

Client-Side Web Programming: JavaScript (Part 3)

Copyright © 2025 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 (cont.)**
- JavaScript libraries
- jQuery

AJAX Enhancements

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

AJAX Enhancements

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

Aside: Template Engines

- Python
 - Mustache, CheetahTemplate, Django, Genshi, **Jinja2**, Kid, Topsite, ...
- JavaScript
 - **Mustache**, Squirrelly, Handlebars, ...
- Java
 - Mustache, FreeMarker, Hamlets, Tiles, Thymeleaf, WebMacro, WebObjects, Velocity, ...

https://en.wikipedia.org/wiki/Web_template_system

AJAX Enhancements

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

AJAX Enhancements

- **How to fetch the Mustache library...**
- **Option 1**
 - Command browser to fetch Mustache library from the **cdn** website
- **Option 2**
 - Ahead of time copy the Mustache library to **your** website
 - Command browser to fetch Mustache library from your website

AJAX Enhancements

- Mustache
 - Template (informally)
 - A String containing HTML code with placeholders
 - Each placeholder is identified by a Mustache variable

```
Hello <strong>{ {username} }</strong>  
and welcome
```

AJAX Enhancements

- Mustache (cont.)
 - To instantiate a template:

```
let map = {somevar: someval, ...};  
let html = Mustache.render(sometemplate, map);
```

- For each placeholder identified by `somevar` in `sometemplate`, Mustache replaces the placeholder with `someval`
- **Automatically sanitizes/escapes** `someval`
- Returns the resulting string

AJAX Enhancements

- Mustache (cont.)
 - Template can contain:
 - Variables

```
... {{author}} ...
```

AJAX Enhancements

- Mustache (cont.)
 - Template can contain:
 - Iteration constructs

```
{{#books}}  
  <strong>{{author}}</strong>  
  ...  
{{/books}}
```

Note:

- Unusual implicit specification of iteration object
- If books is falsy, then block is not rendered

AJAX Enhancements

- Mustache (cont.)
 - Template can contain:
 - Selection constructs

```
{ {#books} }  
...  
{ {/books} }  
{ {^books} }  
...  
{ {/books} }
```

If books is truthy, then first block is rendered
If books is falsy, then the second block is rendered

AJAX Enhancements

- Mustache (cont.):
 - Template can contain:
 - Includes of other templates

```
...  
{ {>header} }  
...  
...  
{ {>footer} }  
...
```

AJAX Enhancements

- There is more to Mustache
- For more information:
 - <https://github.com/janl/mustache.js>

AJAX Wrap-Up

- Each PennyAjax*N* app is a *single page app (SPA)*
- SPAs are common
 - Google docs, Gmail, Slack, Ed, ...
- SPAs are implemented using AJAX

AJAX Wrap-Up

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

AJAX Wrap-Up

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)

