



## Web scraping

## Web technology

I thank Dr Emi Tanaka for this part, adapted from her "Communicating with Data" course.

2 / 33

### World Wide Web (WWW)

WWW (or the **Web**) is the information system where documents (web pages) are identified by Uniform Resource Locators (**URLs**)

A web page consists of:

-  **HTML** provides the basic structure of the web page
-  **CSS** controls the look of the web page (optional)
-  **JS** is a programming language that can modify the behaviour of elements of the web page (optional)

3 / 33

4 / 33

### Hypertext Markup Language (HTML)

- with the extension **.html**.
- rendered using a web browser via an URL.
- text files that follows a special syntax that alerts web browsers how to render it.

## HTML structure

```
<!DOCTYPE html>
<html>
  <!--This is a comment and ignored by web client.-->
  <head>
    <!--This section contains web page metadata.-->
    <title>STATS 220 Data Technology</title>
    <meta name="author" content="Earo Wang">
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <!--This section contains what you want to display on your web page.-->
    <h1>I'm a first level header</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

5 / 33

## HTML syntax

```
<span style="color:blue;">Author content</span> → Author content
start tag: <span style="color:blue;">Author content</span>
end tag: <span style="color:blue;">Author content</span>
content: <span style="color:blue;">Author content</span>
element name: <span style="color:blue;">Author content</span>
attribute: <span style="color:blue;">Author content</span>
attribute name: <span style="color:blue;">Author content</span>
attribute value: <span style="color:blue;">Author content</span>
```

Not all HTML tags have an end tag:

```
 → R
```

6 / 33

## HTML elements

```
block element: <div>content</div>
inline element: <span>content</span>
paragraph: <p>content</p>
header level 1: <h1>content</h1>
header level 2: <h2>content</h2>
italic: <i>content</i>
emphasised text: <em>content</em>
strong importance: <strong>content</strong>
link: <a href="https://stats220.earo.me/">content</a>
unordered list: <ul>
  <li>item 1</li>
  <li>item 2</li>
</ul>
```

7 / 33

## Cascading Style Sheet (CSS)

- with the extension .css
- 3 ways to style elements in HTML:
  - **inline** by using the `style` attribute inside HTML start tag:  
`<h1 style="color:blue;">Blue Header</h1>`
  - **externally** by using the `<link>` element:  
`<link rel="stylesheet" href="styles.css">`
  - **internally** by defining within `<style>` element:  
`<style type="text/css">
h1 { color: blue; }
</style>`

By convention, the `<style>` and `<link>` elements tend to go into the `<head>` section of the HTML document.

8 / 33

## ⌚ CSS syntax

```
<style type="text/css">
h1 { color: blue; }
</style>
This is a header</h1>
```

selector: h1 { color: blue; }  
property: h1 { color: blue; }  
property name: h1 { color: blue; }  
property value: h1 { color: blue; }

You may have multiple properties for a single selector.➡

```
h1 {
  color: blue;
  font-size: 16pt;
}
```

9 / 33

## ⌚ CSS properties

```
<div>Sample text</div>
```

background color: div { background-color: yellow; }  
text color: div { color: purple; }  
border: div { border: 1px dashed brown; }  
left border only: div { border-left: 10px solid pink; }  
text size: div { font-size: 10pt; }  
padding: div { background-color: yellow;
 padding: 10px; }  
margin: div { background-color: yellow;
 margin: 10px; }

Sample text  
Sample text

10 / 33

## ⌚ CSS selector

.classname selects all elements with the attribute  
the attribute  
class="classname".  
.c1.c2 selects all elements with both c1 and c2 within its class attribute.  
.c1 .c2 selects all elements with class c2 that is a descendant of an element with class c1.  
#p1 selects all elements with the attribute id="p1".

⚠ Unlike class, you can only have one id value and must be unique in the whole HTML document.

```
<h1>This is a sample html</h1>
<blockquote>
<p>Maybe stories are just dreams</p>
<div><h1>Bren Brown</h1>
<div>How are you?
<div class="child nice">
  <p>Hello!</p>
</div>
</div>
</div>
<div id="p1" class="parent">
  Hm
  <h1>Hello!</h1>
  <p>How are you?
  <div class="child nice">
    <p>Hello!</p>
  </div>
  </div>
</div>
<div class="parent">
<p>Household 1</p>
<div class="parent">
<p>I</p>
<div class="child rebel">
  <p>Don't talk to me!</p>
</div>
</div>
<div>
<span class="child">
<span class="parent child rebel">
  <p>Clean your room!</p>
</span>
</span>
</div>
<p>End of households</p>
```

11 / 33

## ⌚ JavaScript (JS)\*

➢ JS is a programming language and enable interactive components in HTML documents.

➢ 2 ways to insert JS into a HTML document:  
➢ internally by defining within <script> element:

