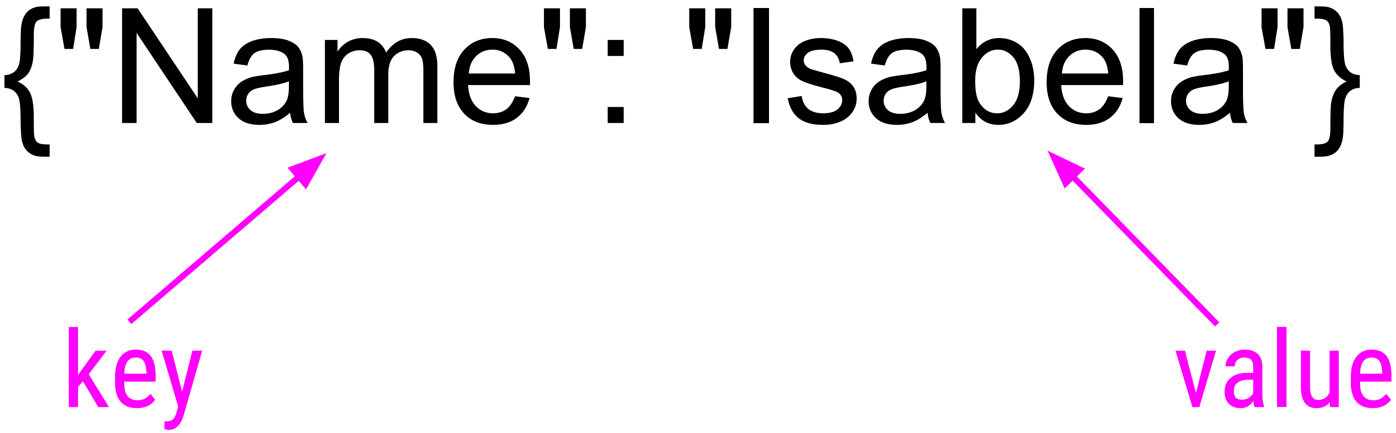
Getting data from hierarchical sources

Getting and Cleaning Data