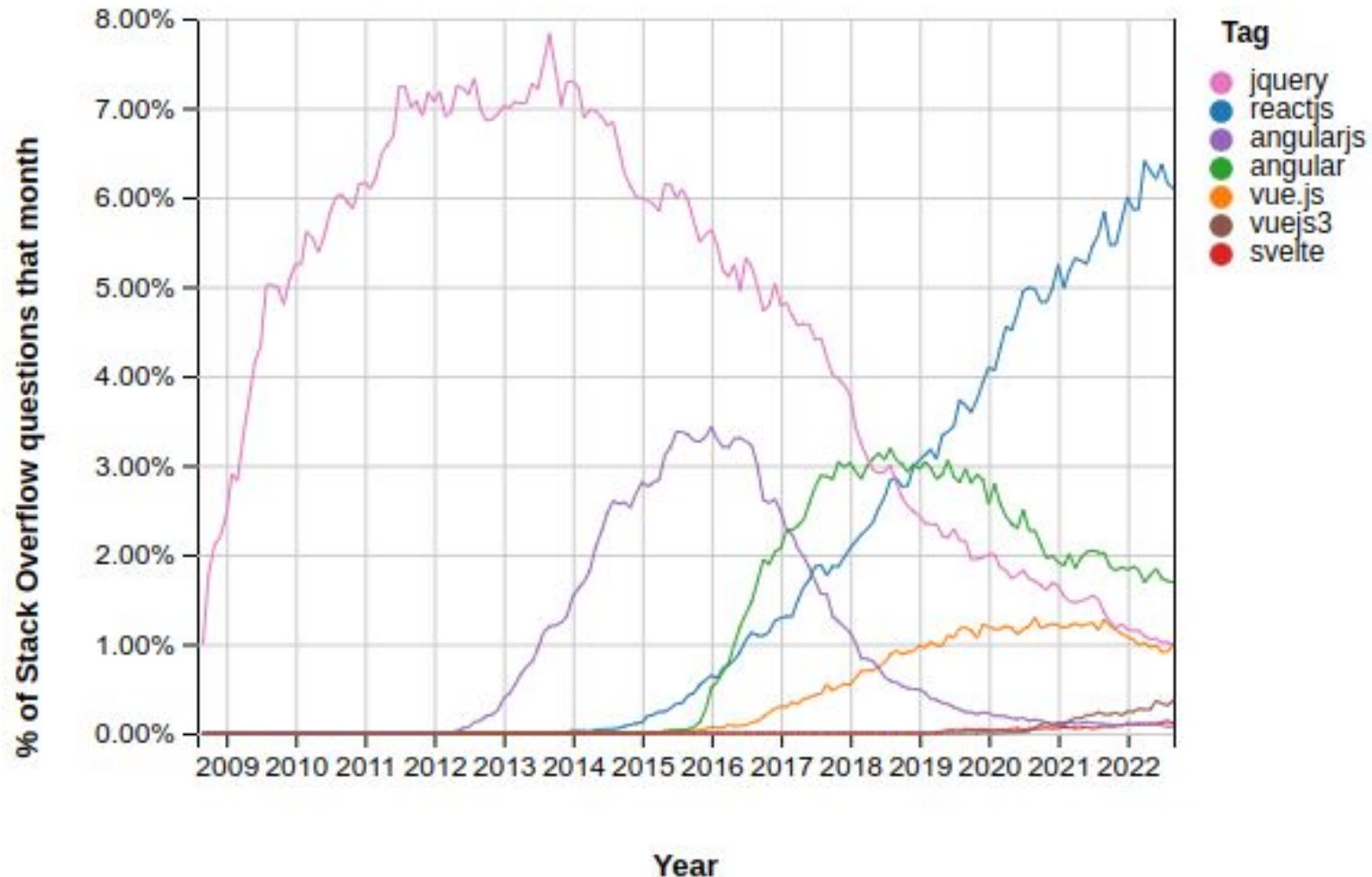
Agenda

- AJAX via XMLHttpRequest enhancements (cont.)
- **JavaScript libraries**
- jQuery

JavaScript Libraries

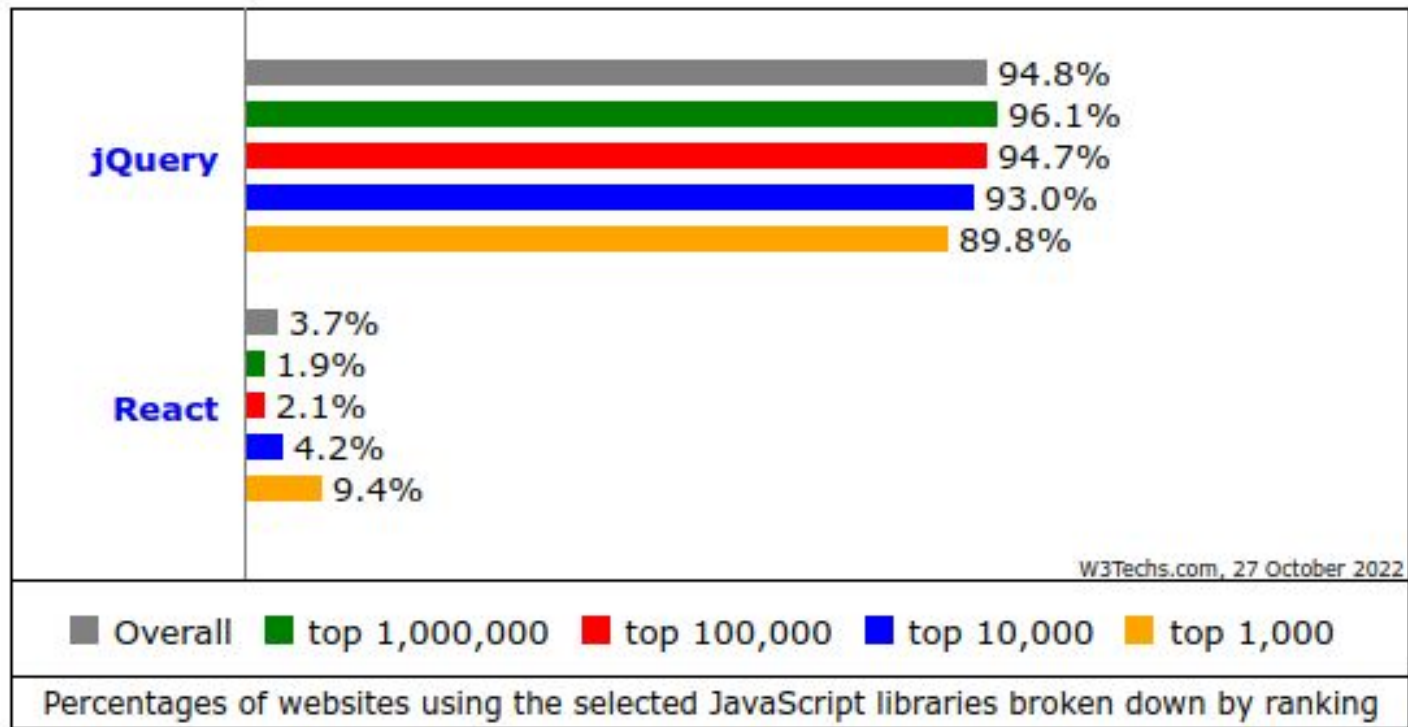
- **Problem:**
 - JavaScript/AJAX code uses common patterns and often is repetitive
- **Solution:**
 - Use a JavaScript **library**
 - **jQuery**, **React**, AngularJS, Angular, vue.js, vue3, svelte, ...

