Web Scraping & APIs

Nel Escher

many slides lifted from EECS 485 lectures
thank u bbs

Agenda

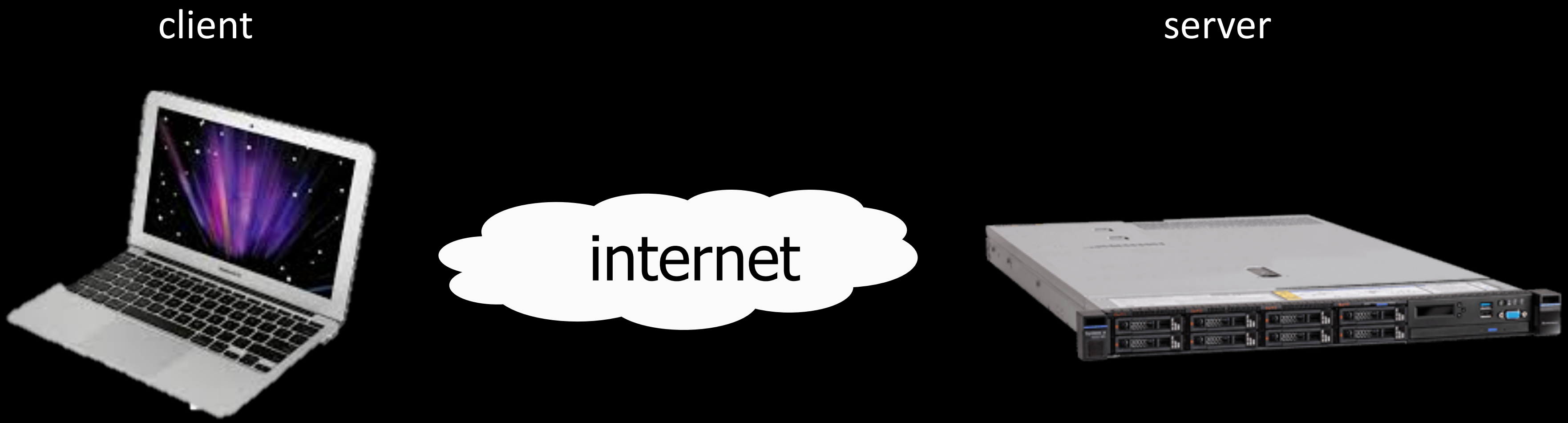
- Web sites
- Requests
- Scraping
- APIs
- API Wrappers

What is the internet?

The request response cycle

- The request response cycle is how two computers communicate with each other on the web

1. A client requests some data
2. A server responds to the request



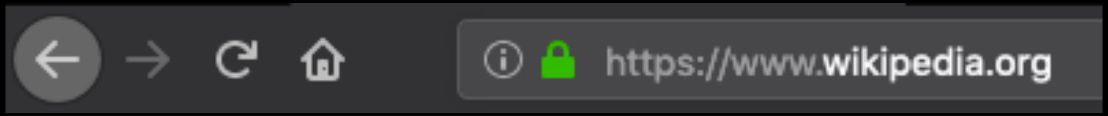
client

internet

server

The request response cycle

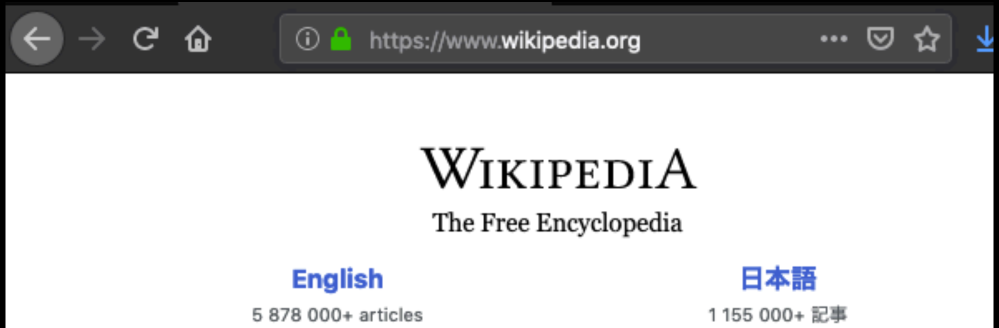
- A client (YOU) requests a web page



- A server responds with an HTML file
  - The content might be created dynamically

```
<!DOCTYPE html>
...
```