JSON: JavaScript Object Notation

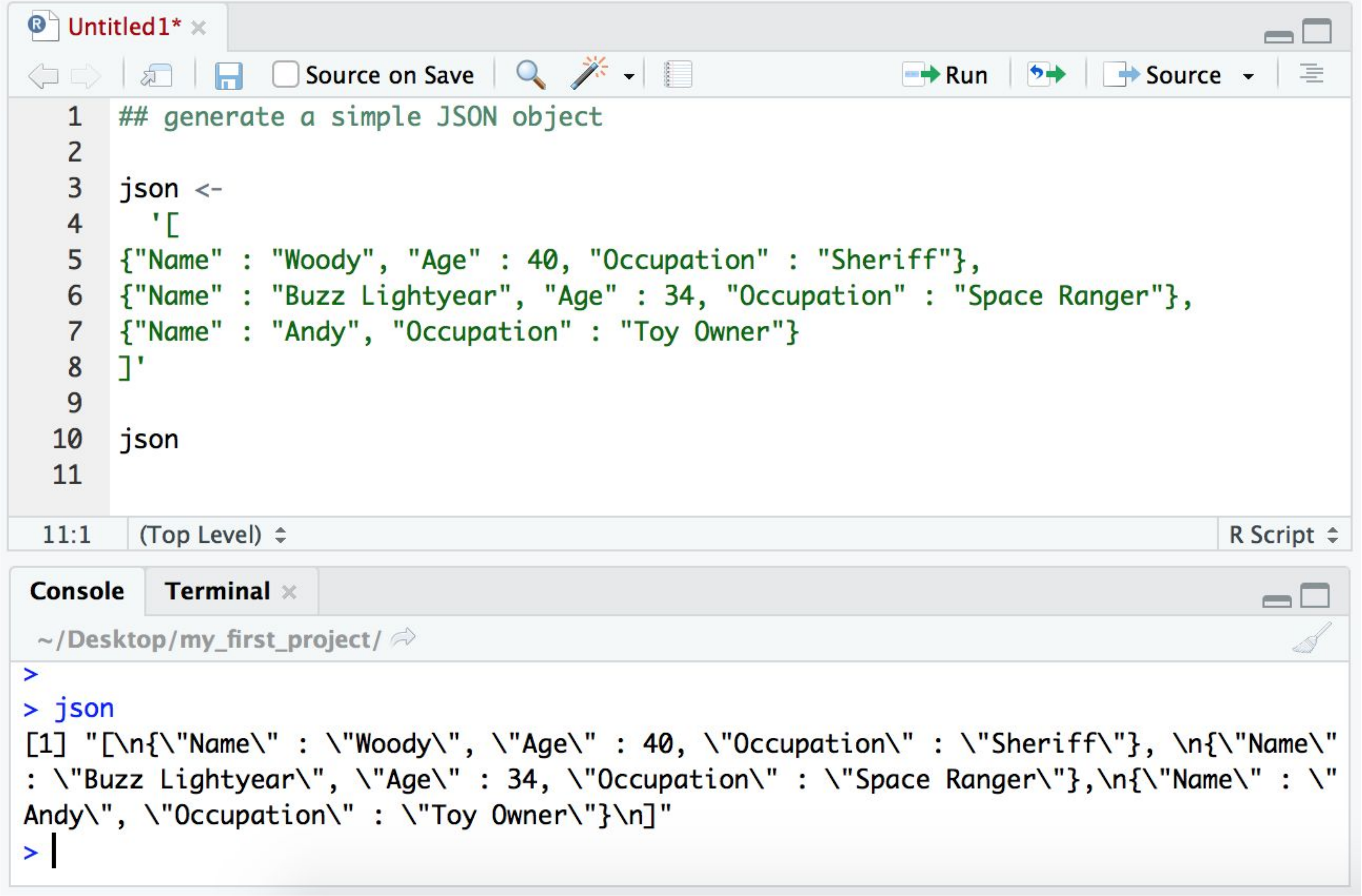
key-value pairs



```
{"Name": "Isabela"}
```

