Introduction to JavaScript

GW Libraries & Academic Innovation
October 9, 2023

Max Turer, Front-End Web Developer
Dan Kerchner, Senior Software Developer

go.gwu.edu/jsws

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

About Max

Front-End Web Developer for GW Libraries & Academic Innovation

Background in UI/UX development and interactive design

Contact: max.turer@gwu.edu

About Dan

Senior Software Developer for GW Libraries & Academic Innovation

Graduate student in GWSPH Health & Biomedical Data Science

Contact: kerchner@gwu.edu

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Objectives

Understand basics of the JavaScript language (variables, data types, loops, functions)

Write JavaScript code to add interactivity to a web page built with HTML and CSS

Write JavaScript code to create a web page that accepts user inputs and creates a data visualization

Web basics

HTML - Structure
CSS - Styles
JavaScript - Functionality

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>Hello world!</title>

    <!-- import the webpage's stylesheet -->
    <link rel="stylesheet" href="/style.css" />

    <!-- add styles directly to the HTML -->
    <style>
      body {
        font-family: sans-serif;
      }
    </style>

    <!-- import the webpage's javascript file -->
    <script src="/script.js" defer></script>

    <!-- add javascript directly to the HTML -->
    <script>
      console.log("Hello world!");
    </script>
  </head>
  <body>
    <!-- this is the start of content -->
    <h1>Hello World!</h1>
    <p>
      This starter gives you everything you need to start working on a new
      website, and nothing more.
    </p>
  </body>
</html>
```

What can JavaScript do?

- User interaction: menu dropdowns, tabs, carousels
- Dynamically add content to page (Ajax)
- Data visualization (Plotly, D3.js)
- Applications (Instagram)
- Anything another programming language could do (Node.js)

JavaScript syntax basics

- Whitespace mostly doesn't matter, except for readability
- Semicolons `;` end statements/lines
- Curly braces `{}` group statements
- Comments can be `// Single line` or
  ```
  /*
  Multi-line
  */
  ```
- Use `console.log(SOMETHING);` to show values

Data types

- Strings (text): `'a string has quotes around it'`
- Numbers: `1, -1`
- Boolean: `true, false`
- Arrays: `['a', 'list', 'of', 'other', 'values', true, 123]`
  - Get a specific value: `array[index]` (eg. `array[0] -> 'a'`)
- Objects: `{'key': 'value pairs', 'length': 2}`
  - Get a specific value: `object[key]` (eg. `object['length'] -> 2`)
- Variables: **`const`** `varName = 1;`
  - Use **`let`** to reassign later (like in loops)
- Functions:
  - **`function`** `functionName(name) {` **`return`** `` `Hello '${name}`; }``
  - Also arrow functions: `(name) => `` `Hello ${name}`;``

Operators

- Math: `+, -, *, /, %`
  - `1 + 2 -> 3`
  - `'Hello' + 'world' -> 'Helloworld'`
  - `'1' + 2 -> '12'`
- Logical: `&&, ||`
  - `true && false -> false`
  - `true || false -> true`
- Equality: `===, !==, ==, !=`
  - `'1' === 1 -> false`
  - `'1' == 1 -> true`
  - `'1' !== 1 -> true`
  - `'1' != 1 -> false`

Conditionals

```
if (some statement that results in true or false)
{
  // Do something if the condition is true
} else if (another statement that is true or
false) {
  // Do something if the new condition is true
} else {
  // Do something if the all conditions are false
}
```

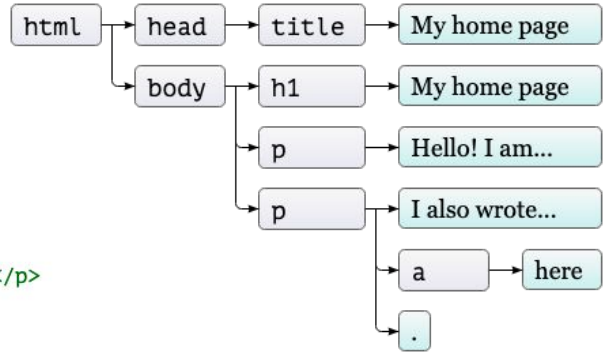
Array methods

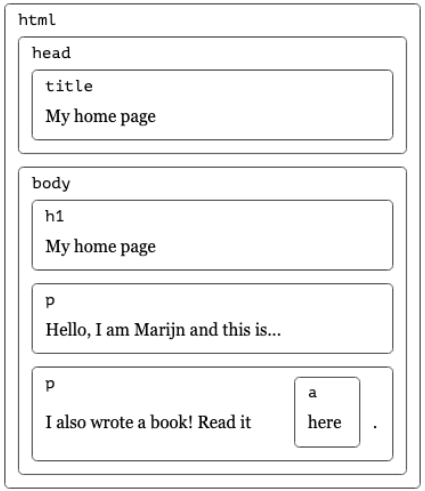
```
const nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
const double = nums.map((value) => value * 2);
    -> [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
const evens = nums.filter((value) => value % 2 === 0);
    -> [0, 2, 4, 6, 8]
const sum = nums.reduce((prev, value) => prev + value, 0);
    -> 45
```

DOM/Events