```
<script>
document.getElementById("p1").innerHTML = "content";
</script>
```

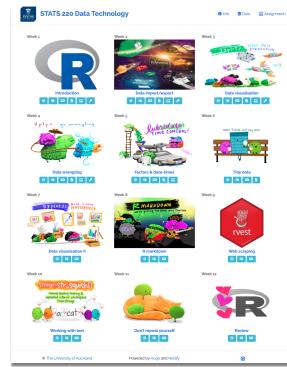
➢ externally by using the src attribute to refer to the external file:

```
<script src="js/myjs.js"></script>
```

12 / 33

## Web scraping 🕸️

13 / 33



Use {rvest} >= v1.0.0 (if not, update)

The screenshot shows the RStudio interface with the 'Packages' tab selected. A red box highlights the 'Update' button next to the 'rvest' package entry. The package details show it is version 1.0.0.

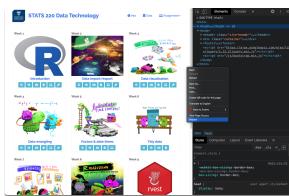
```
library(rvest)
course <- "https://stats220.eaero.me"
stats220 <- read_html(course)
stats220
```

```
#> {html_document}
#> <html>
#> [1] <head>\n<meta http-equiv="Content ...
#> [2] <body>\n\n    <header class="site ...
```

14 / 33

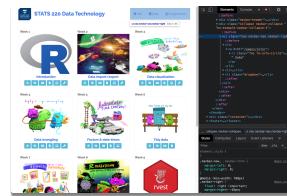
## 🔍 Inspect elements



## 🧩 CSS selectors



15 / 33



html\_element() select element

```
navbar <- stats220 %>%
  html_element(".navbar-right")
navbar
```

```
#> {html_node}
#> <ul class="nav navbar-nav navbar-right">
#> [1] <li><a href="/pages/info/"><i cla ...
#> [2] <li><a href="/pages/data/"><i cla ...
#> [3] <li class="dropdown">\n    ...
```

html\_name() get element name

```
navbar %>%
  html_name()
```

```
#> [1] "ul"
```

16 / 33

html\_children() get element children

```
navbar %>%
  html_children()

#> {xml_nodeset (3)}
#> [1] <li><a href="/pages/info/"><i cla ...
#> [2] <li><a href="/pages/data/"><i cla ...
#> [3] <li class="dropdown">\n  ...
  ...
  ...

html_name() get element name
```

```
navbar %>%
  html_children() %>%
  html_name()

#> [1] "li" "li" "li"
```

17 / 33

html\_text2() get element text

```
navbar %>%
  html_children() %>%
  html_text2()

#> [1] "Info"
#> [2] "Data"
#> [3] "Assignment\nAssignment 1\nAssignment 2
```

html\_attr() get element attributes

```
navbar %>%
  html_elements("a") %>%
  html_attr("href")

#> [1] "/pages/info/"
#> [2] "/pages/data/"
#> [3] "#"
#> [4] "/assignments/assignment1/"
```

18 / 33

## url\_absolute(): turn relative urls to absolute urls

```
navbar %>%
  html_elements("a") %>%
  html_attr("href") %>%
  url_absolute(course)

#> [1] "https://stats220.earo.me/pages/info/"
#> [2] "https://stats220.earo.me/pages/data/"
#> [3] "https://stats220.earo.me#"
#> [4] "https://stats220.earo.me/assignments/assignment1/"
#> [5] "https://stats220.earo.me/assignments/assignment2/"
#> [6] "https://stats220.earo.me/assignments/assignment3/"
#> [7] "https://stats220.earo.me/assignments/assignment1-sol/"
#> [8] "https://stats220.earo.me/assignments/assignment2-sol/"
```

19 / 33

html\_elements() select elements

```
navbar %>%
  html_elements("i")

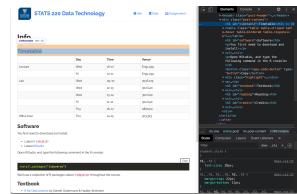
#> {xml_nodeset (8)}
#> [1] <i class="fas fa-info-circle"></i>
#> [2] <i class="fas fa-database"></i>
#> [3] <i class="fas fa-laptop-code"></i>
#> [4] <i class="fas fa-laptop-code"></i>
#> [5] <i class="fas fa-laptop-code"></i>
#> [6] <i class="fas fa-laptop-code"></i>
#> [7] <i class="fas fa-key"></i>
#> [8] <i class="fas fa-key"></i>
```

 fontawesome icons

20 / 33

select all <h3> elements

