

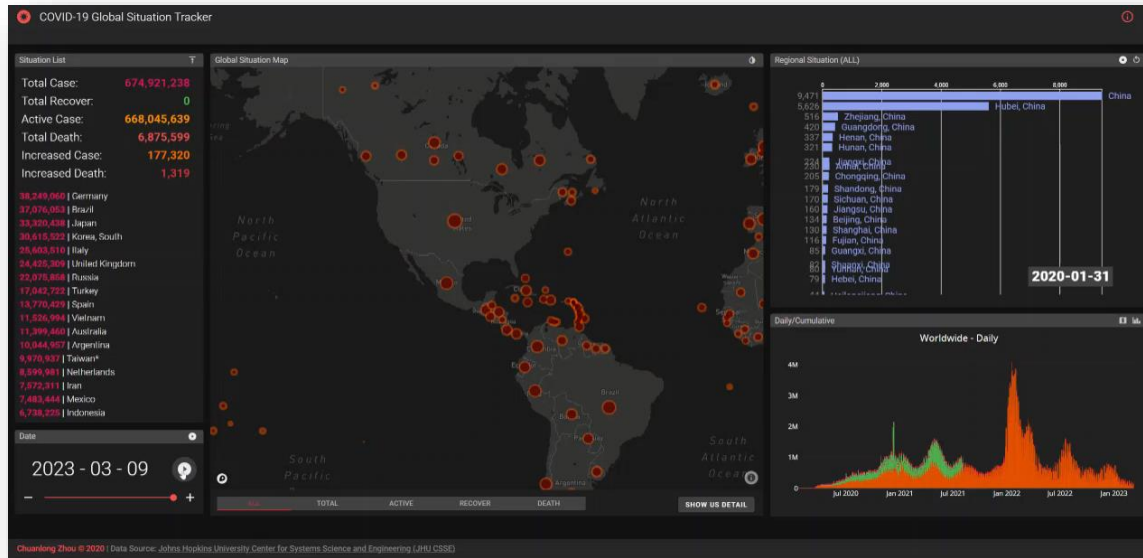
# Visualize Your Project with Web Application

Why you should know some web development technologies

CZ, 2024/03/06

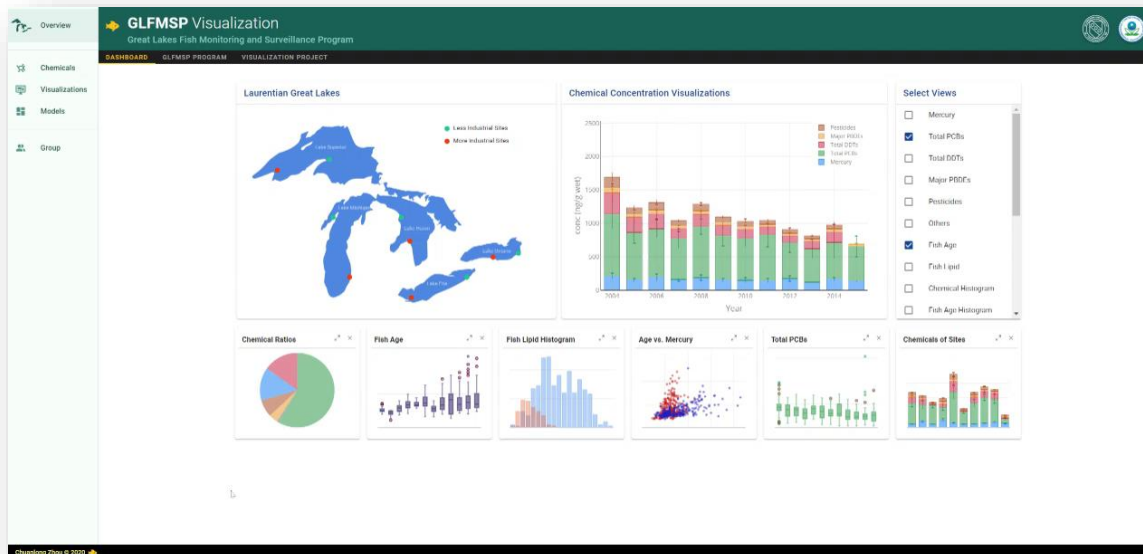
<https://defve1988.github.io/>

# Share your project with cool visualizations

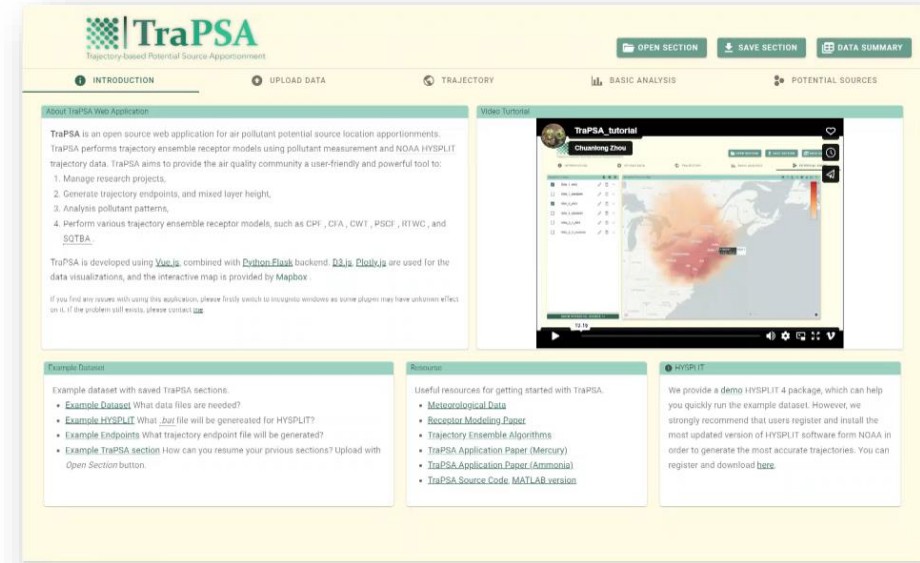


↑ COVID data visualization (<https://covid19-tracker-defve.netlify.app>)

