes:

```
html_obj %>% html_nodes("img")
```

```
## {xml_nodeset (8)}
## [1] <img src="https://www.triviaquestionsnow.com/img/trivia-questions.pn
## [2] <img src="https://www.triviaquestionsnow.com/img/trivia-questions.pn
## [3] <img src="https://www.triviaquestionsnow.com/img/category/360x130/-c
## [4] <img src="https://www.triviaquestionsnow.com/img/category/360x130/-c
## [5] <img src="https://www.triviaquestionsnow.com/img/category/360x130/-c
## [6] <img src="https://www.triviaquestionsnow.com/img/category/360x130/ap
## [7] <img src="https://www.triviaquestionsnow.com/img/category/360x130/-c
## [8] <img src="https://www.triviaquestionsnow.com/img/category/360x130/-c
```

```
html_obj %>% html_nodes("a")
```

```
## {xml_nodeset (44)}
##  [1] <a href="/">\n                <img src="https://www.triviaquestions
##  [2] <a href="https://www.triviaquestionsnow.com" class="no-pad">\n
##  [3] <a href="https://www.triviaquestionsnow.com/easy-trivia-questions">
##  [4] <a href="https://www.triviaquestionsnow.com/for/sports-trivia">Spor
##  [5] <a href="https://www.triviaquestionsnow.com/for/music-trivia">Music
```

`html_attrs()`

Getting a specific attribute from those nodes:

```
html_obj %>% html_nodes("img") %>% html_attr("src")
```

```
## [1] "https://www.triviaquestionsnow.com/img/trivia-questions.png"
## [2] "https://www.triviaquestionsnow.com/img/trivia-questions.png"
## [3] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1
## [4] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1
## [5] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1
## [6] "https://www.triviaquestionsnow.com/img/category/360x130/apologetics
## [7] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1
## [8] "https://www.triviaquestionsnow.com/img/category/360x130/-category-2
```

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

```
##  [1] "/"
##  [2] "https://www.triviaquestionsnow.com"
##  [3] "https://www.triviaquestionsnow.com/easy-trivia-questions"
##  [4] "https://www.triviaquestionsnow.com/for/sports-trivia"
##  [5] "https://www.triviaquestionsnow.com/for/music-trivia"
##  [6] "https://www.triviaquestionsnow.com/for/math-trivia"
##  [7] "https://www.triviaquestionsnow.com/categories"
```

html_text()

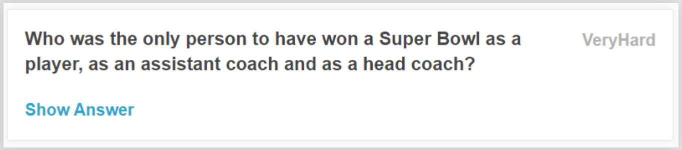
Getting the text from whatever set of elements we have:

```r
html_obj %>% html_nodes("a") %>% html_text()
```

```
##  [1] "\n                "
##  [2] "\n                            "
##  [3] "Easy Trivia"
##  [4] "Sports Trivia"
##  [5] "Music Trivia"
##  [6] "Math Trivia"
##  [7] "Categories"
##  [8] "All Trivia"
##  [9] "\n                What male tennis player has won the most Grand S
## [10] "Show Answer"
## [11] "\n                In what state is the Pro Football Hall of Fame l
## [12] "Show Answer"
## [13] "\n                Which team won the first Super Bowl ever?\n
## [14] "Show Answer"
## [15] "\n                Which player from the 1998 NFL Draft is considere
## [16] "Show Answer"
## [17] "\n                In American football, how many points is a touch
## [18] "Show Answer"
## [19] "\n                Andre Agassi lost the Wimbledon Championship fin
## [20] "Show Answer"
```

How to get to those questions? Option 1

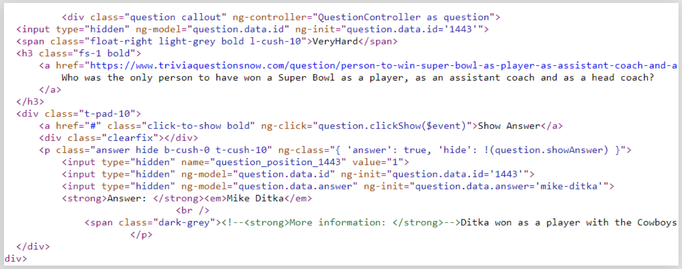
Look at the page source, get some identifier yourself (class, ID, link)



Who was the only person to have won a Super Bowl as a player, as an assistant coach and as a head coach?

