Data Visualization in Web Browsers with JavaScript APIs

Joshua A. Lemli

August 2018

About the speaker

(Speaker not pictured)

About this session

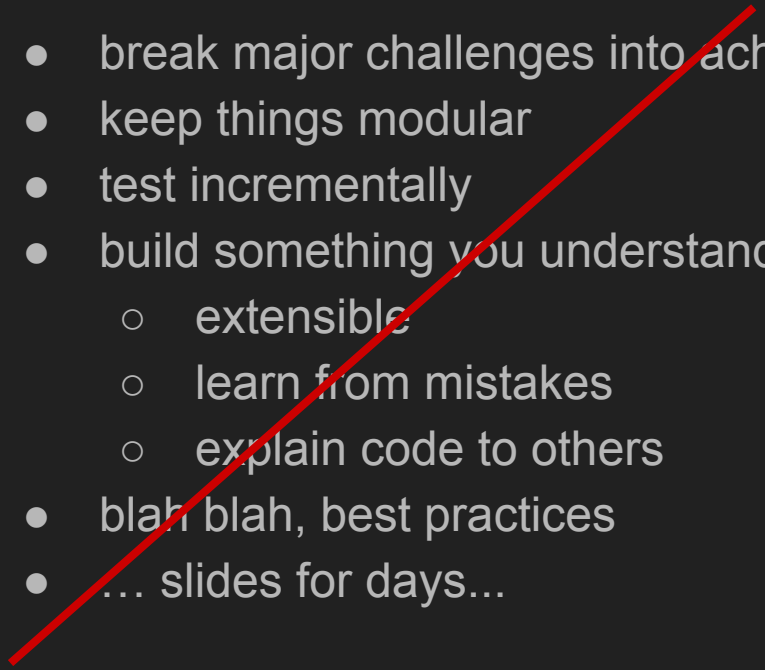
I. Important concepts
II. Overview of (JS) APIs
III. Data considerations
IV. Code 
V. Closing remarks and Q&A

About... YOU!

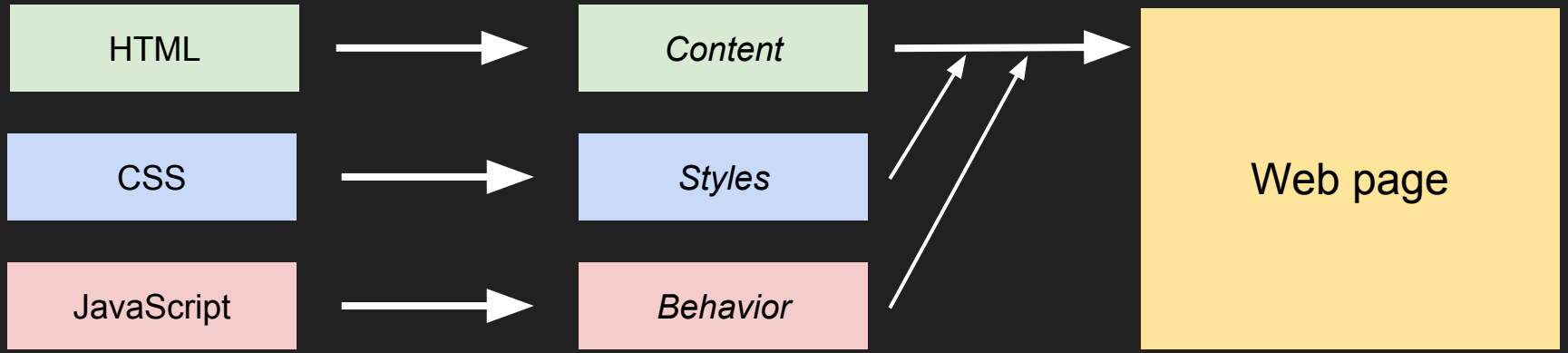
What do you hope to get out of this session?