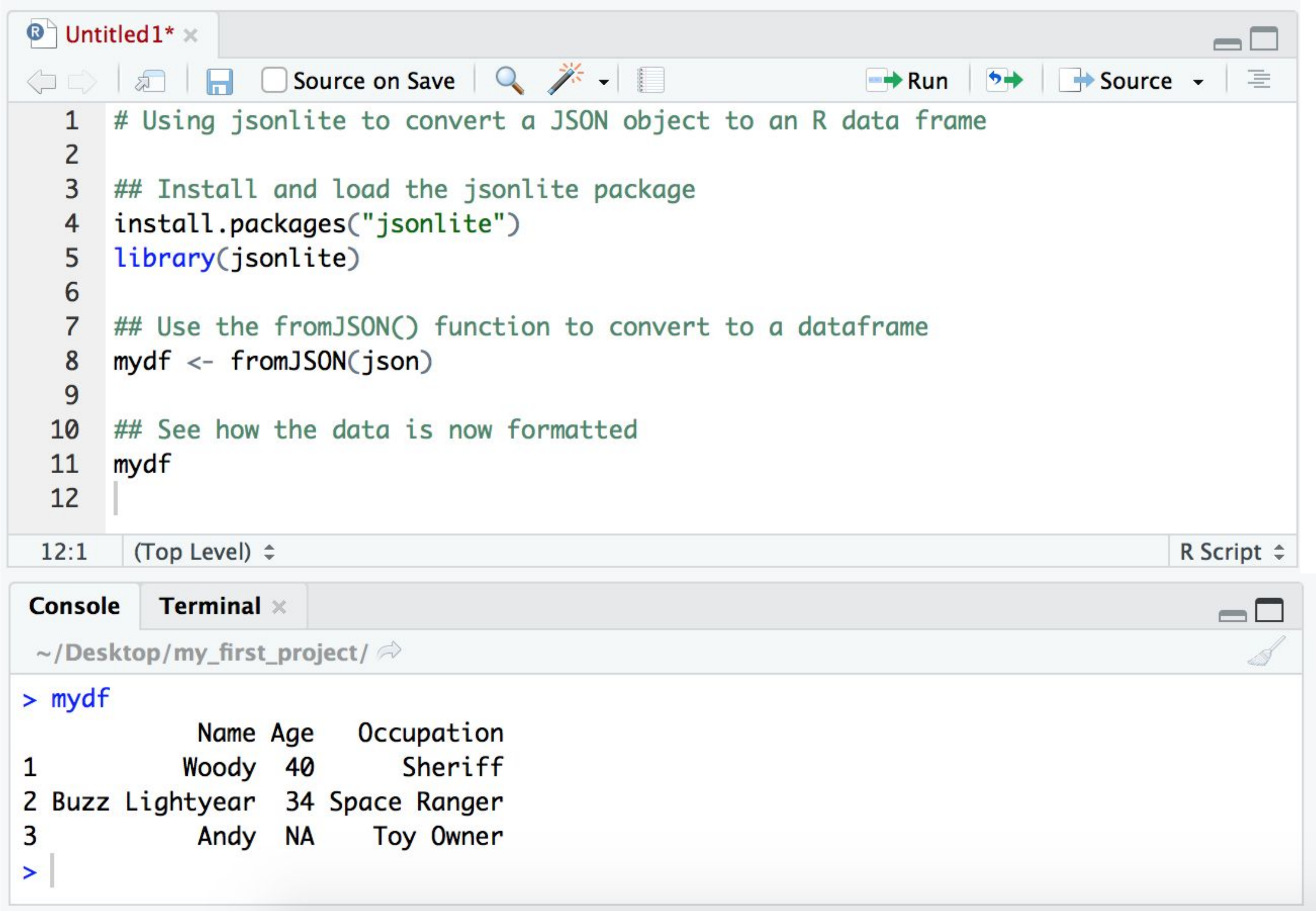
key

value



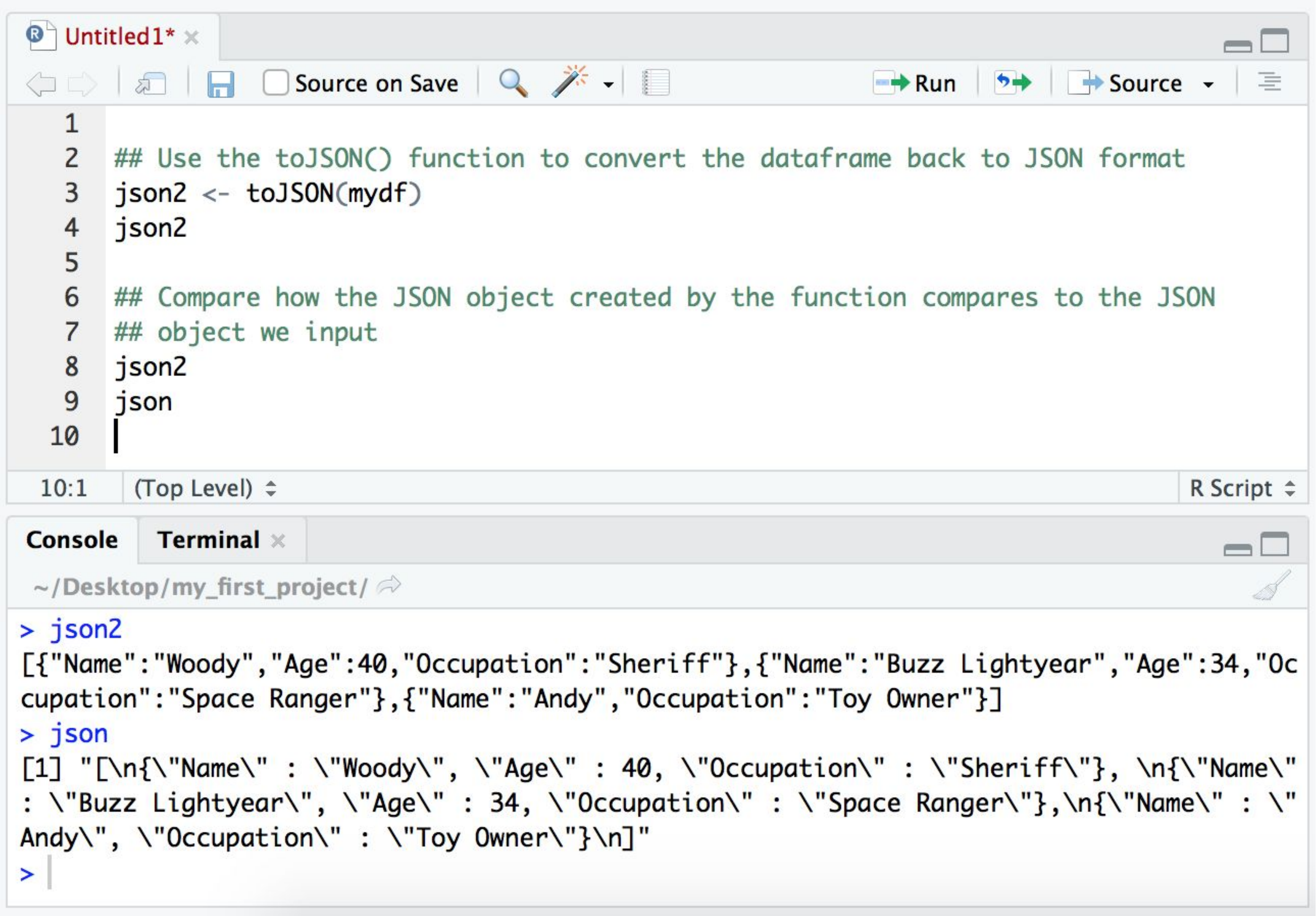
```
## generate a simple JSON object

json <-
  '[
{"Name" : "Woody", "Age" : 40, "Occupation" : "Sheriff"},
{"Name" : "Buzz Lightyear", "Age" : 34, "Occupation" : "Space Ranger"},
{"Name" : "Andy", "Occupation" : "Toy Owner"}
]'

json