```
stats220_info <- read_html(  
  "https://stats220.earo.me/pages/info/")
```



```
stats220_info %>%  
  html_elements("h3") %>%  
  html_text()
```

select #timetable id

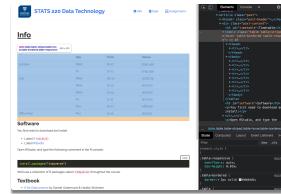
```
stats220_info %>%  
  html_elements("#timetable") %>%  
  html_text()
```

#> [1] "Timetable"

21 / 33

select the first <table> element

```
stats220_info %>%  
  html_element("table") %>%  
  html_table()
```



```
#> # A tibble: 8 x 4  
#>   ``` Day Time Venue  
#> 1 "Lecture" Wed 16-17 Eng1.439  
#> 2 "" Fri 10-11 Eng1.439  
#> 3 "Lab" Wed 09-10 303S.175  
#> 4 "" Wed 12-13 302.G40  
#> 5 "" Wed 13-14 302.G40  
#> 6 "" Fri 11-12 302.G40  
#> 7 "" Thu 16-17 offshore  
#> 8 "Office hour" Thu 14-15 303.323
```

22 / 33

## Download pdf slides at once

```
stats220_urls <- stats220 %>%  
  html_elements(".panel-body .btn") %>%  
  html_attr("href")  
stats220_urls
```

```
#> [1] "/objs/obj01"  
#> [2] "/R/01-intro.R"  
#> [3] "/01-intro.Rmd"  
#> [4] "/01-intro.pdf"  
#> [5] "/labs/lab01"  
#> [6] "/labs/lab01-sol"  
#> [7] "/objs/obj02"  
#> [8] "/R/02-import-export.R"  
#> [9] "/02-import-export.Rmd"  
#> [10] "/02-import-export.pdf"  
#> [11] "/labs/lab02"  
#> [12] "/labs/lab02-sol"  
#> [13] "/objs/obj03"  
#> [14] "/R/03-data-vis.R"  
#> [15] "/03-data-vis.Rmd"  
#> [16] "/03-data-vis.pdf"
```

23 / 33

## Download pdf slides at once

```
library(stringr) # manipulate strings in week 10  
(pdf_urls <- stats220_urls[str_detect(stats220_urls, "pdf"))]
```

```
#> [1] "/01-intro.pdf"  
#> [2] "/02-import-export.pdf"  
#> [3] "/03-data-vis.pdf"  
#> [4] "/04-data-wrangle.pdf"  
#> [5] "/05-fcts-dates.pdf"  
#> [6] "/06-tidy-data.pdf"  
#> [7] "/07-data-vis2.pdf"  
#> [8] "/08-rmd.pdf"
```

```
pdf_files <- str_remove(pdf_urls, "/")  
purrr::walk2( # below for week 11  
  url_absolute(pdf_urls, course), pdf_files,  
  ~ download.file(url = .x, destfile = .y))
```

24 / 33

# REST API

25 / 33

## Github REST API

- Each URL is called a **request**.
- The data sent back to you is called an **HTTP response** that consists of headers and a body.

The root-endpoint of Github's API is <https://api.github.com>.

```
library(httr)
endpoint <- "https://api.github.com"
GET(endpoint)
```

```
#> Response [https://api.github.com]
#>   Date: 2021-05-10 08:45
#>   Status: 200
#>   Content-Type: application/json; charset=utf-8
#>   Size: 2.31 kB
#> {
#>   "current_user_url": "https://api.github.com/users",
#>   "current_user_authorizations_html_url": "https://api.github.com/auth/authorizations",
#>   "authorizations_url": "https://api.github.com/auth/authorizations",
#>   "code_search_url": "https://api.github.com/search/code",
#>   "commit_search_url": "https://api.github.com/search/commit",
#>   "emails_url": "https://api.github.com/user/emails",
#>   "emojis_url": "https://api.github.com/images/emojis",
#>   "events_url": "https://api.github.com/events",
#>   "feeds_url": "https://api.github.com/feeds",
#>   ...
#> }
```

26 / 33

## HTTP methods

- GET** to **retrieve** resource data/information only, and NO change in state of the resource
- POST** to **create** new subordinate resources, e.g. upload a file
- PUT** to **update/replace** an existing resource in its entirety
- DELETE** to **delete** resources
- PATCH** to make **partial update** on a resource (not all browsers support PATCH)

27 / 33

## Path

The **path** determines the resource you're requesting for.