VeryHard

Show Answer



```
        <div class="question callout" ng-controller="QuestionController as question">
  <input type="hidden" ng-model="question.data.id" ng-init="question.data.id='1443'">
  <span class="float-right light-grey bold l-cush-10">VeryHard</span>
  <h3 class="fs-1 bold">
      <a href="https://www.triviaquestionsnow.com/question/person-to-win-super-bowl-as-player-as-assistant-coach-and-a
          Who was the only person to have won a Super Bowl as a player, as an assistant coach and as a head coach?
      </a>
  </h3>
  <div class="t-pad-10">
      <a href="#" class="click-to-show bold" ng-click="question.clickShow($event)">Show Answer</a>
      <div class="clearfix"></div>
      <p class="answer hide b-cush-0 t-cush-10" ng-class="{ 'answer': true, 'hide': !(question.showAnswer) }">
          <input type="hidden" name="question_position_1443" value="1">
          <input type="hidden" ng-model="question.data.id" ng-init="question.data.id='1443'">
          <input type="hidden" ng-model="question.data.answer" ng-init="question.data.answer='mike-ditka'">
          <strong>Answer: </strong><em>Mike Ditka</em>
                              <br />
              <span class="dark-grey"><!--<strong>More information: </strong>-->Ditka won as a player with the Cowboys
                      </p>
  </div>
div>
```

```
html_obj %>% html_nodes(".question") %>% .[[1]]
```

```
## {html_node}
## <div class="question callout" ng-controller="QuestionController as quest
## [1] <input type="hidden" ng-model="question.data.id" ng-init="question.d
## [2] <span class="float-right light-grey bold l-cush-10">Medium</span>
## [3] <h3 class="fs-1 bold">\n                    <a href="https://www.triviaquest
## [4] <div class="t-pad-10">\n                    <a href="#" class="click-to-show
```

After some trial and error...

```r
html_obj %>%
  html_nodes(".question") %>%
  html_nodes(".fs-1") %>%
  html_text() %>%
  str_trim()
```

```
##  [1] "What male tennis player has won the most Grand Slam titles?"
##  [2] "In what state is the Pro Football Hall of Fame located?"
##  [3] "Which team won the first Super Bowl ever?"
##  [4] "Which player from the 1998 NFL Draft is considered by many to be t
##  [5] "In American football, how many points is a touchdown worth?"
##  [6] "Andre Agassi lost the Wimbledon Championship final in 1999. Which 
##  [7] "Andre Agassi won his first Olympic gold medal in which year?"
##  [8] "Which male golfer was the winner of 2018 U.S. Open Champion?"
##  [9] "Novak Djokovic won the 2013 Australian Open tournament. Who did he
## [10] "Up until 2018, who was the only player in NFL history to have rush
```

How to get to those questions? Option 2

SelectorGadget!

From here it's a function fest!

```r
extract_questions_and_answers_from_page <- function(url) {
  html_obj <- read_html(url)
  levels <- html_obj %>% html_nodes(".question") %>%
    html_nodes(".l-cush-10") %>% html_text()
  questions <- html_obj %>% html_nodes(".question") %>%
    html_nodes(".fs-1") %>% html_text() %>% str_trim()
  answers <- html_obj %>% html_nodes(".question") %>%
    html_nodes(".answer") %>% html_text() %>%
    str_extract(., "Answer:.*") %>% str_replace("Answer: ", "")
  tibble(level = levels, question = questions, answer = answers)
}

extract_questions_and_answers_from_page(url)
```

```
## # A tibble: 10 x 3
##    level    question                                           answer
##    <chr>    <chr>                                              <chr>
##  1 Easy     "The winning team of the Davis Cup is called?"     World
##  2 Hard     "What was the duration of the longest playoff drough~ 44 yea
##  3 Hard     "In which year did Fred Couples win his first major ~ 1992
##  4 Hard     "Which player graced the cover of the videogame \"Ma~ Rob Gr
##  5 Medium   "In 2010, which NBA player posed nude for an issue o~ Amare
##  6 VeryHard "Which Super Bowl had the most points ever scored?"   Super
##  7 Easy     "Which country won the 2012 Fed Cup?"                 Czech
##  8 Easy     "Bernhard Langer is a distinguished golfer. Which co~ German
```