```

```
>
> json
[1] "[\n{\"Name\" : \"Woody\", \"Age\" : 40, \"Occupation\" : \"Sheriff\"}, \n{\"Name\"
: \"Buzz Lightyear\", \"Age\" : 34, \"Occupation\" : \"Space Ranger\"},\n{\"Name\" : \"
Andy\", \"Occupation\" : \"Toy Owner\"}\n]"
> 
```



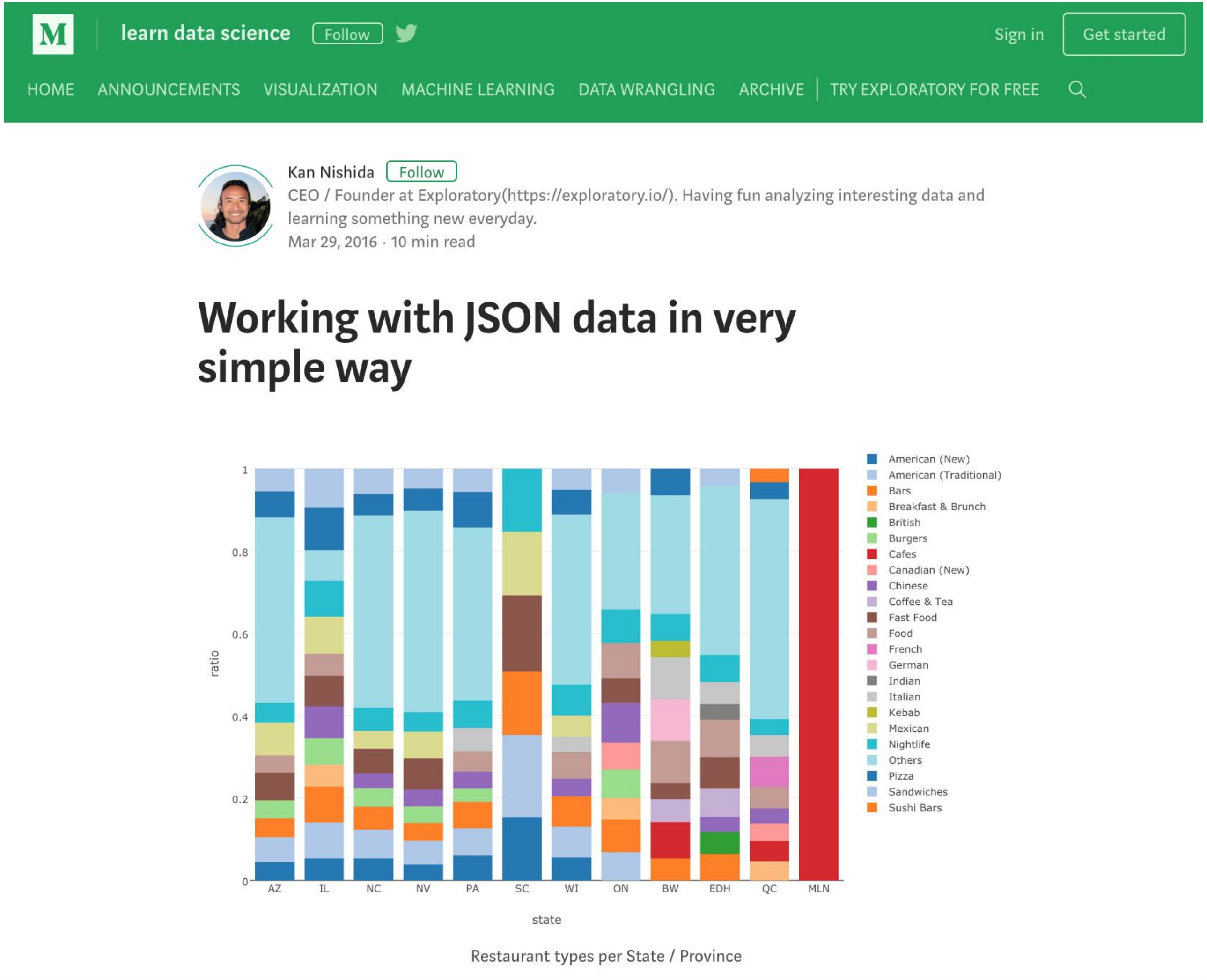
```r
# Using jsonlite to convert a JSON object to an R data frame

## Install and load the jsonlite package
install.packages("jsonlite")
library(jsonlite)

## Use the fromJSON() function to convert to a dataframe
mydf <- fromJSON(json)

## See how the data is now formatted
mydf

```

```
> mydf
           Name Age    Occupation
1         Woody  40       Sheriff
2 Buzz Lightyear  34 Space Ranger
3          Andy  NA     Toy Owner
> 
```



```

## Use the toJSON() function to convert the dataframe back to JSON format
json2 <- toJSON(mydf)
json2

## Compare how the JSON object created by the function compares to the JSON
## object we input
json2
json

```

```
> json2
[{"Name":"Woody","Age":40,"Occupation":"Sheriff"},{"Name":"Buzz Lightyear","Age":34,"Oc
cupation":"Space Ranger"},{"Name":"Andy","Occupation":"Toy Owner"}]
> json
[1] "[\n{\"Name\" : \"Woody\", \"Age\" : 40, \"Occupation\" : \"Sheriff\"}, \n{\"Name\"
: \"Buzz Lightyear\", \"Age\" : 34, \"Occupation\" : \"Space Ranger\"},\n{\"Name\" : \"
Andy\", \"Occupation\" : \"Toy Owner\"}\n]"
>
```



