Day 04: Scraping Part 1

Erin Rossiter

13 February, 2023

Announcements

Announcements

- PS02 graded
  - » grades in Canvas; comments on Rmd files
  - » please review any comments
- PS03 due tomorrow
  - » includes a question on final projects
- Project discussion
  - » questions for class?
  - » remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Announcements

- PS02 graded
  - grades in Canvas; comments on Rmd files
  - please review any comments
- PS03 due tomorrow
  - includes a question on final projects
- Project discussion
  - questions for class?
  - remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Announcements

- PS02 graded
  - » grades in Canvas; comments on Rmd files
  - » please review any comments
- PS03 due tomorrow
  - » includes a question on final projects
- Project discussion
  - » questions for class?
  - » remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Announcements

- PS02 graded
  - » grades in Canvas; comments on Rmd files
  - » please review any comments
- PS03 due tomorrow
  - » includes a question on final projects
- Project discussion
  - » questions for class?
  - » remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Announcements

- PS02 graded
  - » grades in Canvas; comments on Rmd files
  - » please review any comments
- PS03 due tomorrow
  - » includes a question on final projects
- Project discussion
  - » questions for class?
  - » remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Announcements

- PS02 graded
  - » grades in Canvas; comments on Rmd files
  - » please review any comments
- PS03 due tomorrow
  - » includes a question on final projects
- Project discussion
  - » questions for class?
  - » remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Announcements

- PS02 graded
  - » grades in Canvas; comments on Rmd files
  - » please review any comments
- PS03 due tomorrow
  - » includes a question on final projects
- Project discussion
  - » questions for class?
  - » remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Announcements

- PS02 graded
  - » grades in Canvas; comments on Rmd files
  - » please review any comments
- PS03 due tomorrow
  - » includes a question on final projects
- Project discussion
  - » questions for class?
  - » remember a big part of the project is sharing code, building community, let's start by sharing our ideas
    - 30 section description to class

Today's plan

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - » if else
  - » for loops
  - » advanced functions
  - » we've covered the basic ingredients!
- Today
  - » big picture for next two classes
  - » scraping!
  - » our first "non-basics" session!
  - » don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - » if else
  - » for loops
  - » advanced functions
  - » we've covered the basic ingredients!
- Today
  - » big picture for next two classes
  - » scraping!
  - » our first "non-basics" session!
  - » don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Today's plan

- Last time
  - if else
  - for loops
  - advanced functions
  - we've covered the basic ingredients!
- Today
  - big picture for next two classes
  - scraping!
  - our first "non-basics" session!
  - don't worry, we will always be honing these skills (loops, ifelse logic, functions, etc.)
- Save 15 minutes for PS03 questions/hints

Big picture

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - » More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - » Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - » More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - » Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - » More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - » Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - » More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - » Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Big picture

- You have incredible opportunity for easy access to tons of untapped data!
  - » More now than ever with...
    - increasing storage and computational power, even on your personal laptop
    - increasing digitization of... everything
- The goal is to be able to size-up any scraping-related data collection you run into
- But is this legal?
  - » Saving this topic for when we discuss APIs
- Today, legal scraping of publicly available info

Roadmap

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Roadmap

- Three sessions on "webscraping"
  - Writing and reading data from R
  - Understanding the web! (navigating URLs)
  - Understanding HTML content of a page
  - Selenium
  - APIs (and JSON-format and http requests)
  - web-crawlers (if time)
- Caveat
  - I'm not a computer scientist or web developer
  - We're covering the basics so we can know what is possible and overcome roadblocks