```
const element = document.querySelector(CSS SELECTOR);
OR document.querySelectorAll(CSS SELECTOR);
element.addEventListener('click', (event) => {
  // Do something when an event happens.
});
```

```
<!doctype html>
<html>
  <head>
    <title>My home page</title>
  </head>
  <body>
    <h1>My home page</h1>
    <p>Hello, I am Marijn and this is my home page.</p>
    <p>I also wrote a book! Read it
      <a href="http://eloquentjavascript.net">here</a>.</p>
  </body>
</html>
```





Build something

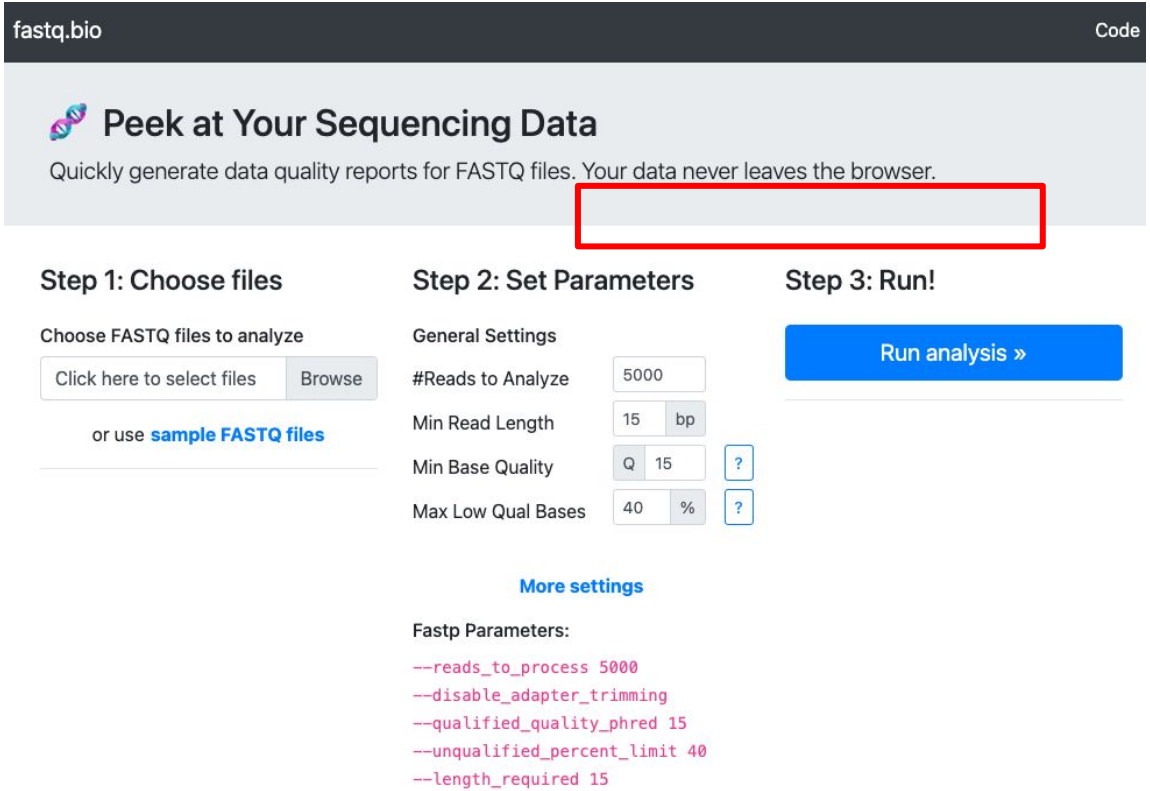
Go to glitch.com/~gwu-intro-js

Select “Remix your own” at bottom right

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

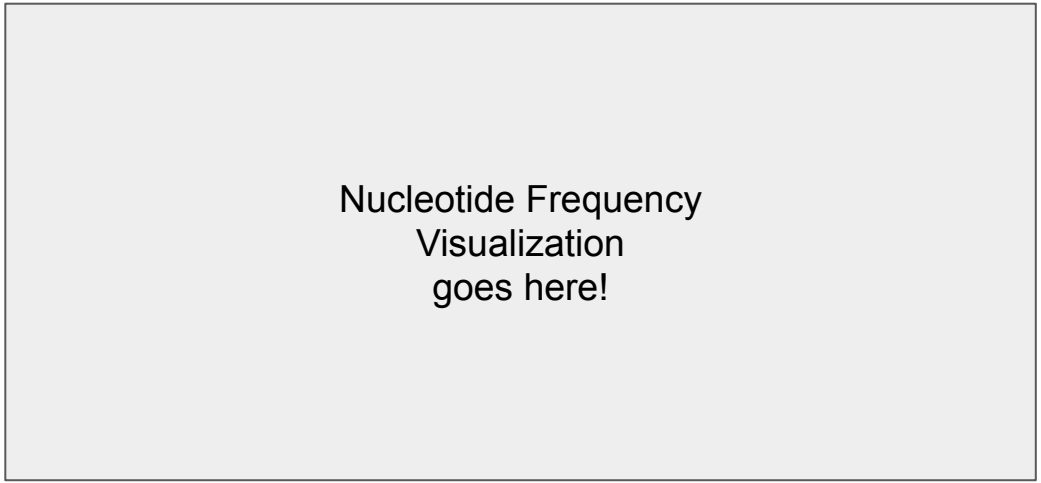
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

A Basic Data Computation and Visualization Example



THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Enter sequence here: AGCTCCTTAGGCATCAGGATCGNNNGTT Compute Reset



Things we need to do

- Lay out the page and its elements
- Make each button call a new JavaScript function
- Enhance our JavaScript "Compute" function to:
  - Create a static ("canned") visualization, then:
  - Calculate nucleotide frequencies, then:
  - Pass the nucleotide frequencies to the visualization (so that it's no longer static)

Where can we publish our page/site?

- Github Pages (free)
  - Example: github.com/kerchner/NucDist (view at kerchner.github.io/NucDist)
- Your web server
- Other ideas?

Multiple contexts for running JavaScript

- At the command line