Kan Nishida

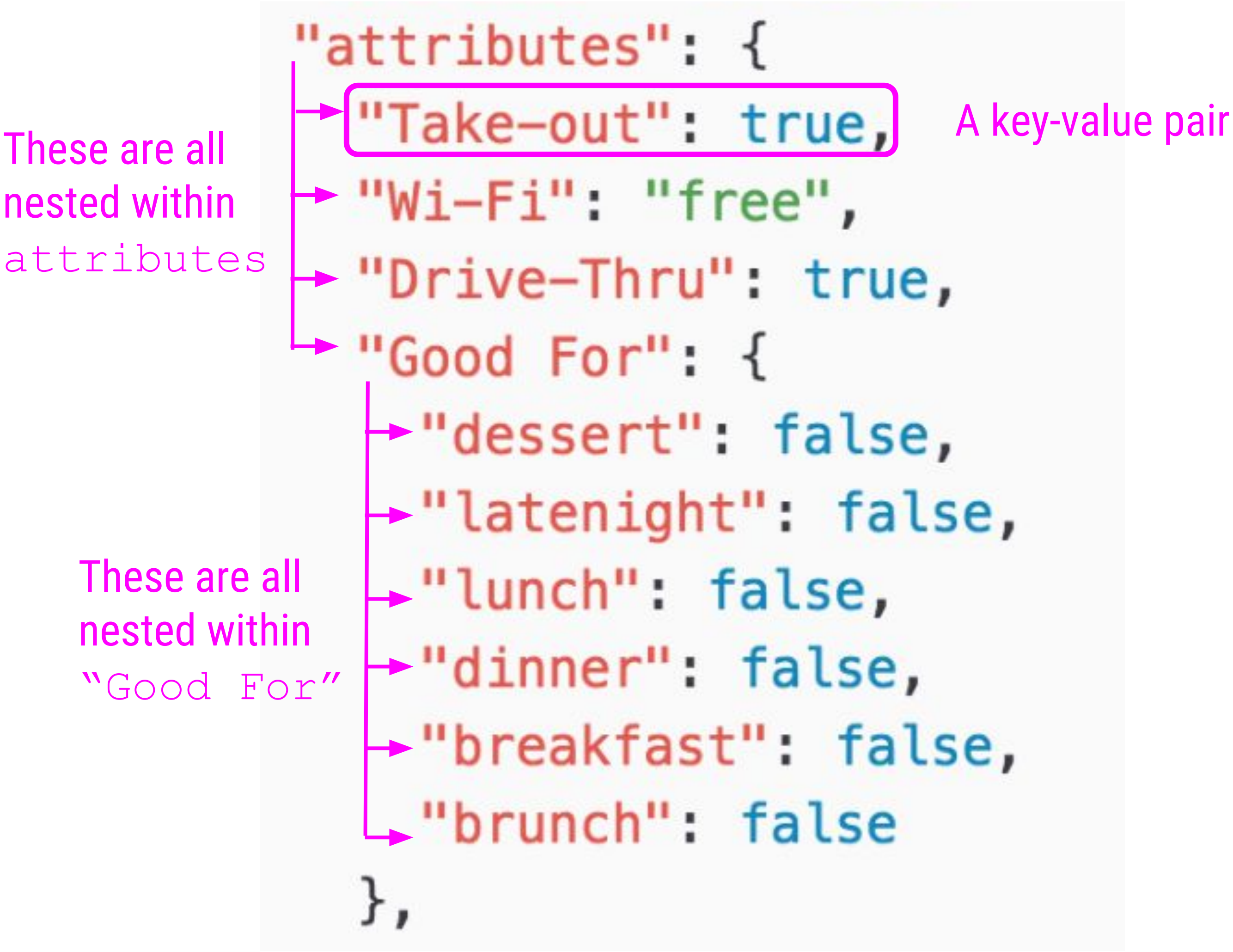
CEO / Founder at Exploratory(https://exploratory.io/). Having fun analyzing interesting data and learning something new everyday.

Mar 29, 2016 · 10 min read

Working with JSON data in very simple way

Restaurant types per State / Province

https://blog.exploratory.io/working-with-json-data-in-very-simple-way-ad7ebcc0bb89



```
"attributes": {
  "Take-out": true,
  "Wi-Fi": "free",
  "Drive-Thru": true,
  "Good For": {
    "dessert": false,
    "latenight": false,
    "lunch": false,
    "dinner": false,
    "breakfast": false,
    "brunch": false
  },
```