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

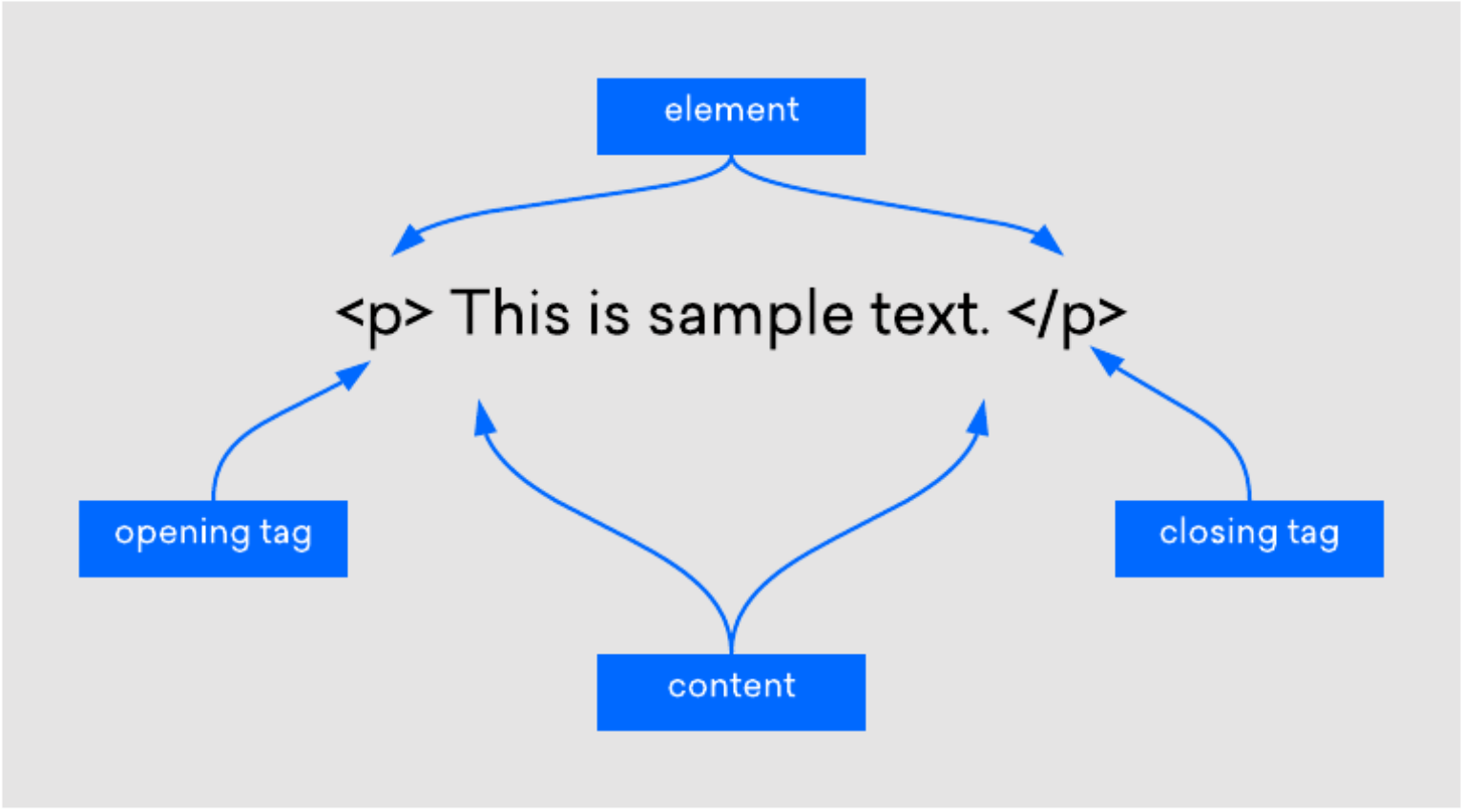
Understanding the language of the web

- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

Understanding the language of the web

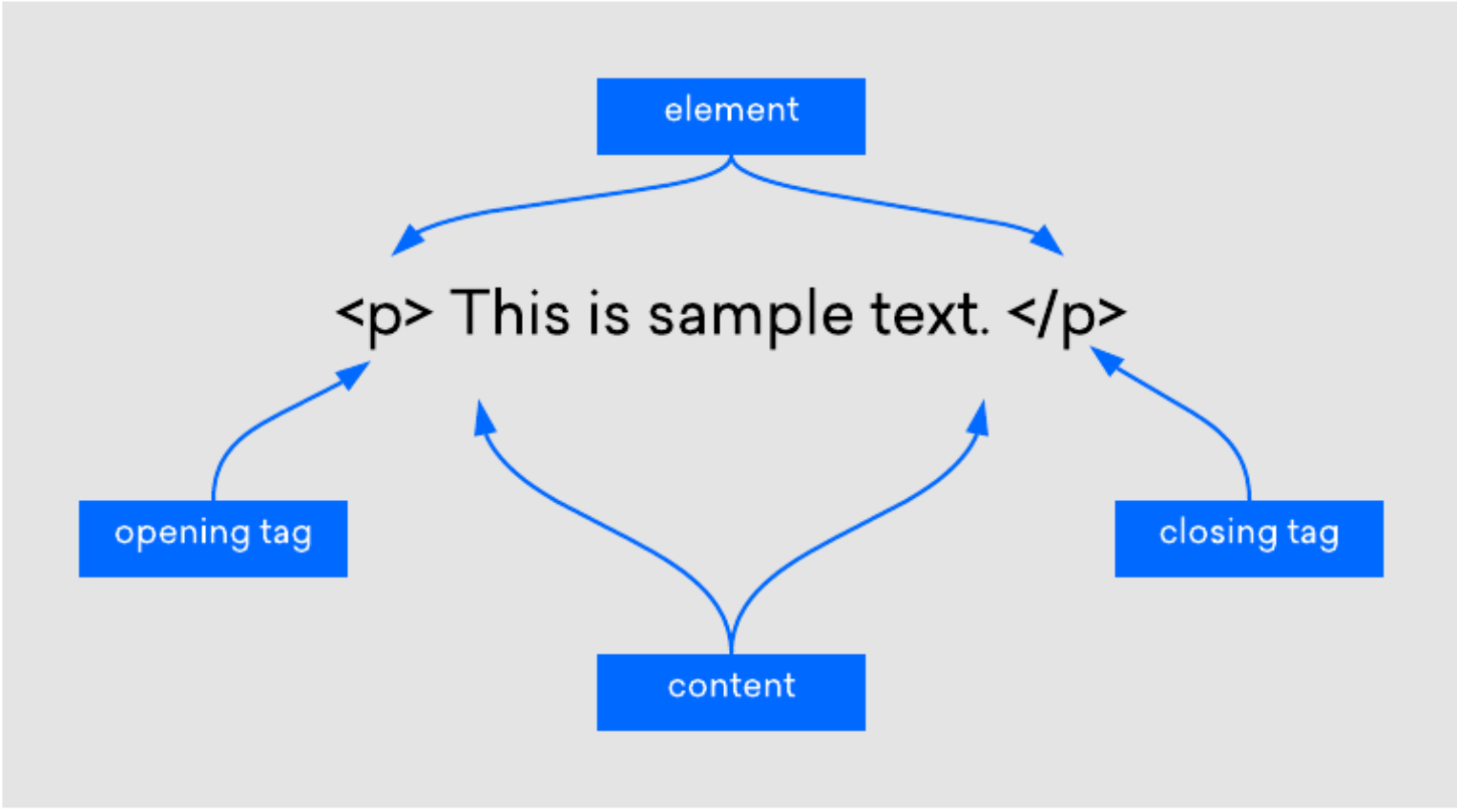
- HTML is the technical representation of that webpage and tells your browser which elements to display and how to display them
  - basic building blocks of the web
  - like how Rmd has its own syntax, so does HTML in order to render "pretty"
  - Ex: we use the ## to make headings in Rmd, we use `<h1>` to make headings in HTML
- Some examples of "tags":
  - `<h1>, <h2>, ..., <h6>` Headings
  - `<p>` Paragraph elements
  - `<ul>` Unordered List
  - `<div>` Division / Section
  - `<table>` Tables
  - `<form>` Web forms

“tags”