- The client browser renders the HTML



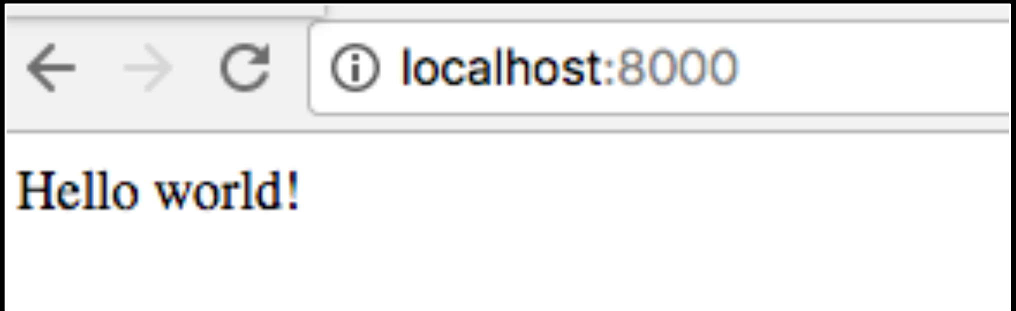
What does a server respond with?

- A server might respond with different kinds of files. Common examples:
  - HTML
  - CSS
  - JavaScript

HTML

- HTML describes the content on a page
- Example `index.html`

```
<!DOCTYPE html>
<html lang="en">
  <body>
    Hello world!
  </body>
</html>
```

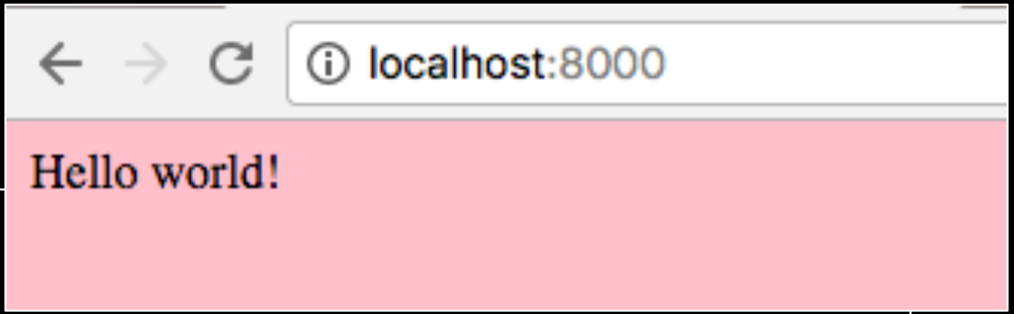


Hello world!

CSS

- CSS describes the layout or style of a page.
- Link to CSS in HTML
- Example `style.css`

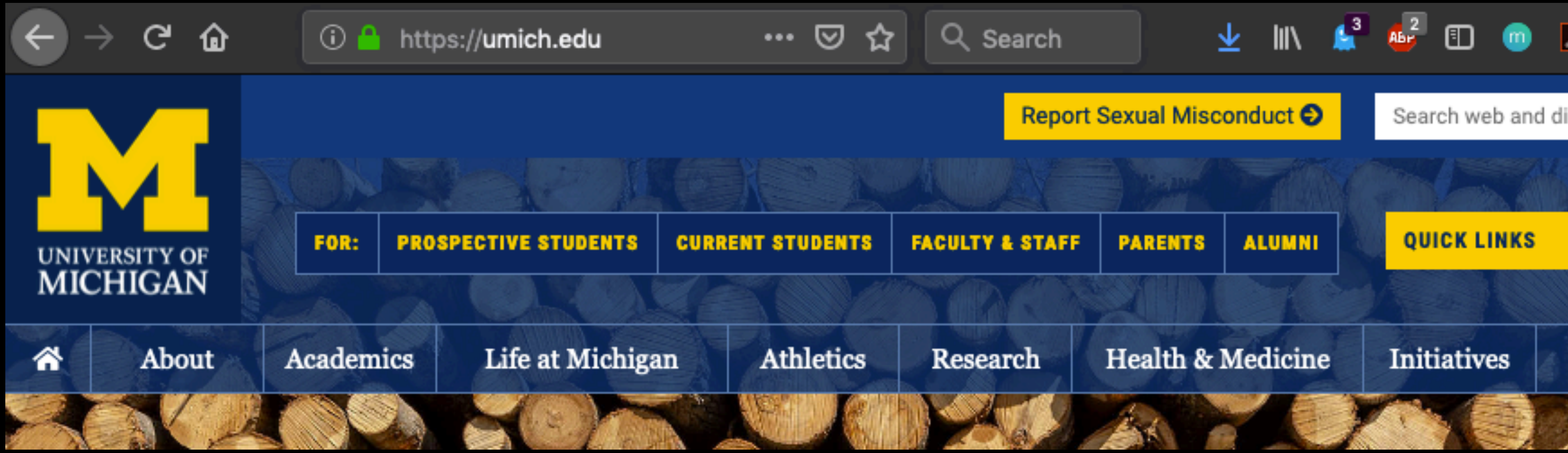
```
body {
    background: pink;
}
```



Hello world!