↓ Great lake fish monitoring (<https://glfm-sp-viz.netlify.app>)

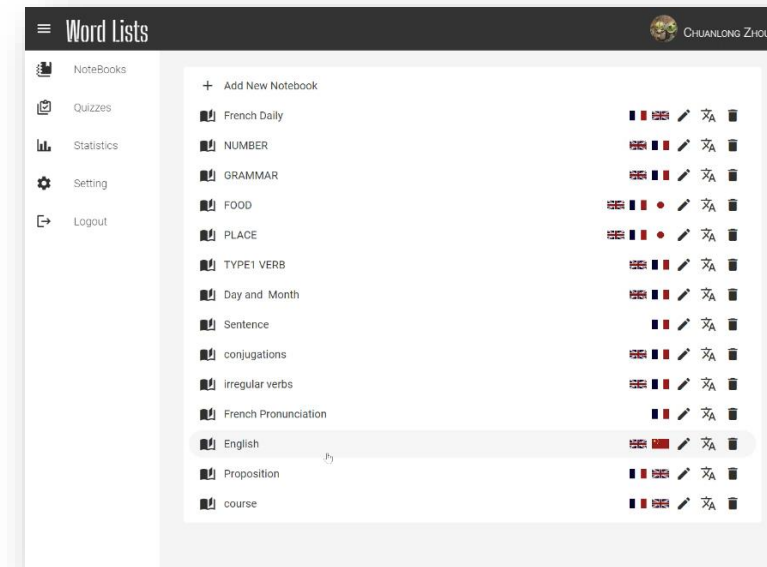


# Develop online applications



↑ Air pollutant source analysis (<https://trapsa.netlify.app/>)

↓ Language Learning (<https://wordlistbydefve.netlify.app>)



Create personal page (<https://defve1988.github.io/>)

**CHUANLONG ZHOU**  
Ph.D. of Environmental Science,  
Enthusiast of Data Science

My name is Chuanlong Zhou | I'm an environmental researcher and data scientist.

### Education

Degree	Date
Georgia Institute of Technology	M.S. of Computer Science 2019-2021
Clarkson University	Ph.D. of Civil and Environmental Engineering 2014-2018
Tongji University	M.S. of Agricultural Engineering 2011-2014
Fuzhou University	B.S. of Water Supply and Sewage Engineering 2007-2011

### Research Experiences

Institute	Date
Le Laboratoire des Sciences du Climat et de l'Environnement	2022-present
Aarhus University	2021
Jinan University	2019-2020
Clarkson University	2014-2018
Clarkson University	2017-2018
Clarkson University	2015-2017
Tongji University	2011-2014

### Projects

- French Forest Inventory (IGN) analysis, ongoing project**  
This project aims to provide insights of the ecological changes of French forests under the past 20 years' climate change and forest management practices. The candidate will be expected to conduct spatial-temporal patterns and trends of tree species distribution, biomass, and mortality and explore the causes of those changes use statistical modeling or machine learning models.
- Building Integral Gridded Carbon Emission Disaggregation Model Framework (BIGCarbon), ongoing project with World Bank**  
This project aims to develop a high-resolution and near real-time CO2 emission model framework based on building level features that predicted with machine learning approach and open datasets for the residential areas in developing countries including Turkey, Egypt, and South America.
- EU27&UK natural gas consumption analysis and Russia-Ukraine crisis**  
In this project, we combine daily data of pipeline flow, production, storage, and consumption of natural gas across EU27 countries and the UK to assess the magnitude of the Russian gas use. We then investigate the magnitude of shortfalls if Russian imports were to stop, and how those gaps could be filled due to the Russia-Ukraine crisis in 2022. We have published a [dataset](#) for daily gas sectoral consumptions in EU27&UK with their supply sources.

Even desktop software: the most famous one ↓

**EXTENSIONS: MARKETPLACE**

- Python** 2019.6.24221 54.9M ★ 4.5  
Linting, Debugging (multi-threaded, ...  
Microsoft **Install**
- GitLens — Git sup...** 9.8.5 23.1M ★ 5  
Supercharge the Git capabilities built...  
Eric Amodio **Install**
- C/C++** 0.24.0 23M ★ 3.5  
C/C++ IntelliSense, debugging, and ...  
Microsoft **Install**
- ESLint** 1.9.0 21.9M ★ 4.5  
Integrates ESLint JavaScript into VS ...  
Dirk Baeumer **Install**
- Debugger for Ch...** 4.11.6 20.6M ★ 4  
Debug your JavaScript code in the C...  
Microsoft **Install**
- Language Supp...** 0.47.0 18.7M ★ 4.5  
Java Linting, Intellisense, formatting, ...  
Red Hat **Install**
- vscode-icons** 8.8.0 17.2M ★ 5  
Icons for Visual Studio Code  
VSCode Icons Team **Install**
- Vetur** 0.21.1 17M ★ 4.5  
Vue tooling for VS Code  
Pine Wu **Install**
- C#** 1.21.0 15.6M ★ 4  
C# for Visual Studio Code (powered ...  
Microsoft **Install**

```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
39
40   checkValidServiceWorker(swUrl, config);
41   // Add some additional logging to localhost, p
42   // service worker/PWA documentation.
43   navigator.serviceWorker.ready.then(() => {
44     product
45     productSub
46     removeSiteSpecificTrackingException
47     removeWebWideTrackingException
48     requestMediaKeySystemAccess
49     sendBeacon
50     serviceWorker (property) Navigator.serviceWorke...
51     storage
52     storeSiteSpecificTrackingException
53     storeWebWideTrackingException
54   } userAgent
55   } vendor
56
57   function registerValidSW(swUrl, config) {
58     navigator.serviceWorker
59       .register(swUrl)
60       .then(registration => {
```

**TERMINAL** 1: node

You can now view **create-react-app** in the browser.

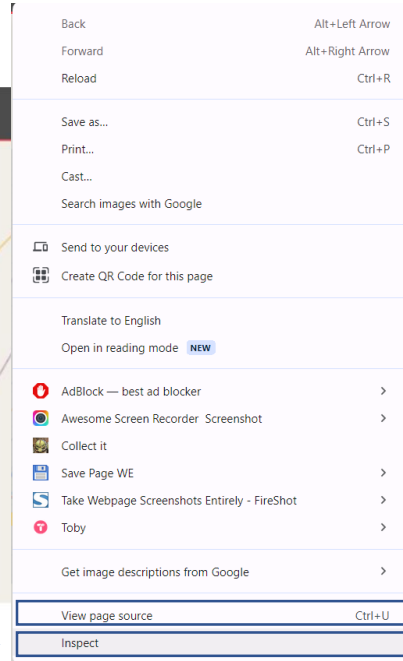
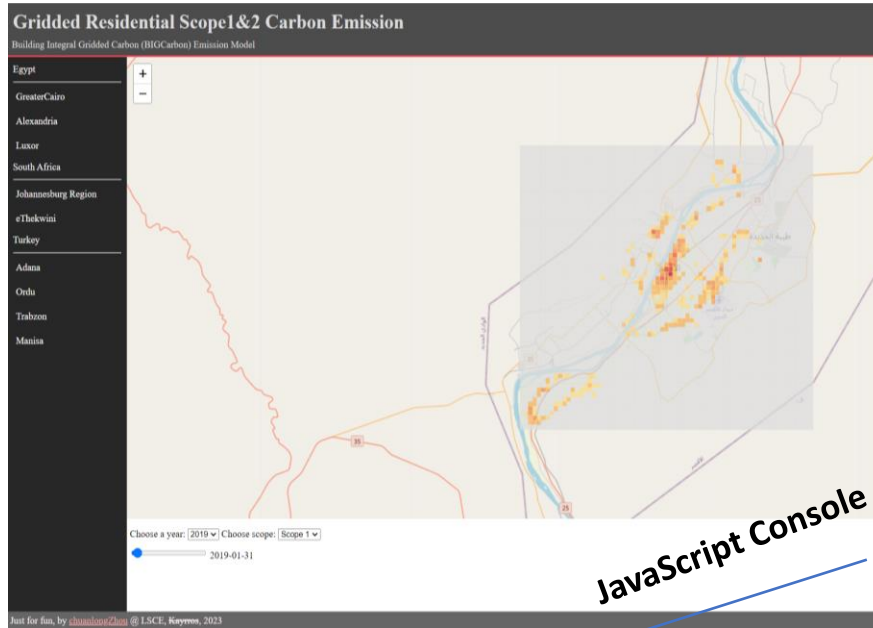
Local: http://localhost:3000/  
On Your Network: http://10.211.55.3:3000/

Note that the development build is not optimized.

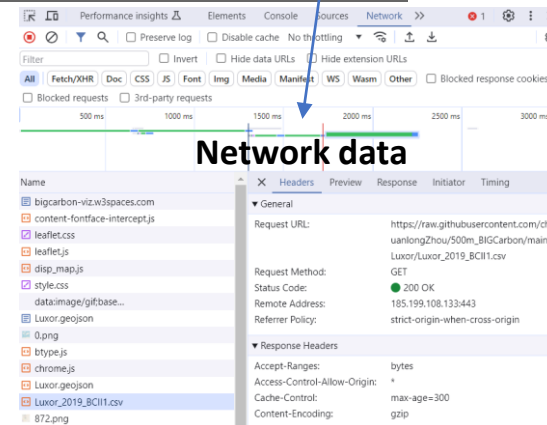
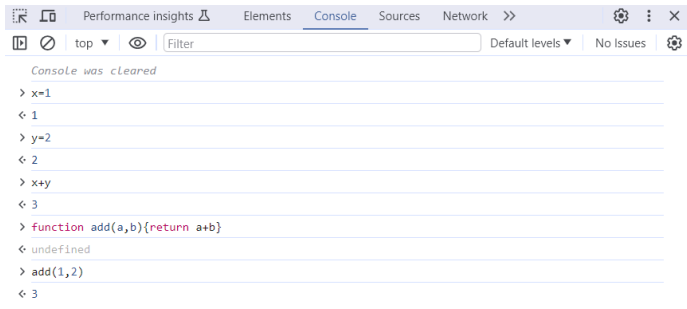
# Goals today

- Understand what's behind a webpage
  - Languages: HTML, JavaScript, CSS, ...
  - Frontend and backend: framework, API, ...
- Know how to choose the right tool for your level
  - Static webpage with vanilla JavaScript
  - Framework/environment: Node.js, Vue, React, Angular, ...
  - Higher level language based: Python, R
  - Visualization: D3, Plotly, Vega, folium, Mapbox, ...
- Everybody can create a dashboard and share it online
  - Streamlit
  - R shiny
- How to host your website
  - Google
  - GitHub
  - W3schools
  - Netlify
  - ...

# What's behind a webpage



JavaScript Console



Network data

Source code

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.3/dist/leaflet.css"
    integrity="sha256-kLaT260sphechsozBflNDzJvE2l5fW4PQ0xYI" crossorigin="" />
  <script src="https://unpkg.com/leaflet@1.9.3/dist/leaflet.js"
    integrity="sha256-HB0x7yKt10Hwt+12uH7pD29fJh0v07ewd" crossorigin=""></script>
  <script src="dis_map.js"></script>
  <link rel="stylesheet" href="style.css" />
  <title>BIGCarbon</title>
</head>

<!-- 'https://raw.githubusercontent.com/chuanlongzhou/500m_BIGCarbon/main/Luxor/Luxor_geojson?token=GHSAT0AAAAAAB6QUX0MBAAS052PYC5QW442ABZK3Q' -->

<body>
  <header class="header">
    <h1>Gridded Residential Scope1&2 Carbon Emission</h1>
    <p>Building Integral Gridded Carbon (BIGCarbon) Emission Model</p>
  </header>

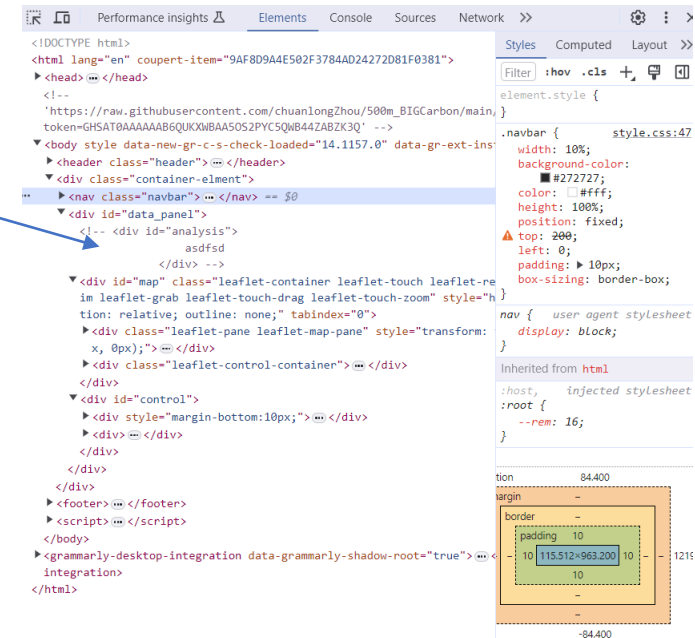
  <div class="container-element">
    <nav class="navbar">
      <ul>
        <li>Egypt</li>
        <li><a href="#" onclick="region_change('GreaterCairo')>GreaterCairo</a></li>
        <li><a href="#" onclick="region_change('Alexandria')>Alexandria</a></li>
        <li><a href="#" onclick="region_change('Luxor')>Luxor</a></li>
        <li>South Africa</li>
        <li><a href="#" onclick="region_change('Johannesburg')>Johannesburg Region</a></li>
        <li><a href="#" onclick="region_change('eThekwin')>eThekwin</a></li>
        <li>Turkey</li>
        <li><a href="#" onclick="region_change('Adana')>Adana</a></li>
        <li><a href="#" onclick="region_change('Ordu')>Ordu</a></li>
        <li><a href="#" onclick="region_change('Trabzon')>Trabzon</a></li>
        <li><a href="#" onclick="region_change('Manisa')>Manisa</a></li>
      </ul>
    </nav>
    <div id="data_panel">
      <!-- <div id="analysis">
        asdfs
      </div> -->

      <div id="map"></div>
      <div id="control">
        <div style="margin-bottom:10px;">

          <label for="years">Choose a year:</label>
          <select name="years" id="years" oninput="input_changes()">
            <option value="2019">2019</option>
            <option value="2020">2020</option>
            <option value="2021">2021</option>
            <option value="2022">2022</option>
          </select>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

HTML  
HyperText Markup Language

Elements





# Frontend: element + style + interaction

**Elements Structure**

**CSS**

**Cascading Style Sheets**

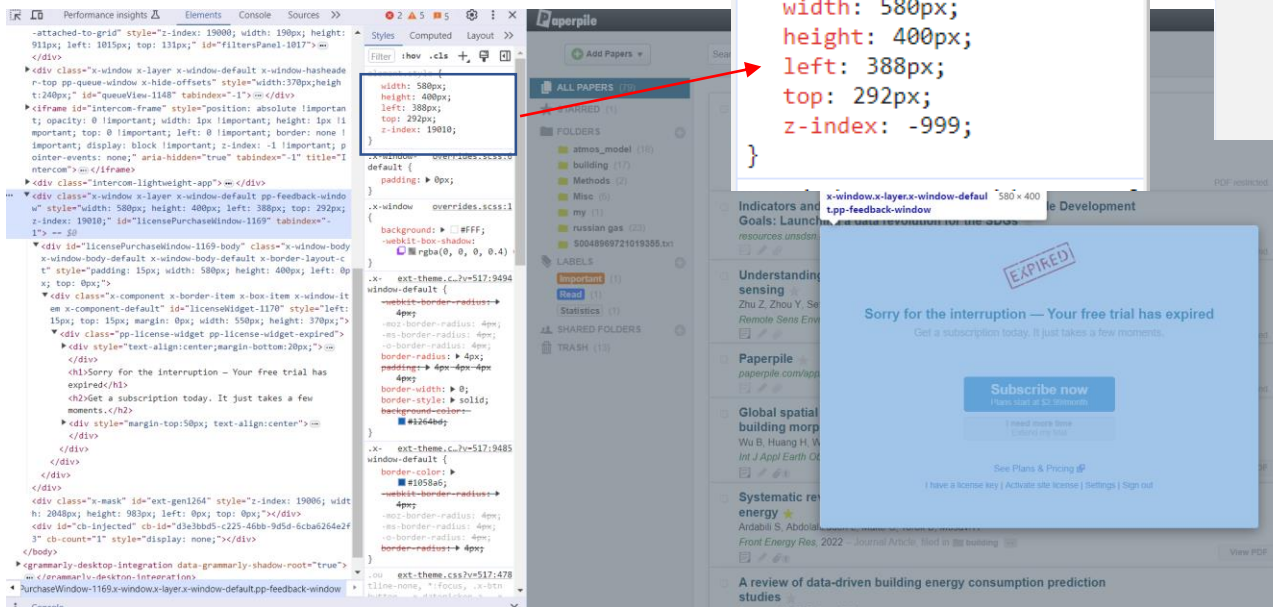
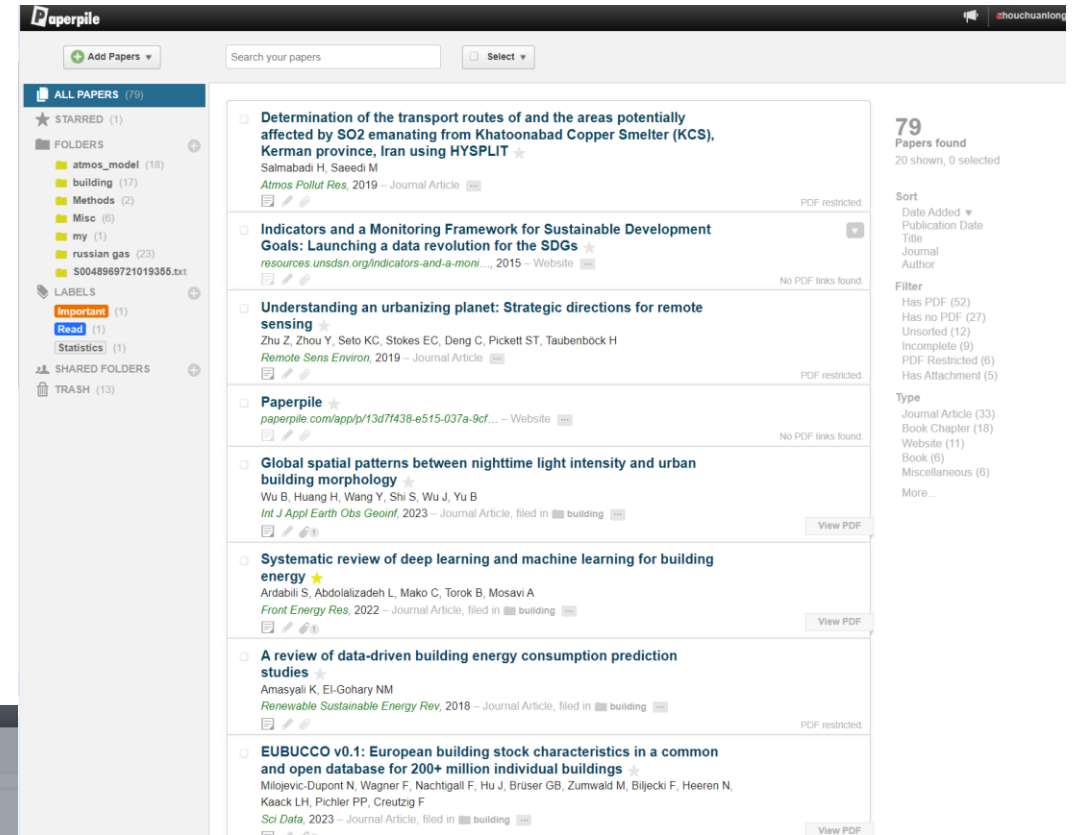
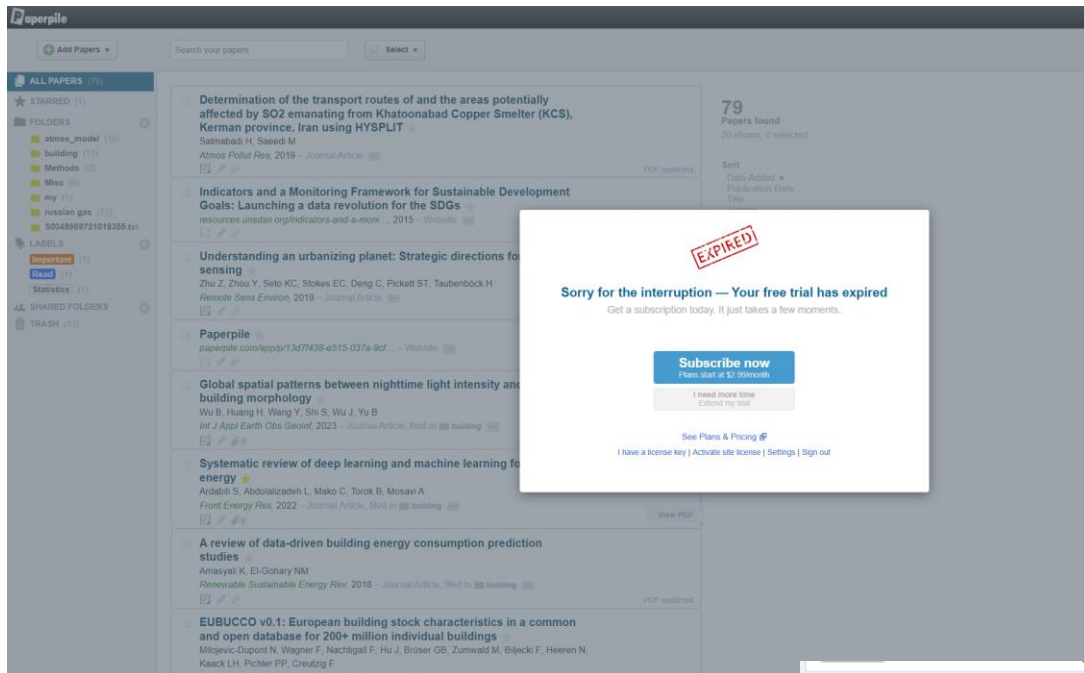
**Carbon Emission**

**Gridded Resid**

**Leaflet | © OpenStreetMap**

You can locate everything from the webpage

You can change the style for the elements



The overlaying window is now on the bottom.

## Interaction: locate the element + manipulations

**Gridded Residential Scope1&2 Carbon Emission**  
Building Integral Gridded Carbon (BIGCarbon) Emission Model

**Regions:** Egypt, GreaterCairo, Alexandria, Luxor, South Africa, Johannesburg, Region, eThekweni, Turkey, Adana, Ordu, Trabzon, Manisa

**Legend:** Regional Emission Intensity Scope 1  
Hover over a grid

**Map Controls:**  
Choose a year: 2019  
Choose scope: Scope 1  
2019-01-31

**Annotations:**  
Interaction function  
Locate the elements  
Manipulate the map

**Code Snippets:**  

```

<select name="years" id="years" oninput="input_changes()" -- $0
<select name="scope" id="scope" oninput="input_changes()" -- $1
<script>
var w = window.innerWidth;
var h = window.innerHeight;
document.getElementById("map").style.height = h*0.8 + "px";

var map = init_map();
var region = "Luxor"
let para = {
  year:2019,
  scope:1,
  date:3,
}

async function input_changes(){
  const year = document.getElementById("years").value;
  const scope = document.getElementById("scope").value;
  const date = document.getElementById("date").value;
  let res = await get_data(region, year, date, scope)
  document.getElementById("date_label").innerHTML = res.date;
  map.off();
  map.remove();
  map = init_map()
  display_region(res.geoJson_data, region, scope, res.grade_list)

  return {year, scope, date}
}

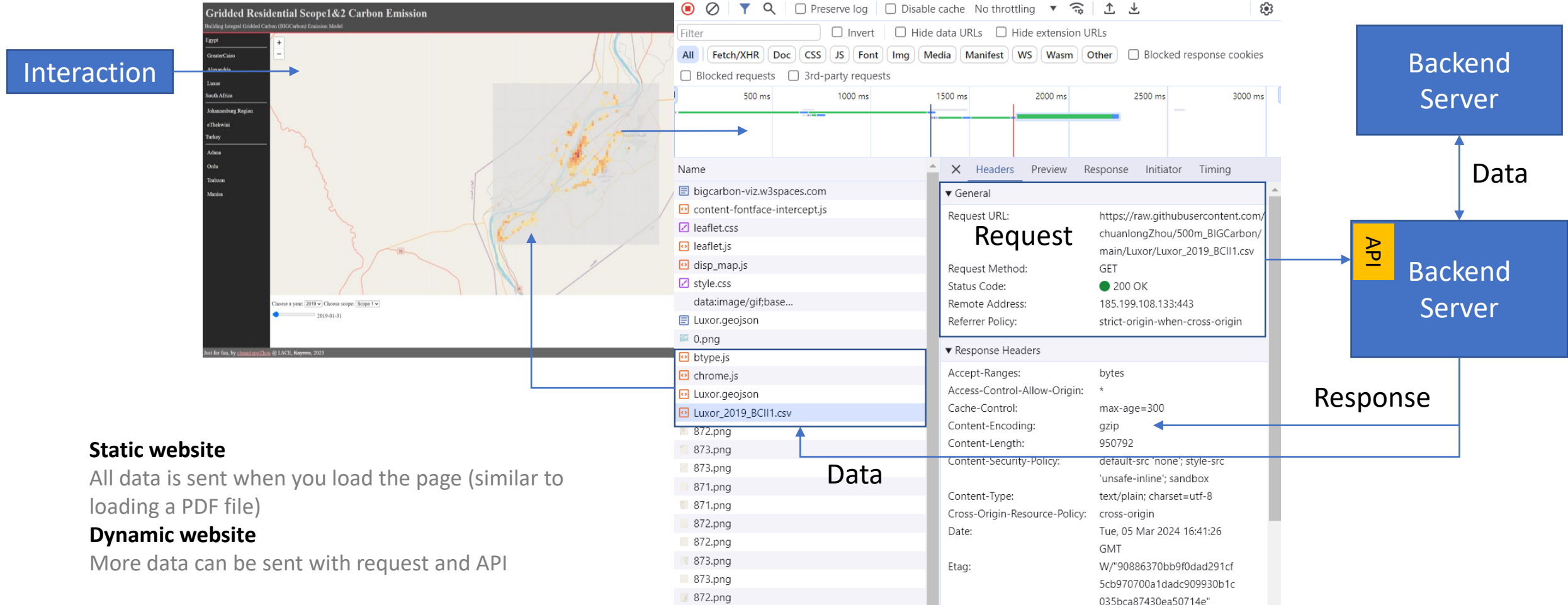
```

**Map Data:** 53.6 × 18.4  
2019-01-31

**Footer:** Just for fun, by [chuanlongZhou](#) @ LSCE, Kayros, 2023

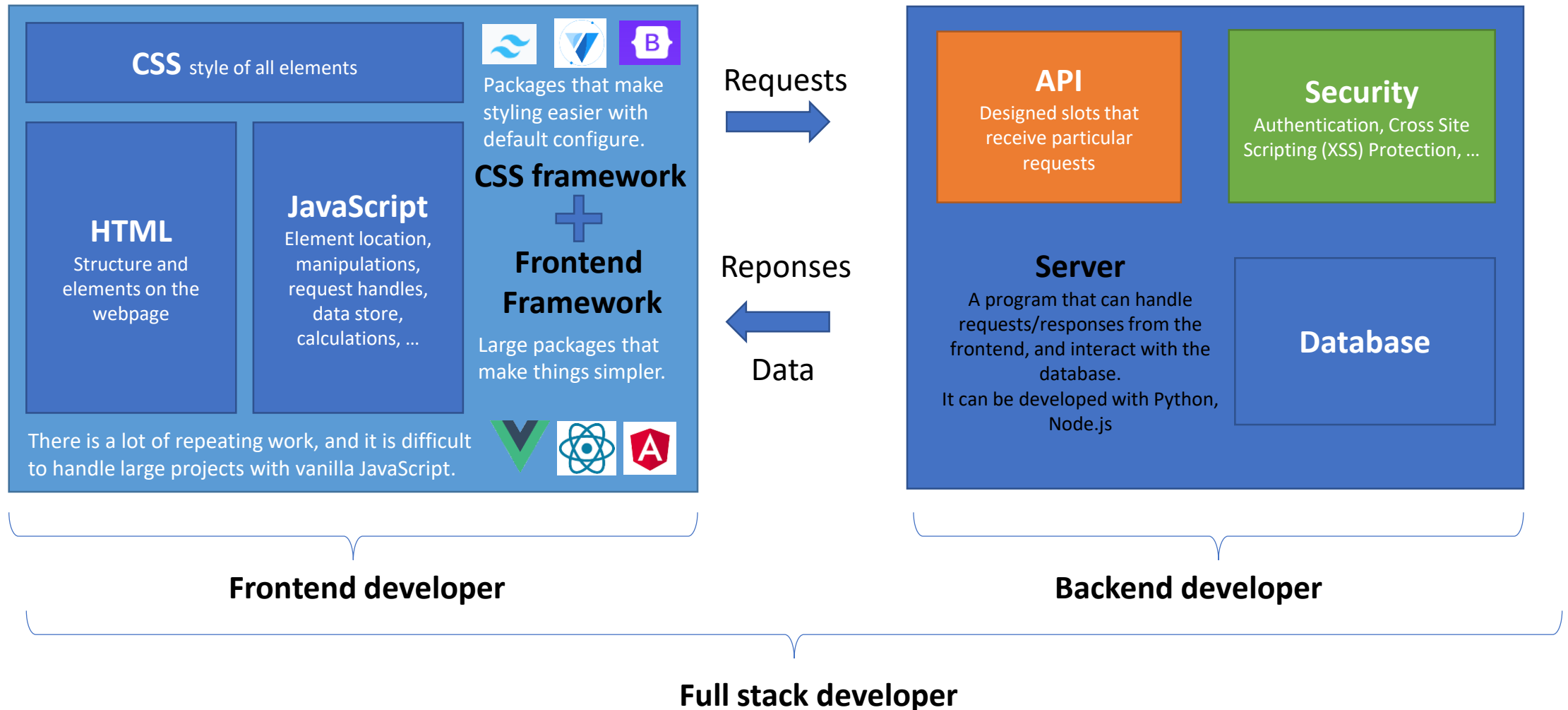


# Backend: request + api



# Normal way

In reality, it would be much more complicated.  
For example, how to handle thousands of concurrent requests?



# Shortcuts!!

- There are lots of “lower-code” web development tools.