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 (cont.)
- JavaScript libraries
- **jQuery**

jQuery



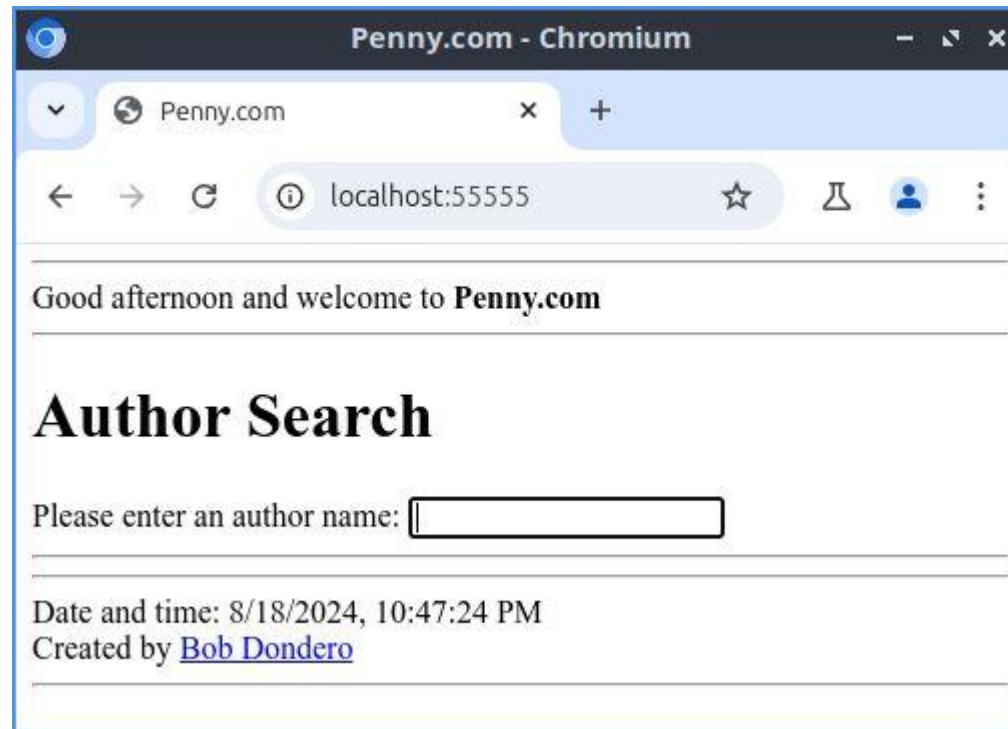
John
Resig

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)

