Introduction to CSS

WEB SCRAPING IN R

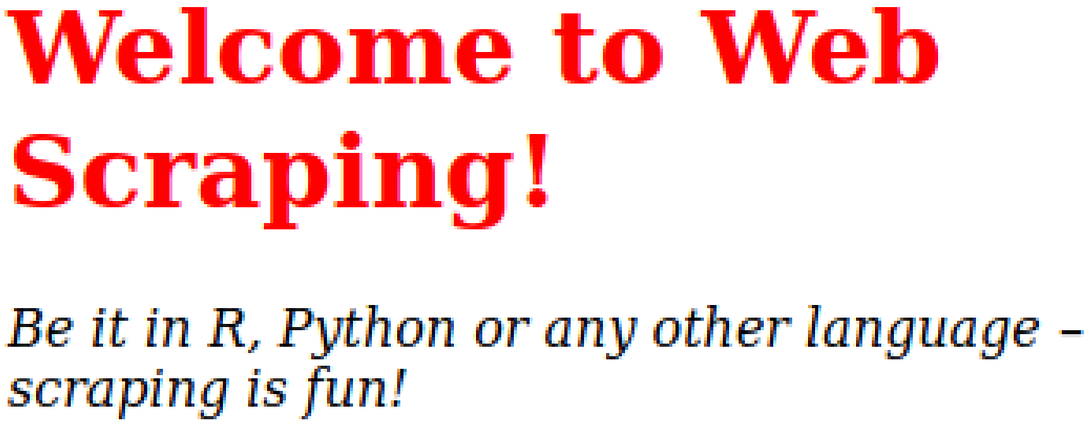
Timo Grossenbacher
Instructor

datacamp

Cascading Style Sheets

```
h1 {
    color: red;
}
p {
    font-style: italic;
}
```

```
<html>
  <body>
    <h1>Welcome to Web Scraping!</h1>
    <p>Be it in R, Python or any
      other language -
      scraping is fun!</p>
  </body>
</html>
```

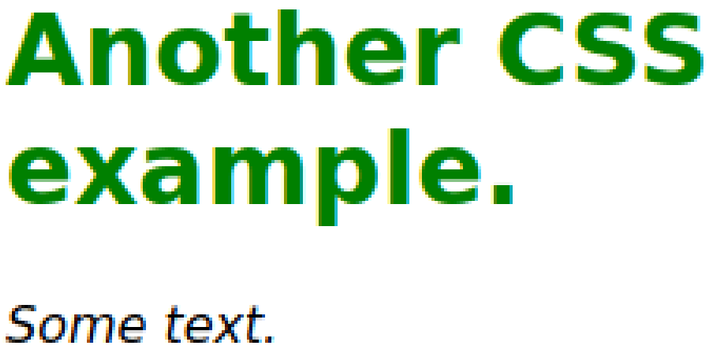


https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

CSS selectors

```
h1 {
  color: green;
}
p {
  font-style: italic;
}
h1, p {
  font-family: sans-serif;
}
```

```
...
    <h1>Another CSS example.</h1>
    <p>Some text.</p>
...
```




```
html %>% html_nodes('h1, p')
```

```
{xml_nodeset (2)}
[1] <h1> Another CSS ex...
[2] <p>Some text</p>
```

Type selectors

```
type {
    key: value;
}
html %>% html_nodes('type') # e.g. 'h1' or 'a' or 'span'
```

```
type1, type2 {
    key: value;
}
html %>% html_nodes('type1, type2')
```

```
* {
    key: value;
}
html %>% html_nodes('*')
```

Let's do this!

WEB SCRAPING IN R

CSS classes and IDs

WEB SCRAPING IN R

Timo Grossenbacher
Instructor

datacamp

Classes

```
.alert {
  color: red;
  font-weight: 800;
}
```

Some text.
Important text.
Some text with an **important link**.

```
...
<div>Some text.</div>
<div class = 'alert'>Important text.</div>
<div>
  Some text with an
  <a href = '#' class = 'alert'>important link</a>.
</div>
...
```

```
html %>% html_nodes('.alert')
```

```
{xml_nodeset (2)}
[1] <div class="alert">Important text...
[2] <a href="#" class="alert">important ...
```

Selecting multiple classes at once

```
.alert {
  color: red;
  font-weight: 800;
}
.emph {
  font-style: italic;
}
```

Some text.
Important text.
Some text with an **important link**.

```
...
<div>Some text.</div>
<div class = 'alert emph'>Important text.</div>
<div>
  Some text with an
  <a href = '#' class = 'alert'>important link</a>.
</div>
...
```

```
html %>%
    html_nodes('.alert.emph') # not: .alert, .emph
```

```
{xml_nodeset (1)}
[1] <div class="alert emph">Important text...
```