Normally... I wax programmer

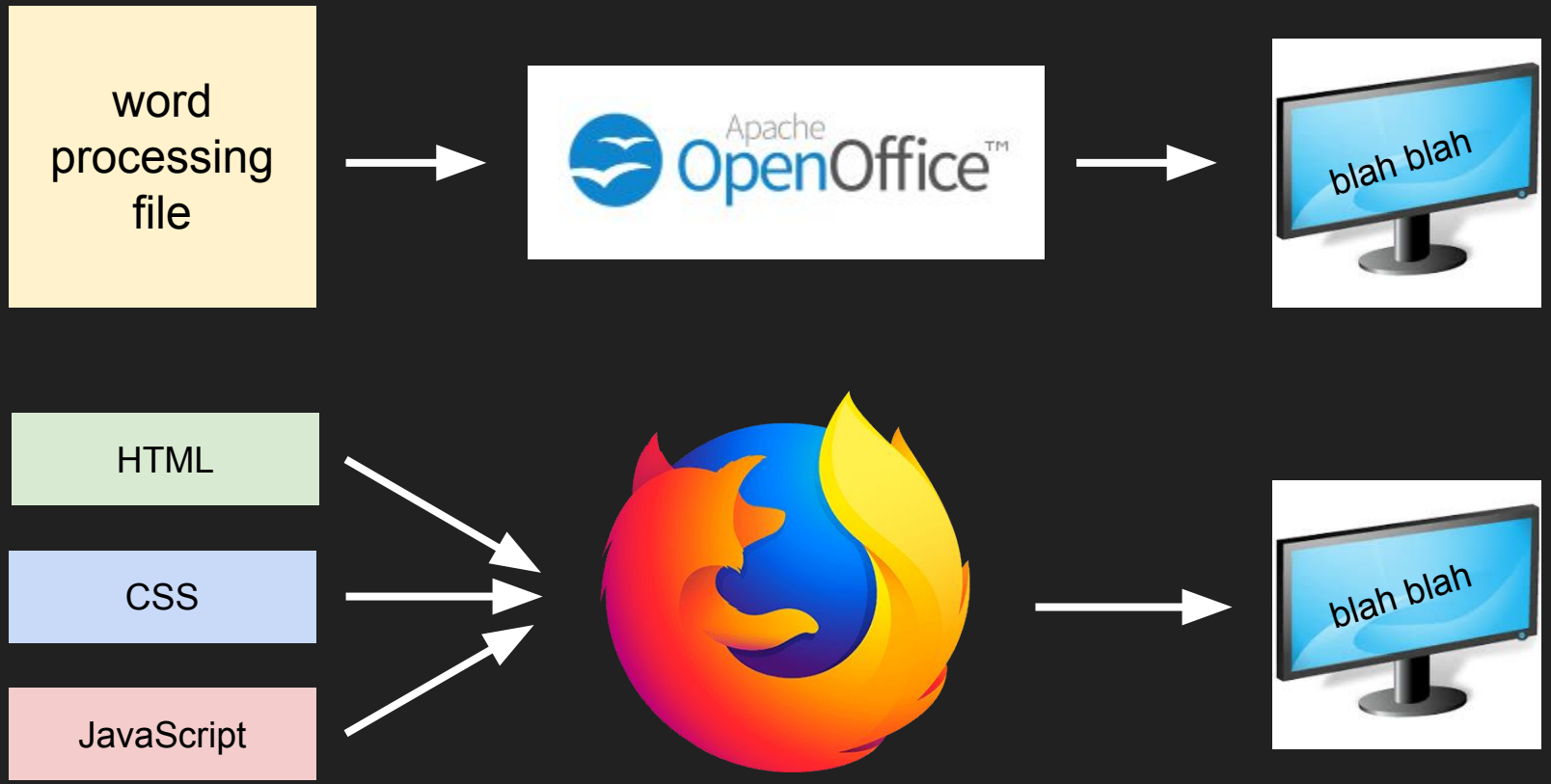
- break major challenges into achievable tasks
- keep things modular
- test incrementally
- build something you understand
  - extensible
  - learn from mistakes
  - explain code to others
- blah blah, best practices
- … slides for days...



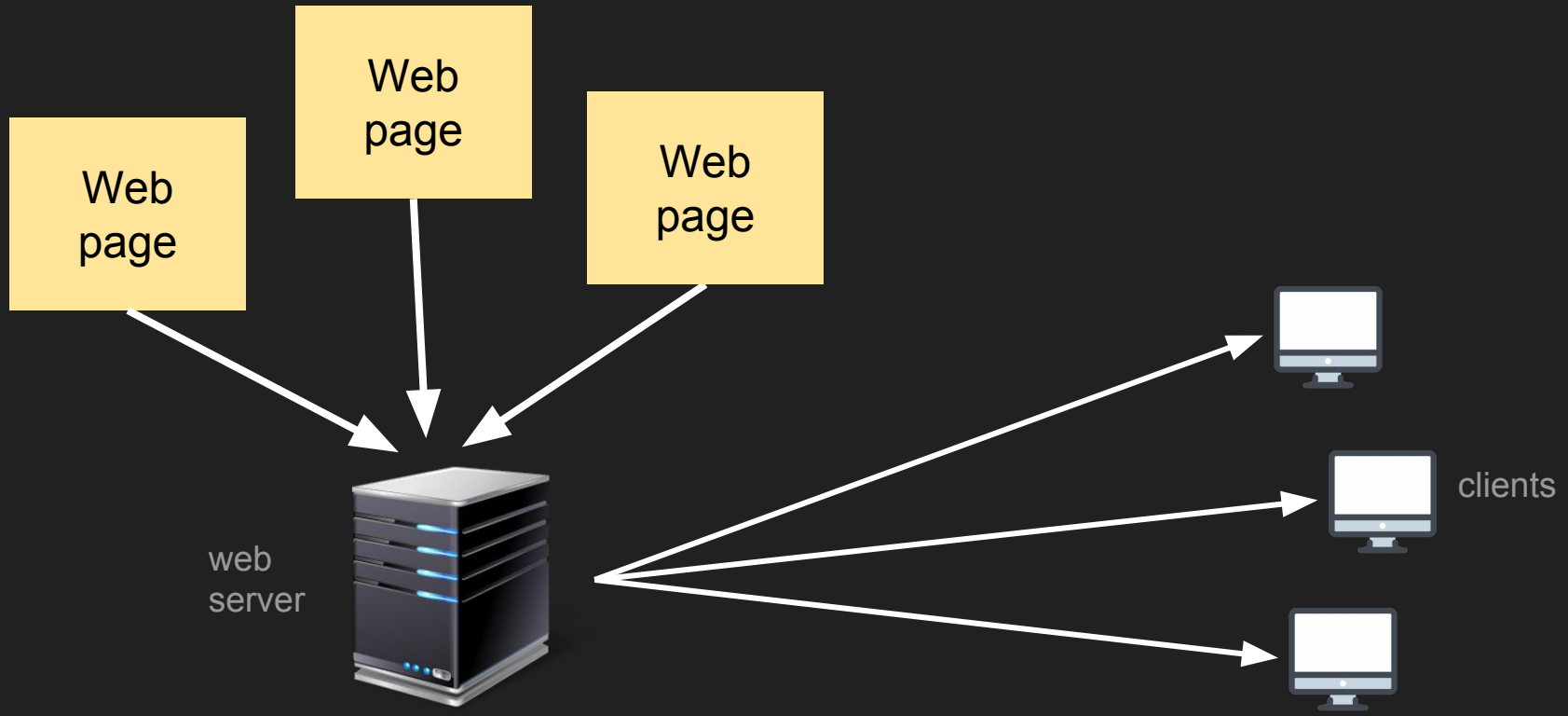
"Web page"



Web browsers



"Website"



Accessing your site

https://<domain>

Web page

web server

client

file:///<local_path>

Web page

Major modern web browsers

"Modern" JavaScript (ECMAScript)

Arrow Functions ("lambda")

```
function(x) {
    return x + 5
}
```

```
x => x + 5
```

Sets and Maps

```
if (myArr.indexOf(myVal) === -1) {
    myArr.push(myVal)
}
```

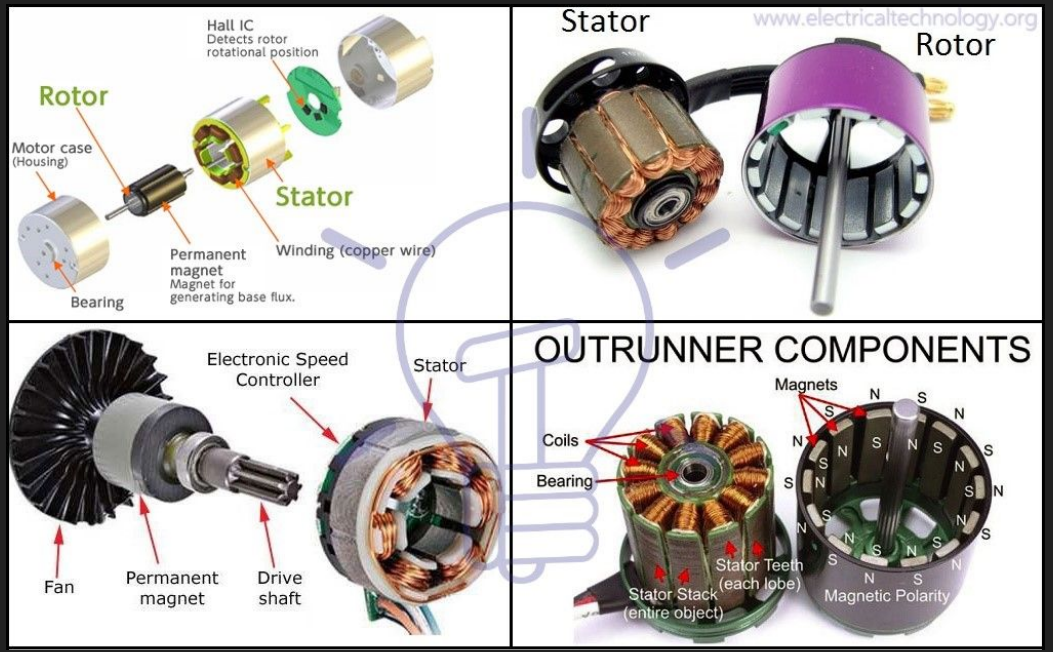
```
mySet.add(myVal)
```

Canvas API, Fetch API, Array.find, etc...

Application Programming Interface (API)

API

"roll your own"

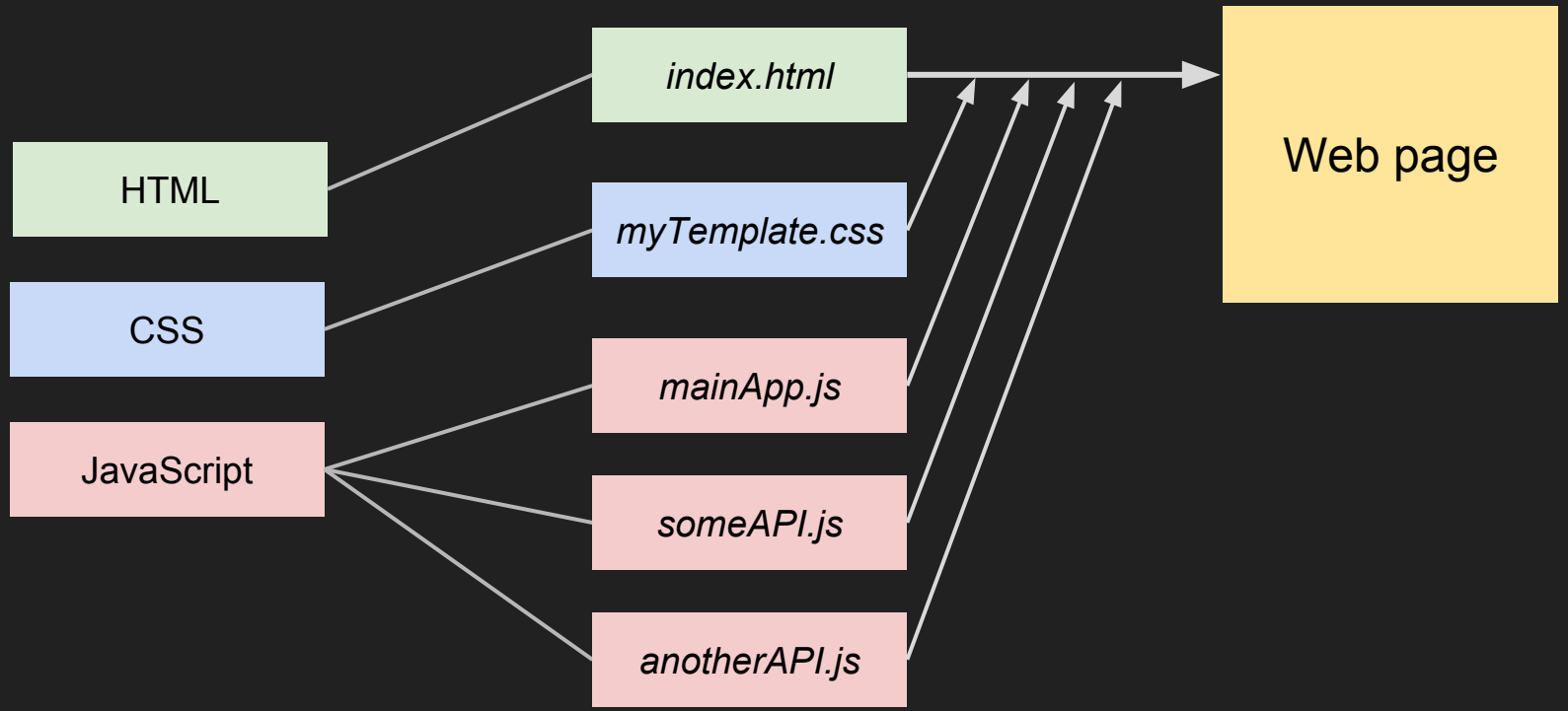


Application Programming Interface (API)