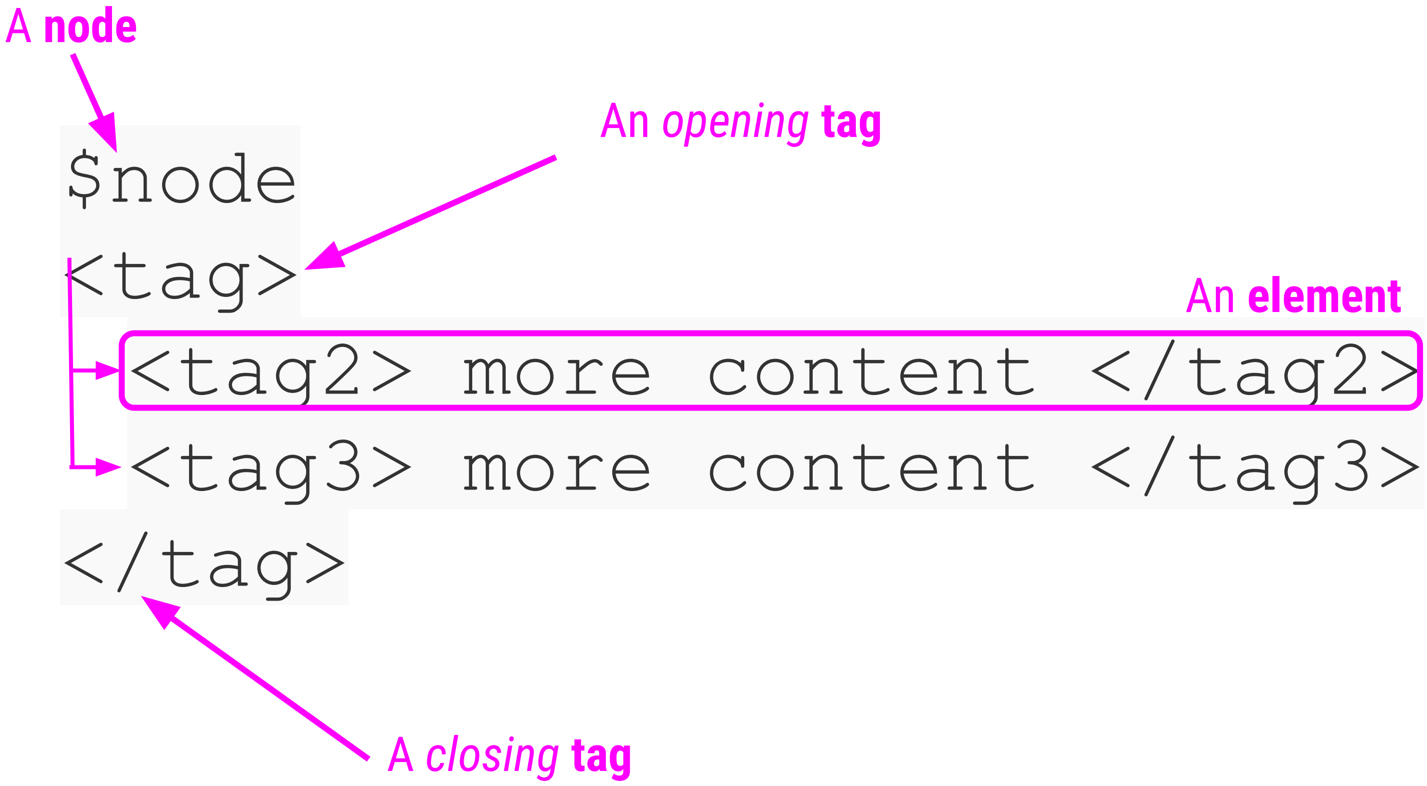
A key-value pair

These are all nested within attributes

These are all nested within “Good For”

https://blog.exploratory.io/working-with-json-data-in-very-simple-way-ad7ebcc0bb89