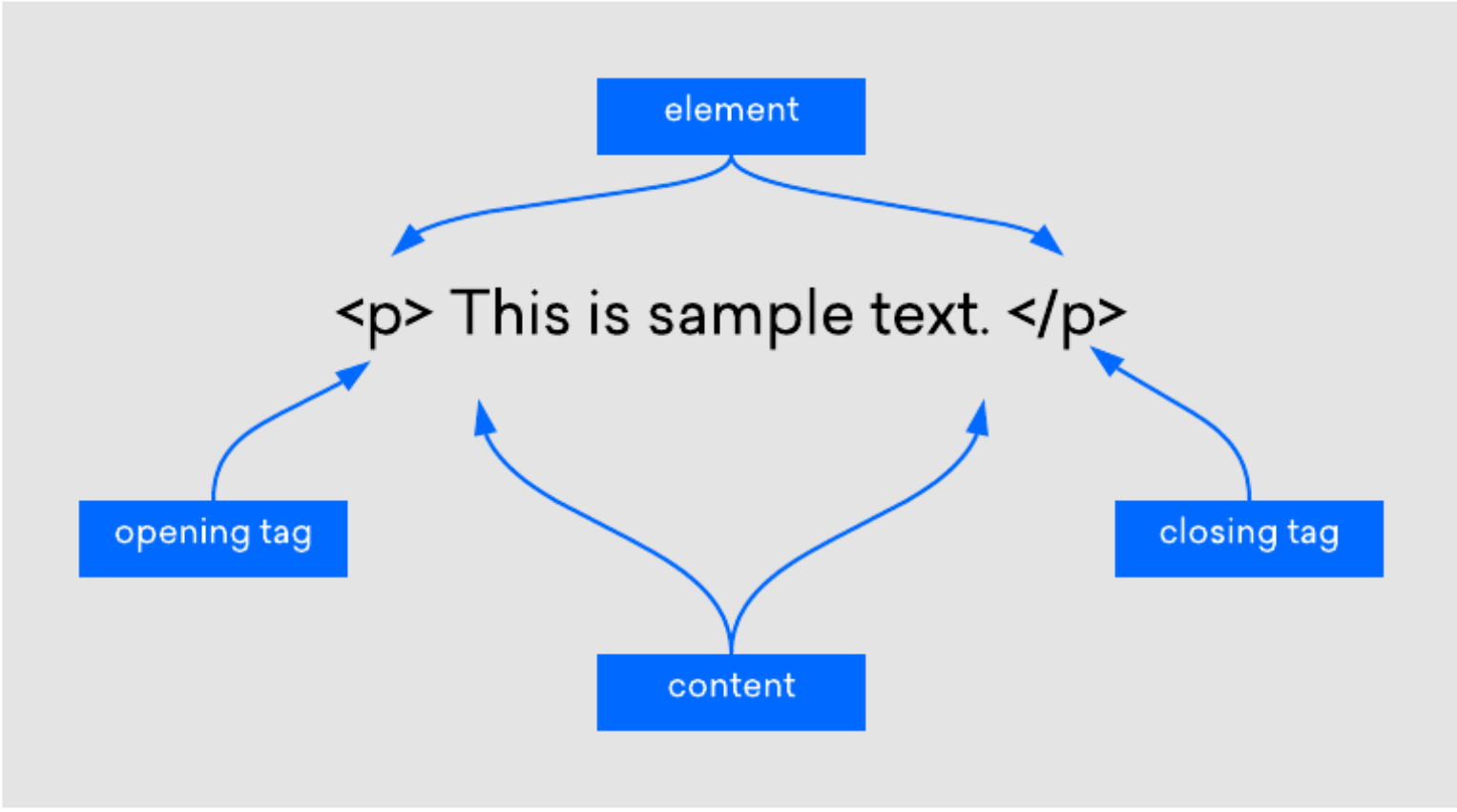
- An HTML “tag” is a piece of markup language used to indicate the beginning and end of an “element”
- Help web browsers convert HTML documents into web pages

“tags”



- An HTML “tag” is a piece of markup language used to indicate the beginning and end of an “element”
- Help web browsers convert HTML documents into web pages

“tags”



- An HTML “tag” is a piece of markup language used to indicate the beginning and end of an “element”
- Help web browsers convert HTML documents into web pages

Tags can also have “attributes”

- Attributes are additional information for how the element should be displayed
- Examples:
  - `<p style="color:blue">Blue text</>`
  - `<a href="https://www.nd.edu/">Notre Dame Website</a>`

Tags can also have “attributes”

- Attributes are additional information for how the element should be displayed
- Examples:
  - `<p style="color:blue">Blue text</>`
  - `<a href="https://www.nd.edu/">Notre Dame Website</a>`

Tags can also have “attributes”

- Attributes are additional information for how the element should be displayed
- Examples:
  - `<p style="color:blue">Blue text</>`
  - `<a href="https://www.nd.edu/">Notre Dame Website</a>`

Tags can also have “attributes”

- Attributes are additional information for how the element should be displayed
- Examples:
  - `<p style="color:blue">Blue text</>`
  - `<a href="https://www.nd.edu/">Notre Dame Website</a>`

