Web Scraping in R: Takeaways

Syntax

- Importing rvest:

```
library(rvest)
```

- Getting the HTML document

```
content  <- read_html("url")
```

- Getting the inside text of a tag:

```
text <- content %>%
  html_nodes("tag_name") %>%
  html_text()
```

- Getting the inside text using an ID name:

```
text <- content %>%
  html_nodes("#id_name") %>%
  html_text()
```

- Getting the inside text using a class name:

```
text <- content %>%
  html_nodes(".class_name") %>%
  html_text()
```

- Getting a table into a dataframe:

```
text <- content %>%
  html_nodes("table") %>%
  html_table()
```

- Returning all attribute values:

```
text <- content %>%
  html_nodes("tag_name or #id_name or .class_name") %>%
  html_attrs()
```

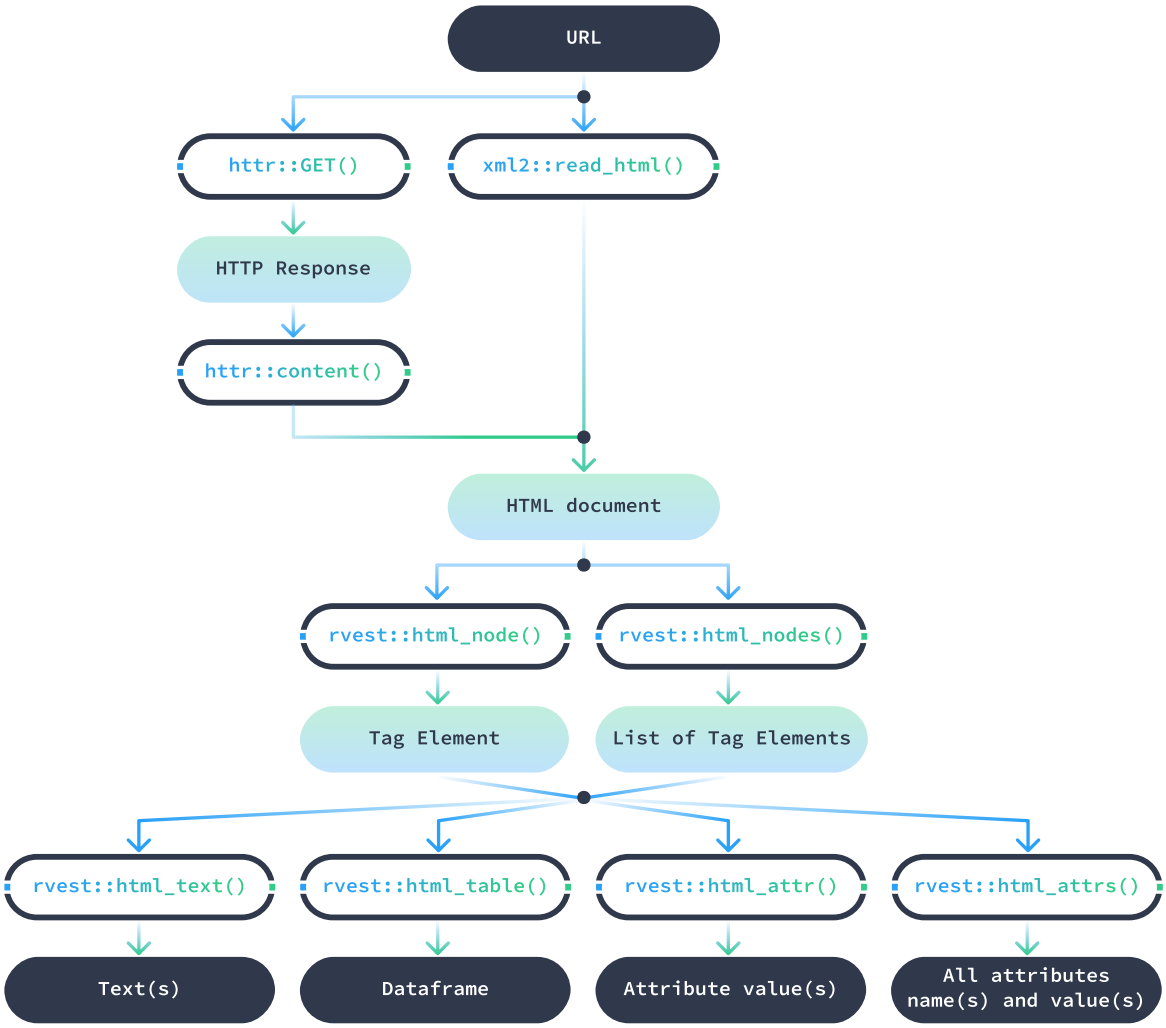
- Returning an attribute values:

```
text <- content %>%
  html_nodes("tag_name or #id_name or .class_name") %>%
  html_attr("attribute_name")
```

Concepts

- Datasets and APIs aren't the only way to access data. A great deal of data exists on the internet in the form of web pages. We can use web scraping to access the data without waiting for the provider to create an API.
- We can use the `xml2` library to download a web page, and we can use `rvest` to extract the relevant parts of the web page.

- Web pages use HyperText Markup Language (HTML) as the foundation for the content on the page. Browsers like Google Chrome and Mozilla Firefox read the HTML to determine how to render and display the page.
- The `head` tag in HTML contains information useful to the web browser that's rendering the page. The `body` section contains the bulk of the content the user interacts with on the page. The `title` tag tells the web browser what page title to display in the toolbar.
- HTML allows elements to have IDs or classes we can use to refer to specific elements (IDs are unique, and classes are not).
- Web scraping process:



Resources

- HTML basics
- HTML element
- rvest Documentation