Pagination

```r
create_page_url <- function(topic, page_num) {
  str_c("https://www.triviaquestionsnow.com/for/", topic, "-trivia
}

extract_multiple_pages_single_topic <- function(topic, n = 5) {
  cat(topic, "\n")
  res <- map_dfr(
    1:n,
    function(i) {
      cat("  ", i)
      url <- create_page_url(topic, i)
      extract_questions_and_answers_from_page(url)
    }
  )
  res$topic <- topic
  cat("\n")
  res
}
```

```r
extract_multiple_pages_single_topic("sports")
```

```
## # A tibble: 50 x 4
##    level    question                                         answer
##    <chr>    <chr>                                            <chr>
##  1 VeryHard "In 1948, which NBA basketball team did the H~ Minneapolis L
##  2 Medium   "The Jacksonville Jaguars and Carolina Panthe~ 1995
##  3 Hard     "An automatic progression by a player to the ~ Bye
##  4 VeryHard "In 2016, Giants' wide receiver Odell Beckham~ Code Black
##  5 Hard     "Which NBA player broke the record for most p~ Jeremy Lin
##  6 Medium   "Before relocating to Foxborough, Massachuset~ Boston
##  7 Easy     "What is the term for the historic jerseys to~ Throwback Jer
##  8 Medium   "Who served as the starting center of the Gol~ Andrew Bogut
##  9 Easy     "In what year was the 4 minute mile achieved?" 1954 by Roger
## 10 Hard     "Who was the first tennis player to complete ~ Don Budge
## # ... with 40 more rows
```

Magic!

```
topics <- c("sports", "kids", "science", "bible", "food-drink", "h

df_all <- map_dfr(
  topics,
  extract_multiple_pages_single_topic
)

df_all %>% count(topic)
```

```
## # A tibble: 8 x 2
##   topic          n
##   <chr>      <int>
## 1 bible         50
## 2 food-drink    50
## 3 geography     50
## 4 history       50
## 5 kids          50
## 6 science       50
## 7 sports        50
## 8 video-games   50
```

BeautifulSoup

APPLICATIONS

OF DATA SCIENCE

Almost always start with

```python
import requests
from bs4 import BeautifulSoup

html_obj = requests.get('https://en.wikipedia.org/wiki/List_of_The

soup = BeautifulSoup(html_obj.content, 'html.parser')
type(soup)
```

```
## <class 'bs4.BeautifulSoup'>
```

This object has all sorts of attributes and methods:

```python
soup.get_text()
soup.prettify()
soup.attrs
soup.children
soup.title
```

`find()` a tag, `find_all()`

```
link_objs = soup.find_all('a', href=True)
type(link_objs)
```

```
## <class 'bs4.element.ResultSet'>
```

```
type(link_objs[3])
```

```
## <class 'bs4.element.Tag'>
```

```
link_objs[3].text
```

```
## 'media franchise'
```

```
link_objs[3].attrs
```

```
## {'href': '/wiki/Media_franchise', 'title': 'Media franchise'}
```

See the actual link in the page.

Getting that `table`

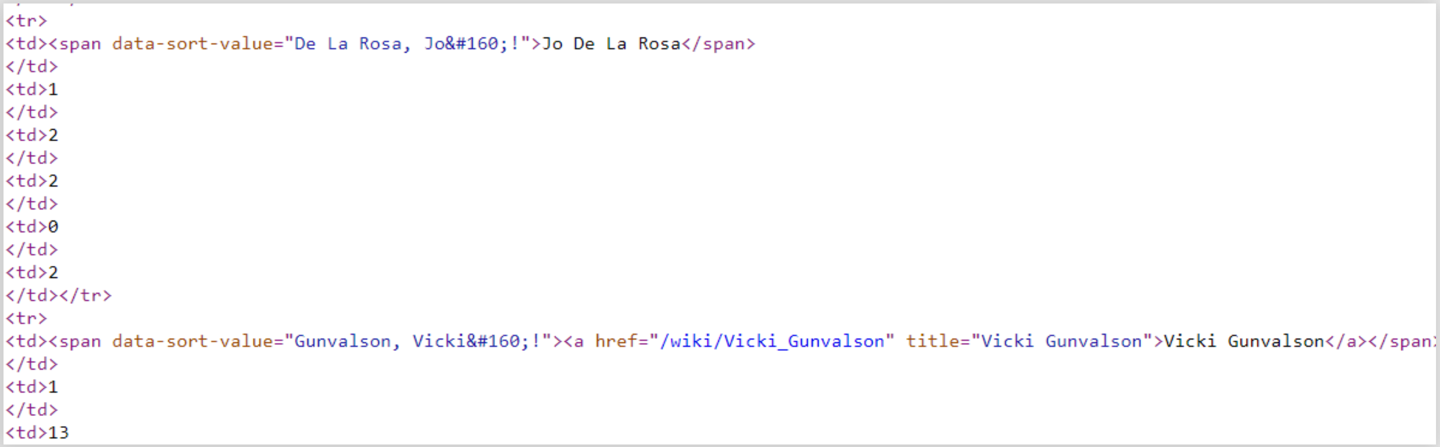
```
table = soup.find('table', attrs={'class':'wikitable'})
table_body = table.find('tbody')

rows = table_body.find_all('tr')
print(len(rows))
```

```
## 142
```

```
print(rows[0])
```

```
## <tr>
## <th rowspan="2">Installment
## </th>
## <th rowspan="2">Housewives
## </th>
## <th rowspan="2">First season<br/>starred
## </th>
## <th rowspan="2">Last season<br/>starred
## </th>
## <th colspan="3">Number of seasons
## </th></tr>
```