```
GET /repos/{owner}/{repo}
```

```
path <- "/repos/STATS-UOA/stats220"
resp <- GET(modify_url(endpoint,
                       path = path))
resp
```

```
#> Response [https://api.github.com/repos/STATS-UOA/stats220]
#>   Date: 2021-05-10 08:45
#>   Status: 200
#>   Content-Type: application/json; charset=utf-8
#>   Size: 6.41 kB
#> {
#>   "id": 242031925,
#>   "node_id": "MDEwOlJlcG9zaXRvcnkyNDIwNjUy",
#>   "name": "stats220",
#>   "full_name": "STATS-UOA/stats220",
#>   "private": false,
#>   "owner": {
#>     "login": "STATS-UOA",
#>     "id": 62915494,
#>     "node_id": "MDEyOk9yZ2FuaXphdGlvb2xvZ2lkb25hdGlvbmF0aW9ucy5jb20=",
#>     "avatar_url": "https://avatars.githubusercontent.com/u/62915494?v=4",
#>     "gravatar_id": null,
#>     "url": "https://api.github.com/users/STATS-UOA",
#>     "html_url": "https://github.com/STATS-UOA",
#>     "followers_url": "https://api.github.com/users/STATS-UOA/followers",
#>     "following_url": "https://api.github.com/users/STATS-UOA/following{/other_user}",
#>     "gists_url": "https://api.github.com/users/STATS-UOA/gists{/gist_id}",
#>     "starred_url": "https://api.github.com/users/STATS-UOA/starred{/owner_login}/{repo_name}",
#>     "subscriptions_url": "https://api.github.com/users/STATS-UOA/subscriptions",
#>     "organizations_url": "https://api.github.com/users/STATS-UOA/orgs",
#>     "repos_url": "https://api.github.com/users/STATS-UOA/repos",
#>     "events_url": "https://api.github.com/users/STATS-UOA/events{/privacy}",
#>     "received_events_url": "https://api.github.com/users/STATS-UOA/received_events",
#>     "type": "Organization",
#>     "site_admin": false
#>   },
#>   "permissions": {
#>     "admin": true,
#>     "push": true,
#>     "pull": true
#>   }
#> }
```

28 / 33

## Parse the response

```
http_type(resp)          content(resp)

#> [1] "application/json"
#> $id                  #> [1] 242031925
#> #> $node_id           #> [1] "MDEwOlJlcG9zaXRvcnkyNDIwMzE5MjU="
Content type of a response
#> "image/png"
#> "application/text"
#> "application/csv"
#> ...
#> $name                #> [1] "stats220"
#> #> $full_name         #> [1] "STATS-UOA/stats220"
#> #> $private            #> [1] FALSE
#> #> $owner               #> $owner$login
#> [1] "STATS-UOA"
#>
```

29 / 33

## Status code

```
status_code(resp)

#> [1] 200
http_status(200) # OK
http_status(201) # Created
http_status(204) # NO CONTENT
http_status(400) # BAD REQUEST
http_status(403) # Forbidden
http_status(404) # NOT FOUND
```

[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)

30 / 33

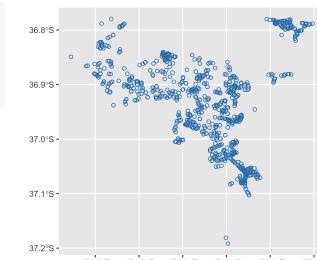
## Auckland Transport Open GIS Data: bus stop

```
endpoint <- "https://services2.arcgis.com/"
path <- "/jKPEGzJGxhSjY0o0/arcgis/rest/services/BusService/FeatureServer/0/"
query <- "query?where=1%3D1&outFields=&outSR=4326&f=geojson"
bus_url <- parse_url(paste0(endpoint, path, query))
resp <- GET(bus_url)
cnt <- geojson::as.geojson(content(resp))
cnt
```

```
#> <geojson>
#>   type: FeatureCollection
#>   features (n): 1000
#>   features (geometry / length) [first 5]:
#>     Point / 2
```

31 / 33

```
library(tidyverse)
library(sf)
bus_sf <- geojsonsf::geojson_sf(cnt)
ggplot() +
  geom_sf(data = bus_sf, pch = 1,
  colour = "#3182bd")
```



32 / 33

## Reading

- › [Get started with {rvest}](#)
- › [{httr} quickstart guide](#)