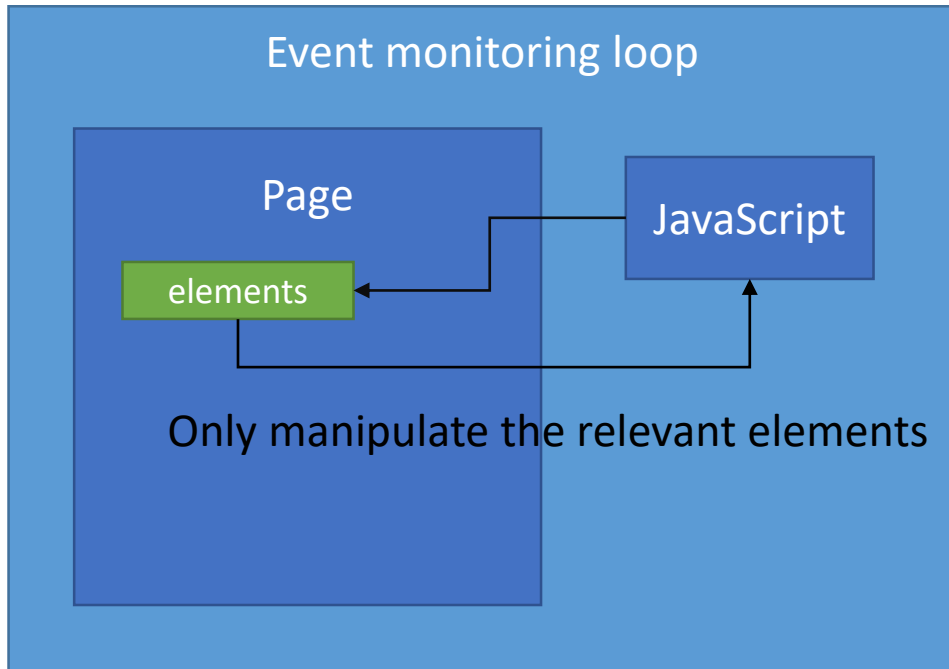
## **Advantages:**

- **Default elements:** You don't need to develop elements from scratch.
- **Default interaction:** You don't need to handle the interactions, or UI animations at the low levels.
- **One language:** You can code your project only with one language, e.g., Python or R.
- **Platform backend:** You don't need to handle the backend server.

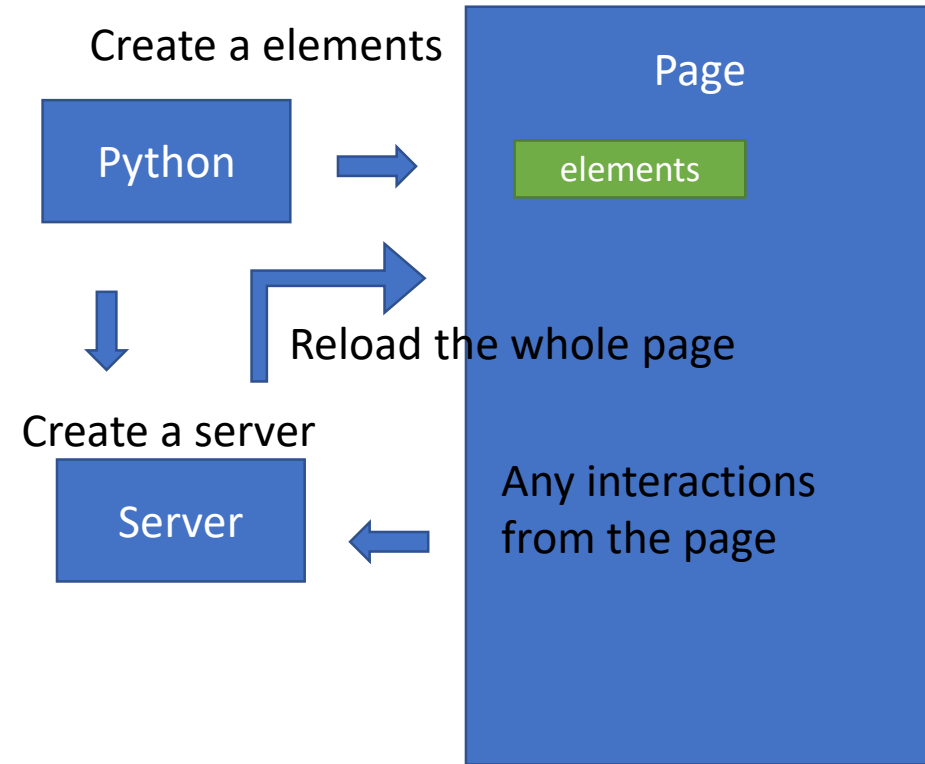
**Drawbacks:** loss of the flexibility to create something fancy

# Streamlit

Normal Website



Streamlit



Not so efficient for page rendering, but simple



# Right tool for your level

- Streamlit tutorials: [https://github.com/chuanlongZhou/streamlit\\_tutorial.git](https://github.com/chuanlongZhou/streamlit_tutorial.git)
- Host your Streamlit app: <https://share.streamlit.io/>
- Carbon monitor 2.0 project: <https://chuanlongzhou-wb-city-web-carbon-monitor-city-2-0-omsxt3.streamlit.app/>
- Biomass recovery: <https://chuanlongzhou-global-grid-map-app-bz8r8l.streamlit.app/>
- French IGN: <https://chuanlongzhou-lulucf-viz-lulucf-egobru.streamlit.app/>

# How to host your website

- Single page
  - GitHub
  - Google
  - W3schools
- Large project
  - Netlify
  - FireBase
  - Heroku (no free version now)

# Interactive visualization packages/tools