Getting a Housewife name



```
<tr>
<td><span data-sort-value="De La Rosa, Jo&#160;!">Jo De La Rosa</span>
</td>
<td>1
</td>
<td>2
</td>
<td>2
</td>
<td>0
</td>
<td>2
</td></tr>
<tr>
<td><span data-sort-value="Gunvalson, Vicki&#160;!"><a href="/wiki/Vicki_Gunvalson" title="Vicki Gunvalson">Vicki Gunvalson</a></span
</td>
<td>1
</td>
<td>13
```

```python
import re

print(rows[3].find('span', attrs = {'data-sort-value': re.compile
```

```
## <span data-sort-value="De La Rosa, Jo&nbsp;!">Jo De La Rosa</span>
```

Getting only HWives with Wiki pages

```
housewives_with_links = []
for row in rows:
  housewife = row.find('span',
    attrs = {'data-sort-value': re.compile(r'.*')})
  if housewife is not None:
    link = housewife.find('a')
    if link is not None:
      housewives_with_links.append((housewife.text, link['href']))

import pandas as pd

h_df = pd.DataFrame(housewives_with_links, columns=['name', 'link'

h_df.head()
```

```
##                      name                         link
## 0        Vicki Gunvalson        /wiki/Vicki_Gunvalson
## 1           Jeana Keough           /wiki/Jeana_Keough
## 2         Heather Dubrow         /wiki/Heather_Dubrow
## 3  Shannon Storms Beador  /wiki/Shannon_Storms_Beador
## 4       Bethenny Frankel       /wiki/Bethenny_Frankel
```

(Though if your table is simple, try:)

```
l = pd.read_html(html_obj.text)

l[0].head()
```

```
##      Installment        Housewives  ... Number of seasons
##      Installment        Housewives  ...            Friend Guest
## 0  Orange County  Kimberly Bryant  ...               0.0   3.0
## 1  Orange County    Jo De La Rosa  ...               0.0   2.0
## 2  Orange County  Vicki Gunvalson  ...               1.0   0.0
## 3  Orange County     Jeana Keough  ...               1.0   4.0
## 4  Orange County   Lauri Peterson  ...               1.0   1.0
##
## [5 rows x 7 columns]
```

Following HWives Links

```
def get_housewife_img_ref(housewife_link):
  html_obj = requests.get('https://en.wikipedia.org' + housewife_l
  soup = BeautifulSoup(html_obj.content, 'html.parser')
  infobox = soup.find('table', attrs = {'class': 'vcard'})
  if infobox is not None:
    img_obj = infobox.find('img', src=True)
    if img_obj is not None:
      return img_obj['src']
  return None

h_df['img_ref'] = h_df['link'].apply(get_housewife_img_ref)
h_df.dropna(inplace=True)

h_df.head()
```

```
##                        name  ...
## 0           Vicki Gunvalson  ...   //upload.wikimedia.org/wikipedia/common
## 1          Luann de Lesseps  ...   //upload.wikimedia.org/wikipedia/common
## 2          Bethenny Frankel  ...   //upload.wikimedia.org/wikipedia/common
## 3  Kelly Killoren Bensimon  ...   //upload.wikimedia.org/wikipedia/common
## 4          Carole Radziwill  ...   //upload.wikimedia.org/wikipedia/common
##
## [5 rows x 3 columns]
```

Downloading HWives Images

```
def make_img_filename(hf_name):
  return 'data/housewives/' + hf_name.lower().strip(',.-').replace

def download_hw_img(hf_name, hf_img_ref):
  img_file = make_img_filename(hf_name)
  img_data = requests.get('http:' + hf_img_ref).content
  with open(img_file, 'wb') as handler:
      handler.write(img_data)

h_df.apply(lambda row: download_hw_img(row['name'], row['img_ref']
  axis=1)
```