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" type="text/css" href="/style.css">
  </head>
  <body>
    Hello world!
  </body>
</html>
```

Example



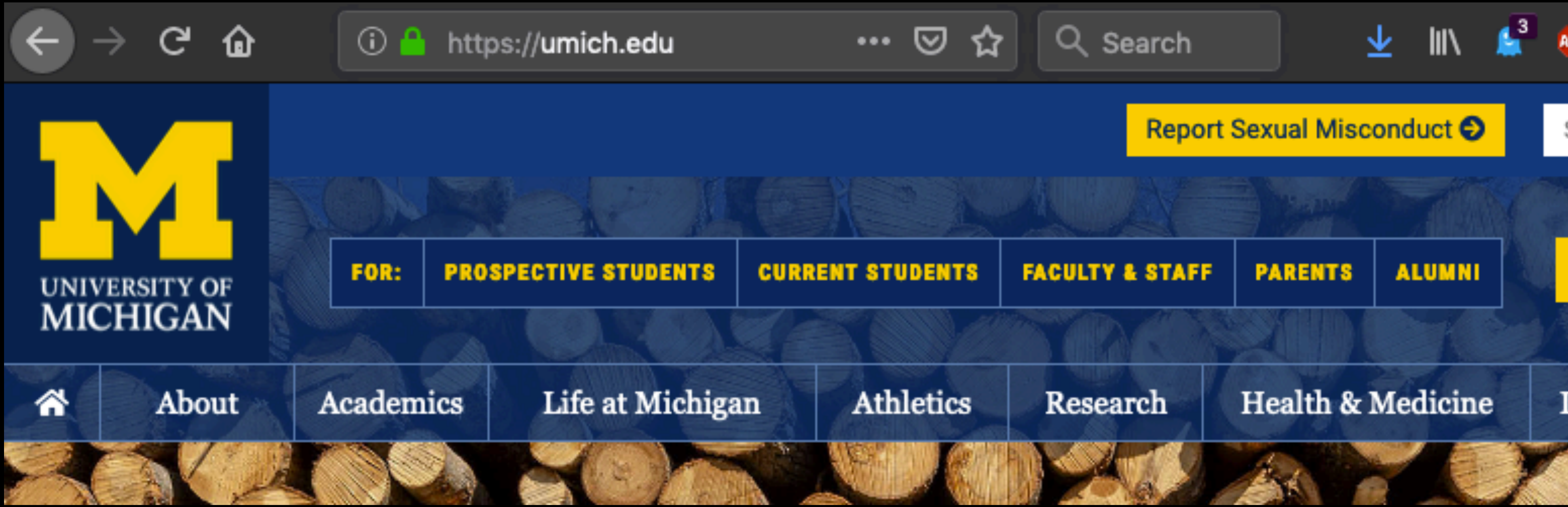
- Add **tags** as "mark up" to text
- Document still "primarily" text

```
<html>
  <head></head>
  <body>
    <nav>
    <ul>
      <li><a href="">About</a></li>
      <li><a href="">Academics</a></li>
      <li><a href="">Life at Michigan</a></li>
      <li><a href="">Athletics</a></li>
      <li><a href="">Research</a></li>
      <li><a href=“">Health & Medicine</a></li>
    </ul>
    </nav>
  </body>
</html>
```

Hypertext

- Text with embedded links to other documents.
- Anchor tag