IDs

```
#special {
  color: green;
}
.alert {
  color: red;
  font-weight: 800;
}
```

Some text.
Important text.
Some text with an **important link**.

```
...
<div id = 'special'>Some text.</div>
<div class = 'alert'>Important text.</div>
<div>
  Some text with an
  <a href = '#' class = 'alert'>important link</a>.
</div>
...
```

```
html %>%
  html_nodes('#special')
```

```
{xml_nodeset (1)}
[1] <div id="special">Some text.</div>
```

Narrowing the selection down with types

```
#special {
  color: green;
}
.alert {
  color: red;
  font-weight: 800;
}
```

```
...
<div id = 'special'>Some text.</div>
<div class = 'alert'>Important text.</div>
<div>
  Some text with an
  <a href = '#' class = 'alert'>important link</a>.
</div>
...
```

```
html %>%
  html_nodes('a.alert')
```

```
{xml_nodeset (1)}
[1] <a href="#" class="alert">important ...
```

```
html %>%
  html_nodes('#special')
```

is equivalent to...

```
html %>%
  html_nodes('div#special')
```

Pseudo-classes for selecting specific children

```
li:first-child { color: blue; }

li:nth-child(2) { color: green; }

li:last-child { color: red; }
```

1. First element.
2. Second element.
3. Third element.

```
...
<ol>
  <li>First element.</li>
  <li>Second element.</li>
  <li>Third element.</li>
</ol>
...
```

```
html %>% html_nodes('li:last-child')
    # or html_nodes('li:nth-child(3)')
```

```
{xml_nodeset (1)}
[1] <li>Third element.</li>
```

[1] https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

To sum it up...

Selector type	HTML	CSS selector
Type	`<p>...</p>`	`p`
Multiple types	`<p>...</p><div>...</div>`	`p, div`
Class	`<p class = 'x'>...</p>`	`.x`
Multiple classes	`<p class = 'x y'>...</p>`	`.x.y`
Type + Class	`<p class = 'x'>...</p>`	`p.x`
ID	`<p id = 'x'>...</p>`	`#x`
Type + Pseudo-class	`<p>...</p><p>...</p>`	`p:first-child`

Let's practice!

WEB SCRAPING IN R

CSS combinators

WEB SCRAPING IN R

Timo Grossenbacher
Instructor

datacamp

There are four different combinators

Structure: `h2#someid {space|>|+|~} .someclass`

`space` : Descendant combinator

`>` : Child combinator

`+` : Adjacent sibling combinator

`~` : General sibling combinator

The descendant and child combinators

```
<html>
  <body>
    <div class = 'first'>
      <a>A link.</a>
      <p>The first paragraph with
        <a>another link</a>.
      </p>
    </div>
    <div>
      Not an actual paragraph,
      but with a <a href="#">link</a>.
    </div>
  </body>
</html>
```

```
html %>%
    html_nodes('div.first a')
```

```
{xml_nodeset (2)}
[1] <a>A link.</a>
[2] <a>another link</a>
```

```
html %>%
    html_nodes('div.first > a')
```

```
{xml_nodeset (1)}
[1] <a>A link.</a>
```

The sibling combinators

```
<html>
  <body>
    <div class = 'first'>
      <a>A link.</a>
      <p>The first paragraph with
        <a>another link</a>.
      </p>
    </div>
    <div>
      Not an actual paragraph,
      but with a <a href="#">link</a>.
    </div>
    <p>A paragraph.</p>
  </body>
</html>
```

```
html %>% html_nodes('div.first + div')
```

```
{xml_nodeset (1)}
[1] <div>\n Not an actual...
```

```
html %>% html_nodes('div.first ~ div')
```

```
{xml_nodeset (1)}
[1] <div>\n Not an actual...
```

```
html %>% html_nodes('div.first ~ *')
```

```
{xml_nodeset (2)}
[1] <div>\n Not an actual... [2] <p>A paragraph...
```

When combinators come in handy

```
...
    <div id = 'start'>
      <h1 class = 'first'>First</h1>
    </div>
    <div id = 'end'>
      <p class = 'text1'>Some text.</p>
      <p class = 'text2'>More text.</p>
    </div>
...
```

```
...
    <div>
      <h1>First</h1>
    </div>
    <div>
      <p>Some text.</p>
      <p>More text.</p>
    </div>
...
```

```
html %>% html_nodes('.text2')
```

```
html %>% html_nodes('p + p')
```

```
{xml_nodeset (1)}
[1] <p class="text2">More text.</p>
```

```
{xml_nodeset (1)}
[1] <p>More text.</p>
```

Let's practice!

WEB SCRAPING IN R