```
kerchner % node test.js
hello, world
```

- In a "notebook" (such as ObservableHQ)
- Within a web page

Also check out…

- Tools/platforms
  - observablehq.com - JavaScript computational notebooks
- JavaScript libraries for data visualization
  - **d3**js.org - Data visualization - also see d3-graph-gallery.com
  - **plotly**.com/javascript
  - vega.github.io/**vega-lite** - grammar of graphics for data visualization
- Python in JavaScript
  - Pyodide: github.com/iodide-project/pyodide/

Resources

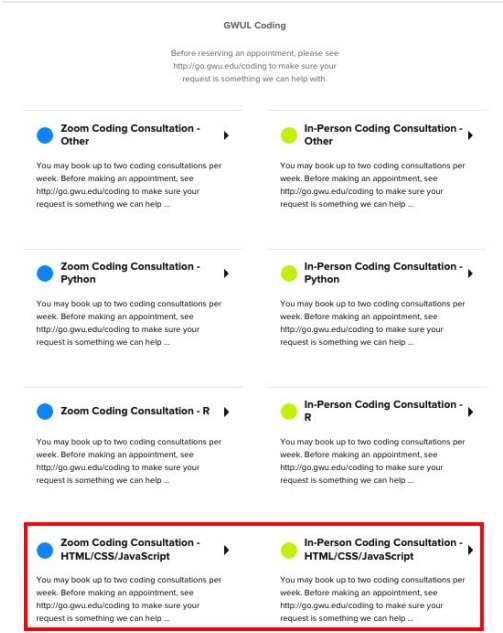
- Web Technology for Developers (Mozilla): developer.mozilla.org/en-US/docs/Web
- Stack Overflow: stackoverflow.com
- Try online: glitch.com, jsfiddle.net, codepen.io
- Online books:
  - Eloquent JavaScript - introduction with exercises & examples: eloquentjavascript.net
  - JavaScript for Data Science: third-bit.com/js4ds
- Tutorials/videos:
  - linkedin.com/learning/learning-the-javascript-language-2/**learn-the-language-of-the-internet** (and more on LinkedIn learning)

Coding Consultations @ GW Libraries

calendly.com/gwul-coding

- **HTML/CSS/JavaScript**
- Python
- R
- General coding questions

These slides can be downloaded from go.gwu.edu/jsws



THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC