

Client-Side Web Programming: JavaScript (Part 3)

Copyright © 2024 by
Robert M. Dondero, Ph.D.
Princeton University

Objectives

- We will cover:
 - JavaScript libraries for client-side web programming
 - jQuery

Agenda

- **AJAX via XMLHttpRequest enhancements**
- JavaScript libraries
- jQuery

AJAX Enhancements

- Recall **PennyAjax1** app
 - runserver.py
 - penny.sql, penny.sqlite
 - database.py
 - **penny.py**
 - **index.html**

AJAX Enhancements

- PennyAjax1 app is a *single page app (SPA)*
- SPAs are enabled by AJAX

AJAX Enhancements

- **Problem:**
 - Code to convert JavaScript data structure to HTML doc is ugly, inefficient
- **Solution:**
 - Use a JavaScript **template engine**, such as *Mustache*

AJAX Enhancements

- See **PennyAjax2** app
 - runserver.py
 - penny.sql, penny.sqlite
 - database.py
 - penny.py
 - **index.html**

AJAX Enhancements

- **Problem:**
 - Server will respond to requests in arbitrary order
- **Solution:**
 - Abort previous request

AJAX Enhancements

- See **PennyAjax3** app
 - runserver.py
 - penny.sql, penny.sqlite
 - database.py
 - penny.py
 - **index.html**

AJAX Enhancements

- **Problem:**
 - Server could be overwhelmed with requests
- **Solution:**
 - *Debounce* the requests

AJAX Enhancements

- See **PennyAjax4** app
 - runserver.py
 - penny.sql, penny.sqlite
 - database.py
 - penny.py
 - **index.html**

AJAX Enhancements

- **Bonus:**
 - Debouncing reduces the need to abort requests

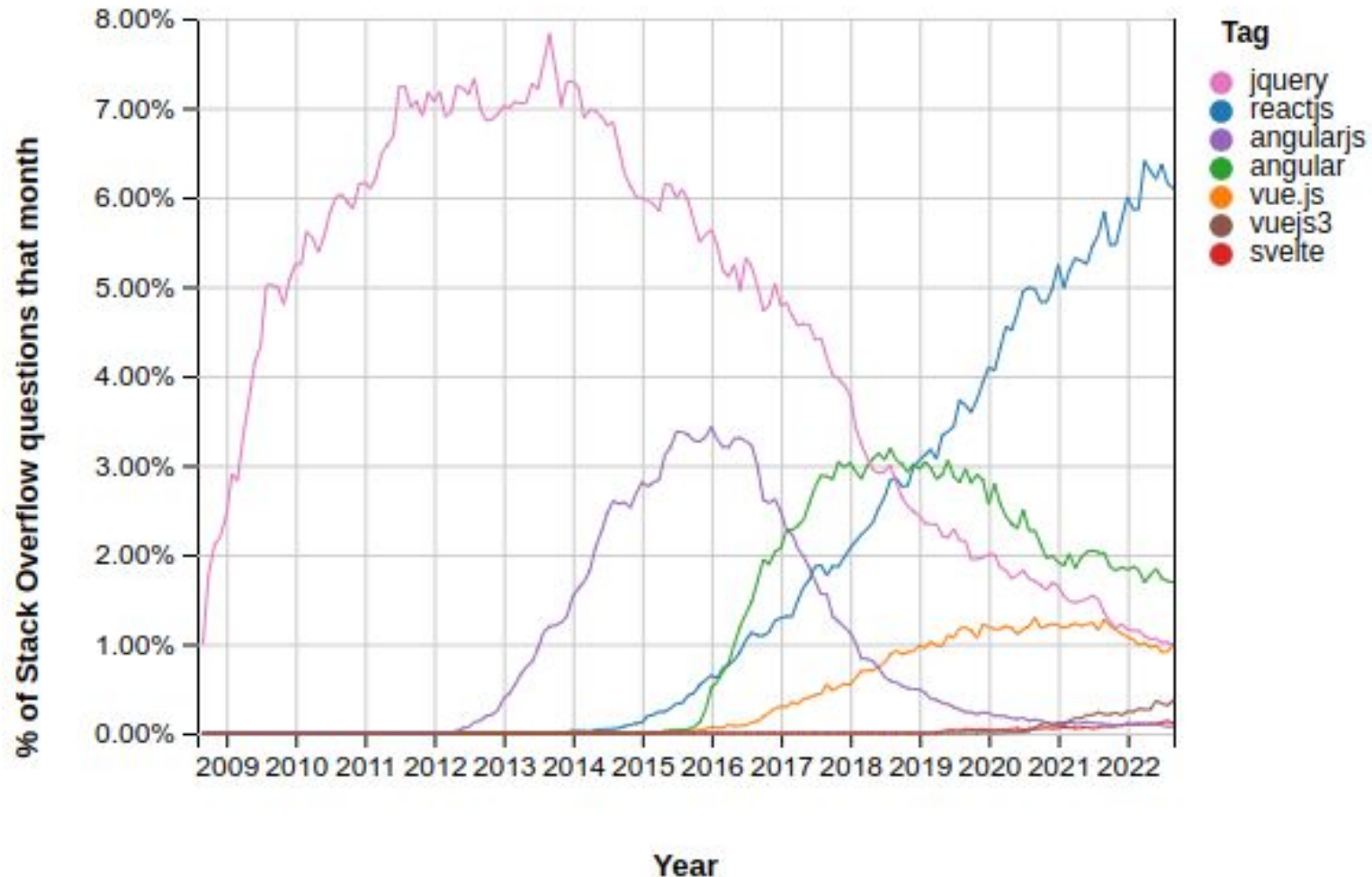
Agenda

- AJAX via XMLHttpRequest enhancements
- **JavaScript libraries**
- jQuery

JavaScript Libraries

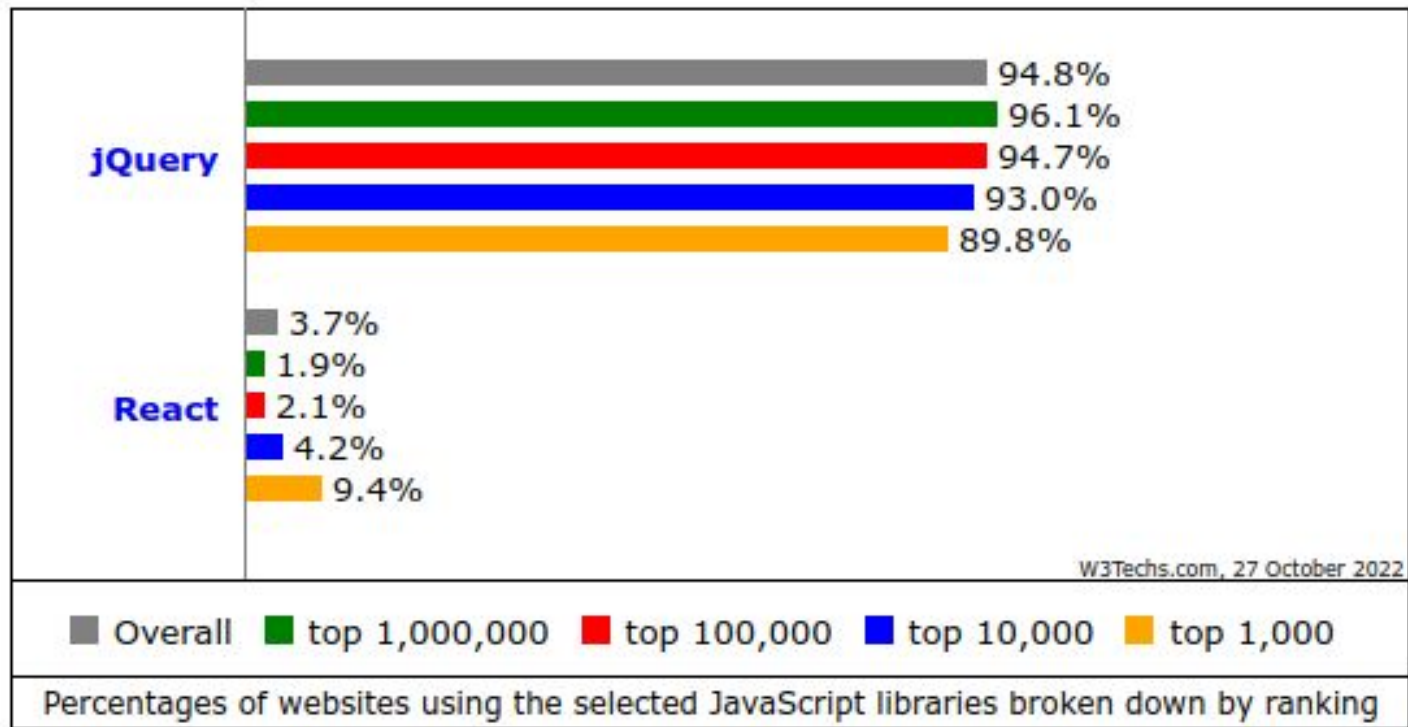
- **Problem:**
 - JavaScript/AJAX code uses common patterns and often is repetitive
- **Solution:**
 - Use a JavaScript **library**
 - **jQuery**, **React**, AngularJS, Angular, vue.js, vue3, svelte, ...

JavaScript Libraries



As of Oct 2022, according to <https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Csvelte%2Cangularjs%2Cvuejs3%2Cjquery>

JavaScript Libraries



As of October 2022, according to <https://w3techs.com/technologies/comparison/js-jquery,js-react>

Agenda

- AJAX via XMLHttpRequest enhancements
- JavaScript libraries
- **jQuery**

jQuery



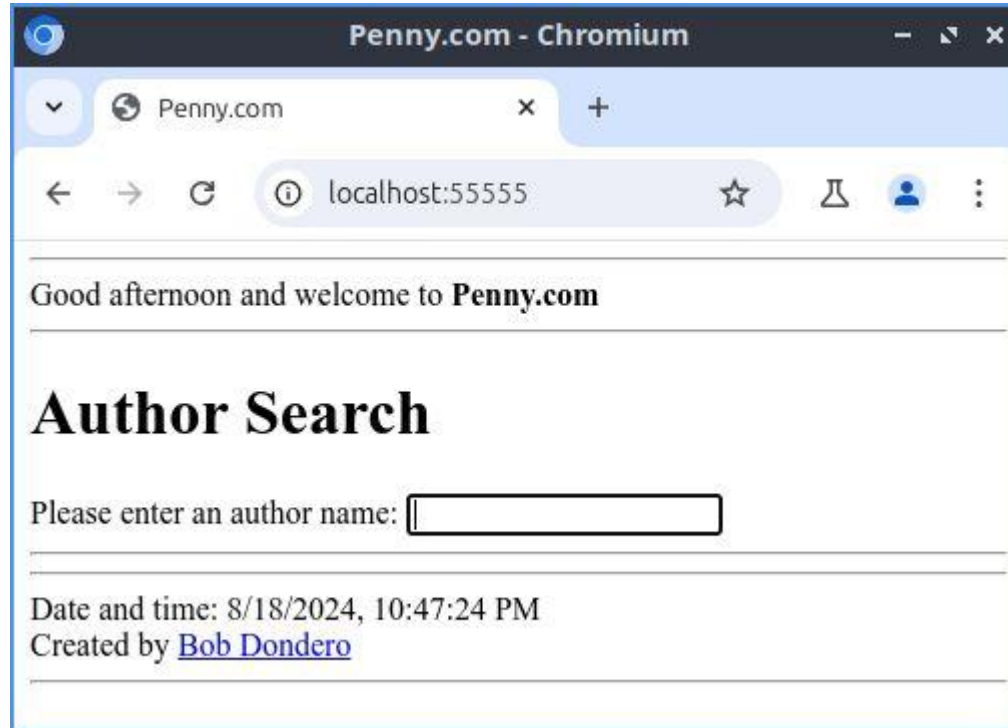
John
Resig

jQuery

- jQuery statement syntax:
 - `jQuery(selector).action()`
 - `$(selector).action()`
 - `$`
 - The jQuery function
 - ***selector***
 - Selects DOM node(s)
 - As in CSS; covered soon
 - ***action()***
 - Specifies an action to be performed on the selected DOM node(s)

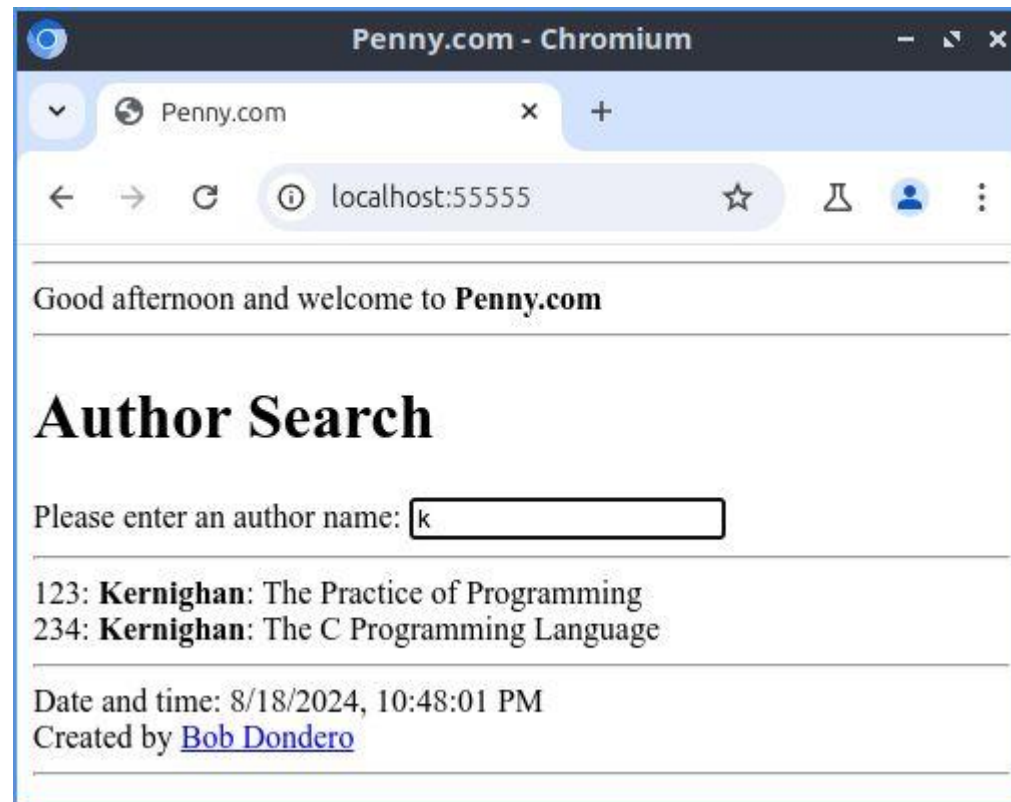
jQuery

- See **PennyjQuery** app



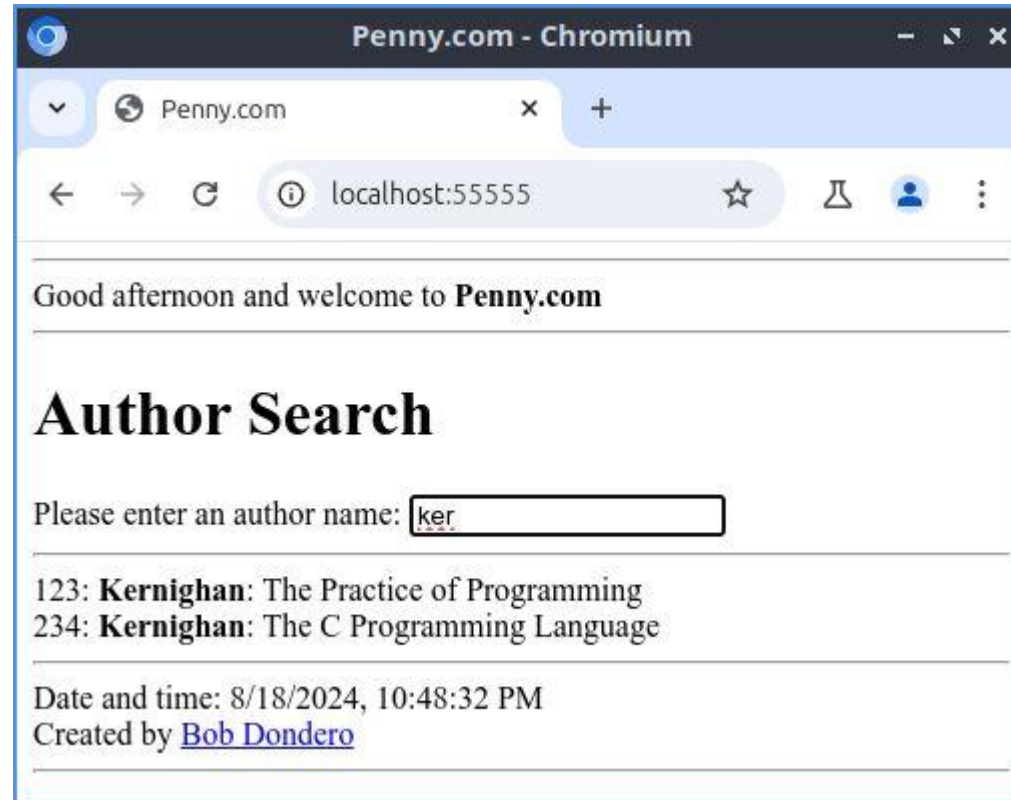
jQuery

- See **PennyjQuery** app



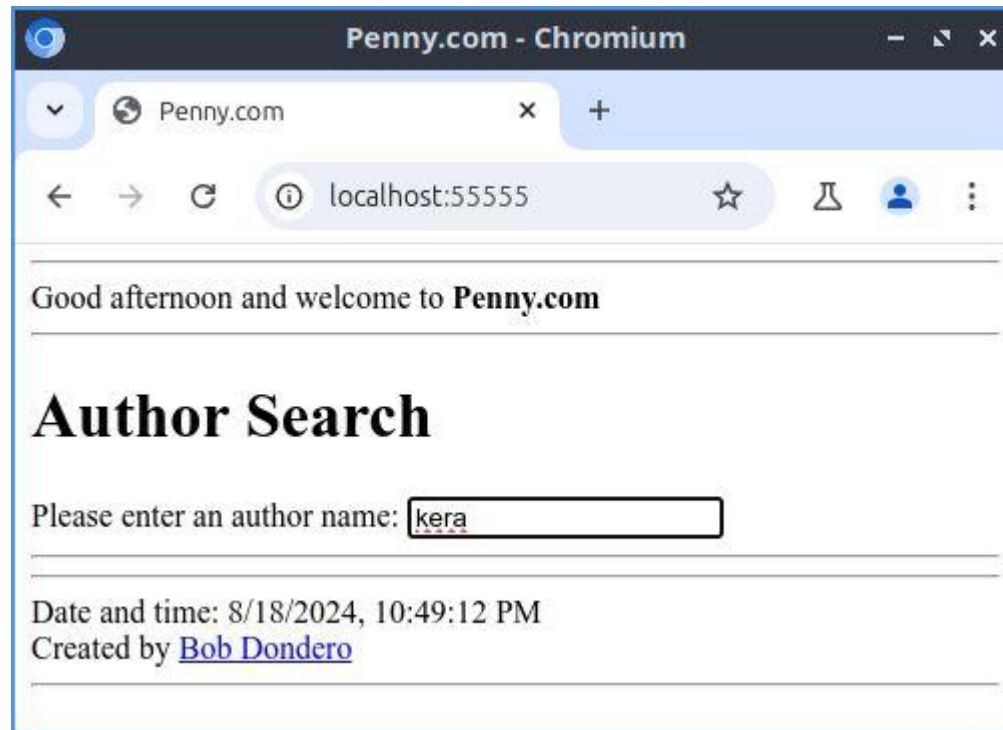
jQuery

- See **PennyjQuery** app



jQuery

- See **PennyjQuery** app



jQuery

- See **PennyjQuery** app (cont.)
 - runserver.py
 - penny.sql, penny.sqlite
 - database.py
 - penny.py
 - **index.html**

jQuery

- jQuery summary...

jQuery

- jQuery makes accessing the DOM easier

Without jQuery:

```
let ampSpan =  
    document.getElementById('ampSpan');  
ampSpan.innerHTML = amPm;
```

With jQuery:

```
$('#ampSpan').html(amPm);
```

=> access by id

jQuery

- jQuery makes AJAX easier

Without jQuery:

```
function handleResponse() {
    if (this.status !== 200) {
        alert('Error: Failed to fetch data from server');
        return;
    }
    let books = JSON.parse(this.response);
    let html = convertToHtml(books);
    let resultsDiv = document.getElementById('resultsDiv');
    resultsDiv.innerHTML = this.responseText;
}
function getResults() {
    ...
    request = new XMLHttpRequest();
    request.onload = handleResponse;
    request.onerror = handleError;
    request.open('GET', url);
    request.send();
}
```


jQuery

- jQuery makes AJAX easier (cont.)

With jQuery:

```
function handleResponse(books) {  
  let html = convertToHtml(books);  
  $('#resultsDiv').html(html);  
}  
  
function getResults() { ...  
  let requestData = {  
    type: 'GET',  
    url: url,  
    success: handleResponse,  
    error: handleError  
  };  
  request = $.ajax(requestData);  
}
```

jQuery
parses JSON
automatically



jQuery

- jQuery pros
 - Easy to learn
 - Especially if you know CSS
 - Lots of web info
 - Easy to use
 - **Handles a wide variety of browsers**

jQuery

- jQuery cons
 - Less necessary with current browsers
 - Incompatible with client-side libraries that use a **virtual DOM** (e.g., React)

Aside: AJAX Implementations

AJAX Implementation	Browser Built-In or Library?	Async Mechanism	COS 333 Coverage
XMLHttpRequest	Built-in	Callbacks	Last lecture
fetch & AbortController	Built-in	Promises	Last lecture appendix
Axios	Library	Promises	None
jQuery	Library	Callbacks (or promises)	This lecture

Aside: AJAX Implementations

AJAX Implementation	Firefox	Chrome
XMLHttpRequest	12+ (2012)	31+ (2013)
fetch	39+ (2015)	42+ (2015)
AbortController	57+ (2017)	66+ (2018)
Axios	12+ (2012)	31+ (2013)
jQuery	12+ (2012)	31+ (2013)

Summary

- We have covered:
 - JavaScript libraries for client-side web programming
 - jQuery