```
<a href="https://umich.edu/about/">
  About
</a>
```

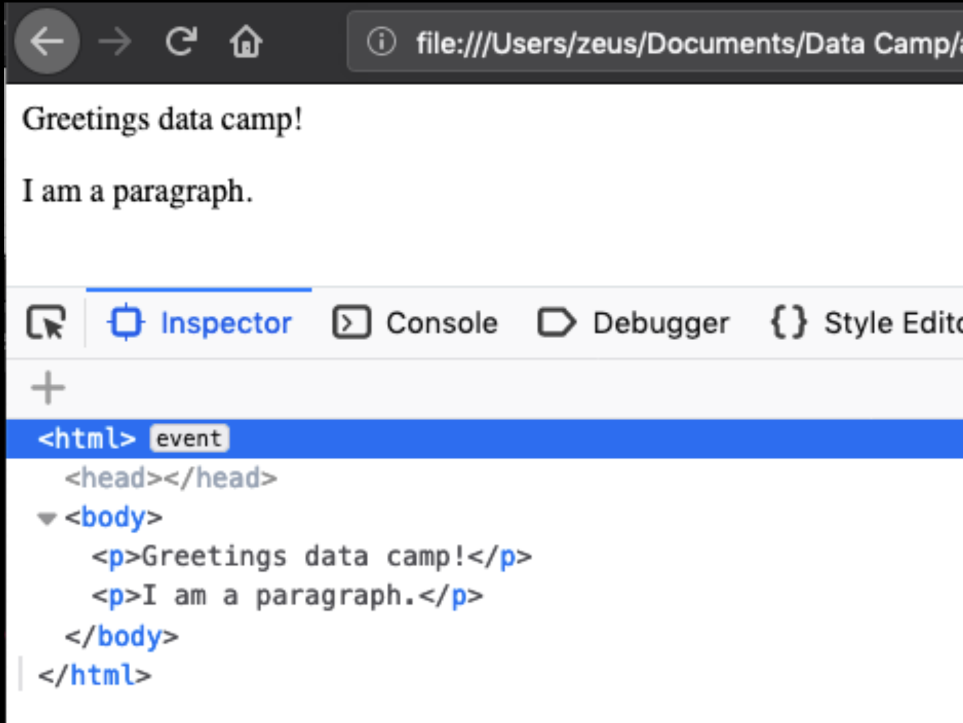


Document Object Model (DOM)

- HTML tags form a tree

```
<html>
  <head></head>
  <body>
    <p>Greetings data camp!</p>
    <p>I am a paragraph.</p>
  </body>
</html>
```

- This tree is called the Document Object Model (DOM)
- Inspect the DOM with
  - Chrome developer tools
  - Firefox developer tools



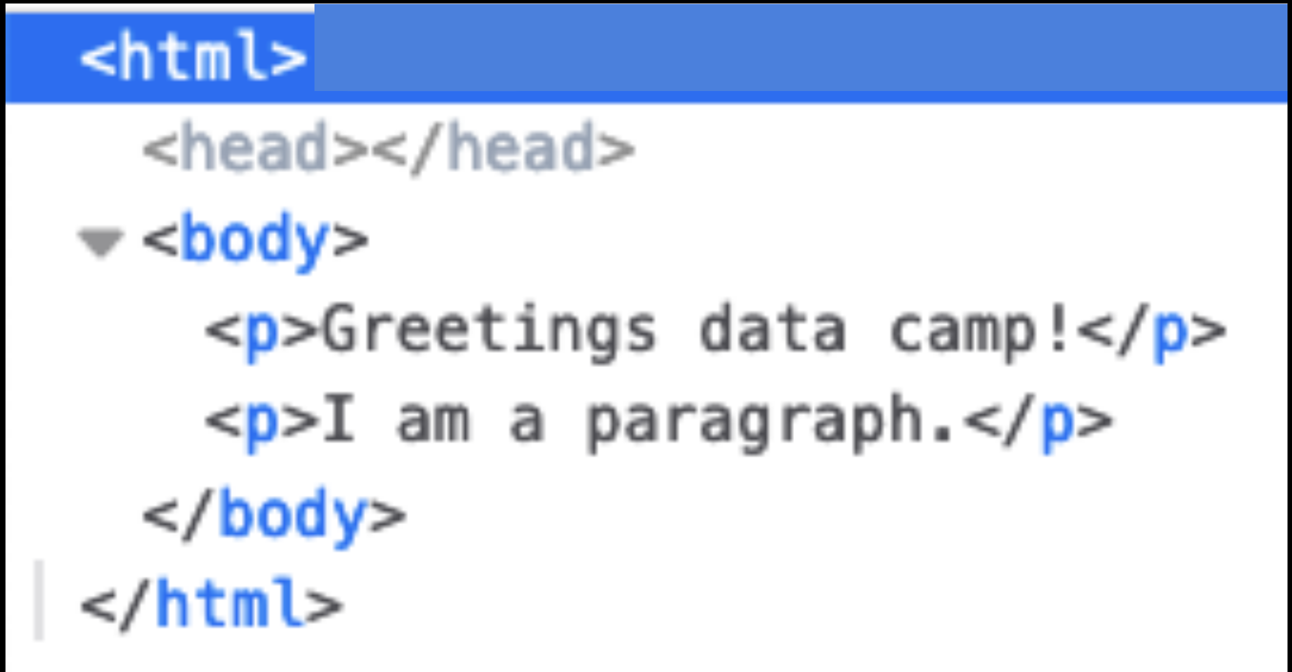
Greetings data camp!

I am a paragraph.

```
<html>
  <head></head>
  <body>
    <p>Greetings data camp!</p>
    <p>I am a paragraph.</p>
  </body>
</html>
```

Document Object Model (DOM)

- The DOM is a data structure built from the HTML
- In the DOM, everything is a **node**
  - All HTML elements are element nodes
  - Text inside HTML elements are text nodes



```
<html>
  <head></head>
  <body>
    <p>Greetings data camp!</p>
    <p>I am a paragraph.</p>
  </body>
</html>
```

What is a scraping a website?

- Extracting data from a website
  - Get the files for the website from a server
  - Parse those files
  - If needed, go back for more files

TO JUPYTER!

Scraping

- Scripts can be brittle
  - If someone were to edit the Wiki page and add another table, my code would break ☹
- Have to hack through a lot of garbage
- Not terrible if it’s all you have to work with

APIs

- Application Programming Interface
- Makes data available for use by different apps
- Help us get the data we want

API Endpoints

Access data by asking for particular URL paths

- Like file paths on yr computer

- https://api.coindesk.com/v1/bpi/currentprice.json

- Sample JSON Response:
  - {"time":{"updated":"Jun 18, 2019 15:33:00 UTC","updatedISO":"2019-06-18T15:33:00+00:00","updateduk":"Jun 18, 2019 at 16:33 BST"},"disclaimer":"This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD currency data converted using hourly conversion rate from openexchangerates.org","chartName":"Bitcoin","bpi":{"USD":{"code":"USD","symbol":"&#36;","rate":"8,977.3100","description":"United States Dollar","rate_float":8977.31},"GBP":{"code":"GBP","symbol":"&pound;","rate":"7,157.6362","description":"British Pound Sterling","rate_float":7157.6362},"EUR":{"code":"EUR","symbol":"&euro;","rate":"8,025.3830","description":"Euro","rate_float":8025.383}}}

API Endpoints

- We can hit these endpoints in our browser and see the data that is returned
- Use a Python library to fetch the same data from the same URLs for use in our programs
- If you’re first learning, try your URL in the browser first!

- Web Scraping

- APIs

Very convenient,
but if you want rings, you’ll have to cut it yourself

REST API verbs

- GET: return datum
- PUT: replace the entire datum
- PATCH: update part of a datum
- POST: create new datum
- DELETE: delete datum

REST API status codes

- 200 OK
- 201 Created
  - Successful creation after POST
- 204 No Content
  - Successful DELETE
- 304 Not Modified
  - Used for conditional GET calls to reduce band-width usage
  - Include Date header
- 400 Bad Request
  - General error
  - Domain validation errors, missing data, etc.

Public APIs

- GitHub
  https://developer.github.com/v3/
- LinkedIn
  https://developer.linkedin.com/
- Facebook
  https://developers.facebook.com/docs/graph-api
- Twitter
  https://dev.twitter.com/rest/public

JSON structures

- Object (key/value pairs) or array (list of values)

```
{
        “name” : “Nel”,
        “num_feet”: 4
}

[“Bifur”, “Bofur”, “Bombur” ]
```

- The values can be of different types:
  - string
  - number
  - `true`
  - `false`
  - `null`
  - Object
  - Array

JSON

- JSON: JavaScript Object Notation
- Lightweight data-interchange format
- Based on JavaScript syntax
    - Uses conventions familiar to programmers in many languages
- Commonly used to send data from a server to a web client
    - Client parses JSON using JavaScript and displays content
- Ubiquitous with REST APIs

API Documentation

- Read it.
- Different resources are located at different paths
- Documentation tells you what data is returned at specific paths

```
GET https://api.spotify.com/v1/albums/{id}
GET https://api.spotify.com/v1/artists/{id}/top-tracks
```

https://developer.spotify.com/documentation/web-api/reference/

Authentication

- Sometimes you will have to get keys or tokens and submit them along with your requests
- This helps prevent abuse of web resources
- Instructions are usually clear; often require you to sign up for an account

Rate Limiting

- Apps often ask you to restrict your request rate (e.g. 100 requests/min)
- If you exceed this threshold, the app can slow down your subsequent requests
- Take it slow :)

Most of programming is knowing what to Google

- APIs

- API Wrapper