API

≠ application

JS APIs in the browser



Selecting appropriate APIs

Representation of the API ecosystem

… API overload!

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

JS Graphics APIs

d3.js

plotly.js

Canvas

Google Charts

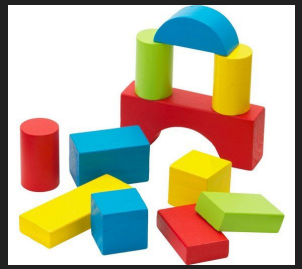
charts.js

Canvas

- low-level
- open-ended
- very powerful

Google Charts

- easy to use
- simple syntax
- limited functionality



charts.js

- robust
- fairly simple syntax
- built on Canvas API (fast!)

d3.js

- immersive
- non-traditional
- powerful and complex
- built on HTML, CSS, and SVG

plotly.js

- simple
- graph-oriented
- robust (built with d3)
- good out-of-the-box features

JS Map APIs

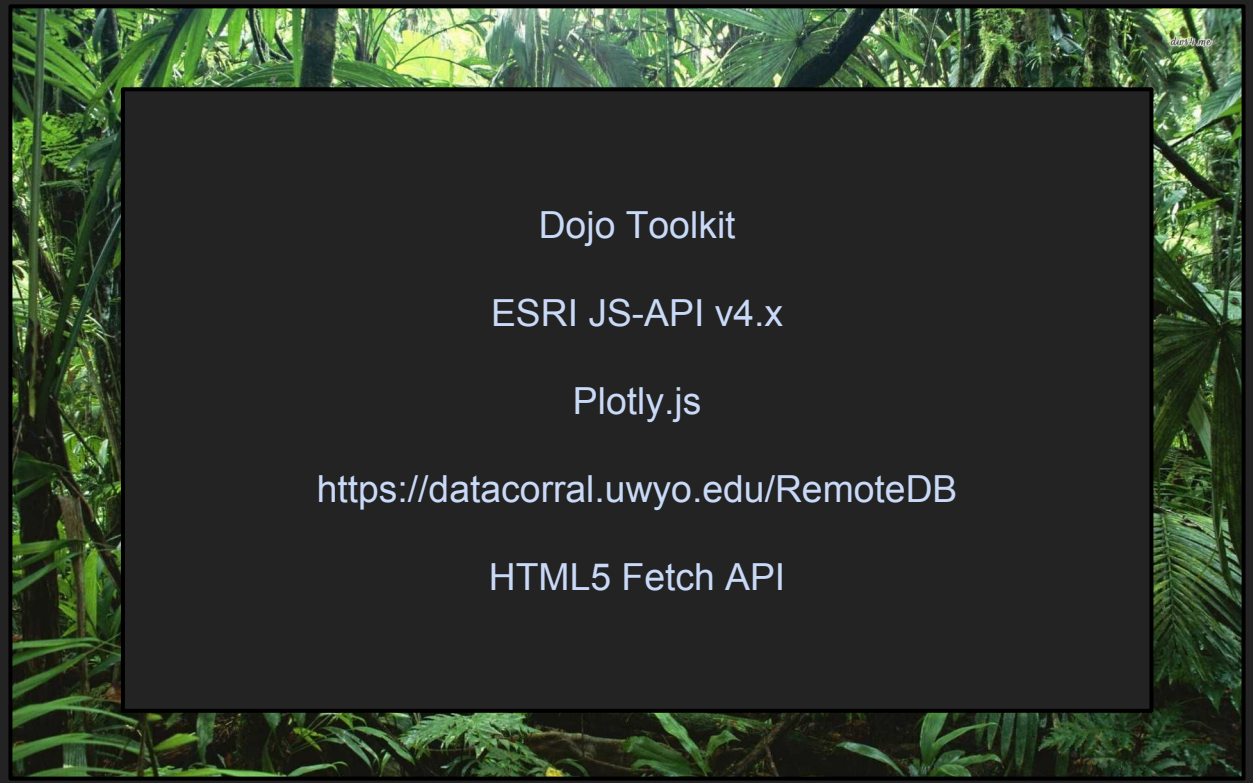
ESRI

MapBox

Google Maps

OpenLayers

Selecting appropriate APIs



Representation of the API ecosystem

APIs + JavaScript + a vision = "Application"