XML: Extensible Markup Language



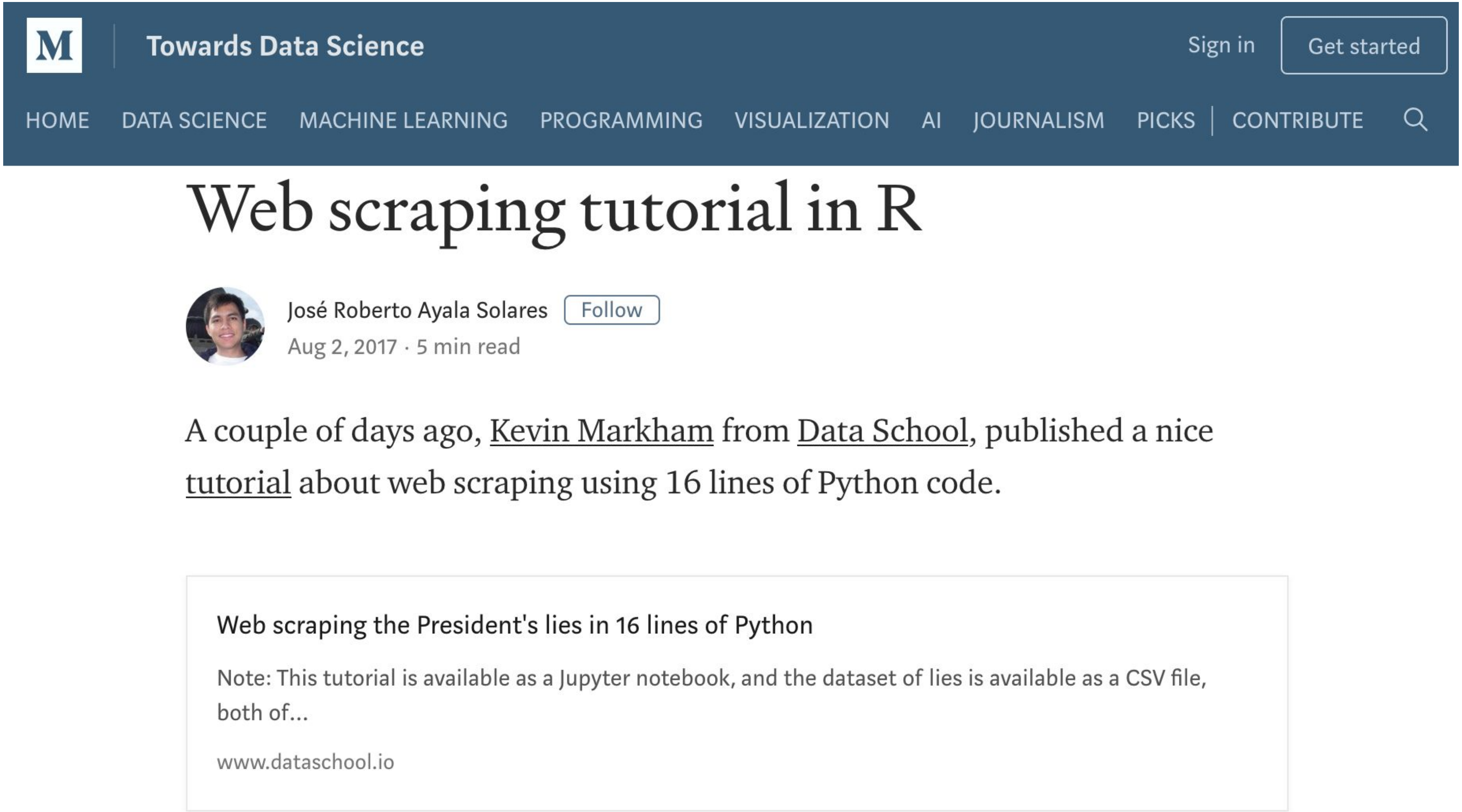
A **node**

An *opening* **tag**

An **element**

A *closing* **tag**

```
$node
<tag>
  <tag2> more content </tag2>
  <tag3> more content </tag3>
</tag>
```



Web scraping tutorial in R

José Roberto Ayala Solares

Aug 2, 2017 · 5 min read

A couple of days ago, Kevin Markham from Data School, published a nice tutorial about web scraping using 16 lines of Python code.

Web scraping the President's lies in 16 lines of Python

Note: This tutorial is available as a Jupyter notebook, and the dataset of lies is available as a CSV file, both of...

www.dataschool.io

Summarizing: Getting data from hierarchical sources

Getting and Cleaning Data