HTML document basic structure

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
   </body>
</html>
```

- Not always consistent across websites, but often consistent across pages on the same website.
  - » This is good news for us – lots of structure!!

Running example 1

ND Political Science faculty

- Today we're going to understand how this website is structured and scrape it

How can we view the HTML of a website? Google chrome

Open an "un-rendered" version of the website in a new tab:

- View > Developer tools > View Source
- Right click > View Page Source

Inspect the HTML of a specific element on the page:

- Right click > Inspect Element

How can we view the HTML of a website? Google chrome

Open an "un-rendered" version of the website in a new tab:

- View > Developer tools > View Source
- Right click > View Page Source

Inspect the HTML of a specific element on the page:

- Right click > Inspect Element

How can we view the HTML of a website? Google chrome

Open an "un-rendered" version of the website in a new tab:

- View > Developer tools > View Source
- Right click > View Page Source

Inspect the HTML of a specific element on the page:

- Right click > Inspect Element

How can we view the HTML of a website? Google chrome

Open an "un-rendered" version of the website in a new tab:

- View > Developer tools > View Source
- Right click > View Page Source

Inspect the HTML of a specific element on the page:

- Right click > Inspect Element

How can we view the HTML of a website? Google chrome

Open an "un-rendered" version of the website in a new tab:

- View > Developer tools > View Source
- Right click > View Page Source

Inspect the HTML of a specific element on the page:

- Right click > Inspect Element

Try it out

- What type of tag is the “Core Faculty” element?
- What type of tag is each professor’s name?
- what type of tag is each professor’s field?

Try it out

- What type of tag is the “Core Faculty” element?
- What type of tag is each professor’s name?
- what type of tag is each professor’s field?

Try it out

- What type of tag is the “Core Faculty” element?
- What type of tag is each professor’s name?
- what type of tag is each professor’s field?

Urls

- So far we've seen the *pages* have great structure.
- Why? Websites aren't usually written by people, but by algorithms.
- For the same reason, *urls* have great structure, so easy to navigate across pages
- What's happening here?
  - https://politicalscience.nd.edu/people/core-faculty/
  - https://politicalscience.nd.edu/people/affiliated-faculty-and-visiting-researchers/
  - https://politicalscience.nd.edu/people/emeritus/
  - ...

Urls

- So far we've seen the *pages* have great structure.
- Why? Websites aren't usually written by people, but by algorithms.
- For the same reason, *urls* have great structure, so easy to navigate across pages
- What's happening here?
  - https://politicalscience.nd.edu/people/core-faculty/
  - https://politicalscience.nd.edu/people/affiliated-faculty-and-visiting-researchers/
  - https://politicalscience.nd.edu/people/emeritus/
  - ...

Urls

- So far we've seen the *pages* have great structure.
- Why? Websites aren't usually written by people, but by algorithms.
- For the same reason, *urls* have great structure, so easy to navigate across pages
- What's happening here?
  - https://politicalscience.nd.edu/people/core-faculty/
  - https://politicalscience.nd.edu/people/affiliated-faculty-and-visiting-researchers/
  - https://politicalscience.nd.edu/people/emeritus/
  - ...

Urls

- So far we've seen the *pages* have great structure.
- Why? Websites aren't usually written by people, but by algorithms.
- For the same reason, *urls* have great structure, so easy to navigate across pages
- What's happening here?
  - https://politicalscience.nd.edu/people/core-faculty/
  - https://politicalscience.nd.edu/people/affiliated-faculty-and-visiting-researchers/
  - https://politicalscience.nd.edu/people/emeritus/
  - ...

Urls

- So far we've seen the *pages* have great structure.
- Why? Websites aren't usually written by people, but by algorithms.
- For the same reason, *urls* have great structure, so easy to navigate across pages
- What's happening here?
  - https://politicalscience.nd.edu/people/core-faculty/
  - https://politicalscience.nd.edu/people/affiliated-faculty-and-visiting-researchers/
  - https://politicalscience.nd.edu/people/emeritus/
  - ...

Url example 2

https://www.cnn.com/election/2018/results/nevada/senate
https://www.cnn.com/election/2018/results/tennessee/senate
https://www.cnn.com/election/2018/results/tennessee/house

Main point: the structure of the URLs allows me to simply substitute each state to navigate to the state results, makes scraping info *across* pages easy

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks

In Sum

If we:

- know the basic structure of the html content of a page . . .
- and understand how the urls are set up to navigate . . .
- we can pretty easily write a script to tear down large parts of a website.

This can be shockingly easy for the simplest websites!

Complexity is introduced if:

- you do not have a pre-defined set of urls
- what you want is burried not in the html but in javascript (or something else)
- they only want you (or allow you) to pull data from their API

More on these issues in the coming weeks