Fetch
request data from a remote source

ESRI JS-API v4.x
interactive map

Dojo Toolkit
control structure, asynchronous module loading

"RemoteDB"
data endpoint API

Plotly.js
graphs / charts / plots

"vanilla" JavaScript
tie it all together, create interactive page elements

CSS + HTML

About your data

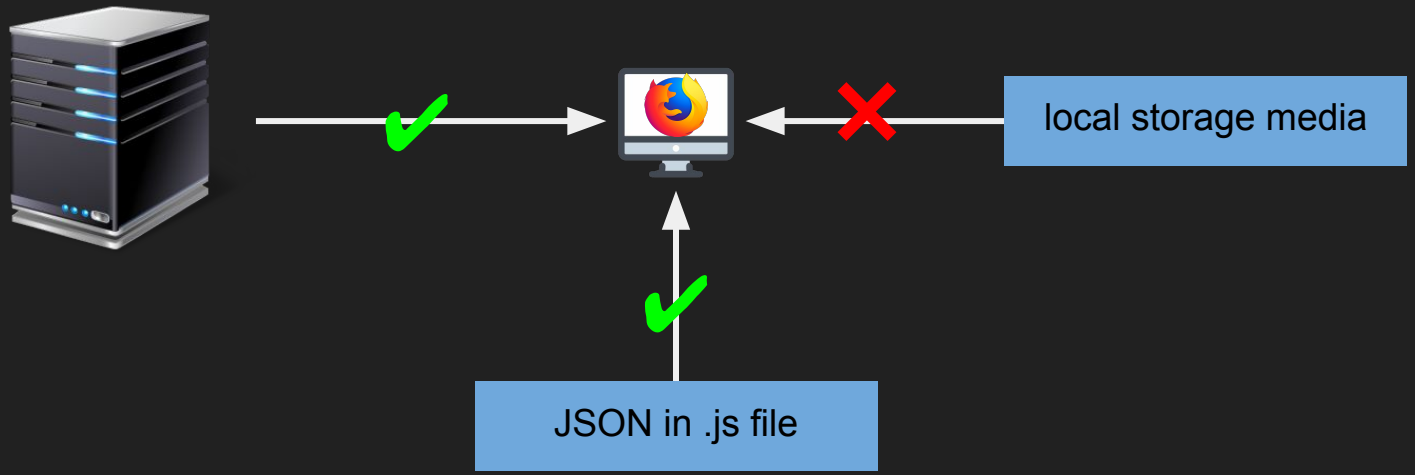
What kind of data are you working with?

"Big" ?

Spatial ?

Ecological ?

Paradigm shift: data and memory in the ephemeral browser!



Publishing Data

- JSON in .js file (local)
- Google Drive API
- github.com
- GeoServer
- host your own
    - REST-ful
    - flat files
    - custom API

Get to the code! Do it!

https://github.com/joshualemli/rmacc2018

Code review

- fetch API and asynchronous control structures
- ploty.js demo with remote data and multiple time series
- ESRI JS-API v4.x basic map demo
- final demo: pulling it all together

Using Web Browsers for Data Visualization

PROS

- publish content
- dynamic (programmatic)
- widely available
- free (mostly)
- make your message heard!

CONS

- slow processing speed
- memory limitations
- requires connectivity or local copy of site
- requires learning additional technologies

Closing remarks

- error checking...
- error checking...
- error checking...
- publishing/accessing your data
- publishing your website with a domain+host
- clear, honest, and meaningful output
- mastering your APIs of choice

Resources

https://developer.mozilla.org/en-US/

https://stackoverflow.com

official API websites

- demos
- documentation
- forums

Attribution

Demo data

Dr. Brent Ewers, UW
DataCorral.uwyo.edu
WyCEHG

Session Materials

plot.ly
esri.com

EPSCoR-WY-t1-2017 "Data Team"

Dr. Shannon Albeke, UW
Samantha Ewers, UW
Dylan Perkins, UW

Moral support

Kema and Coley
Sarah Lemli

WyGISC

Paddington Hodza