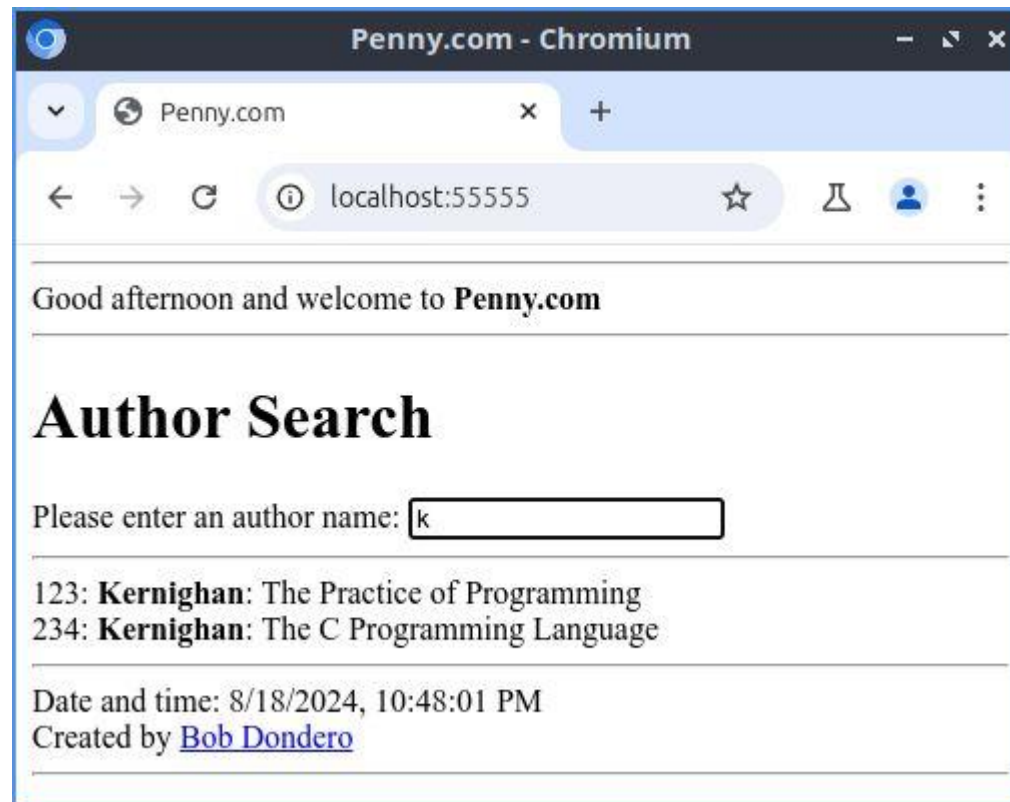
jQuery

- See **PennyjQuery** app



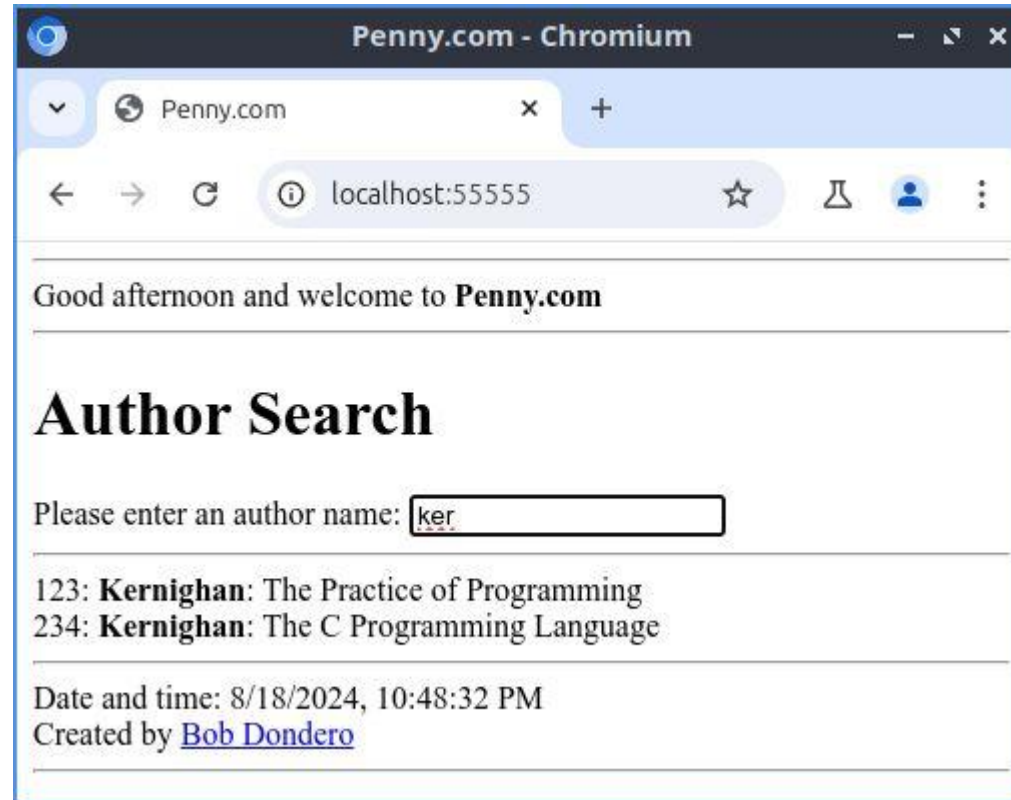
jQuery

- See **PennyjQuery** app



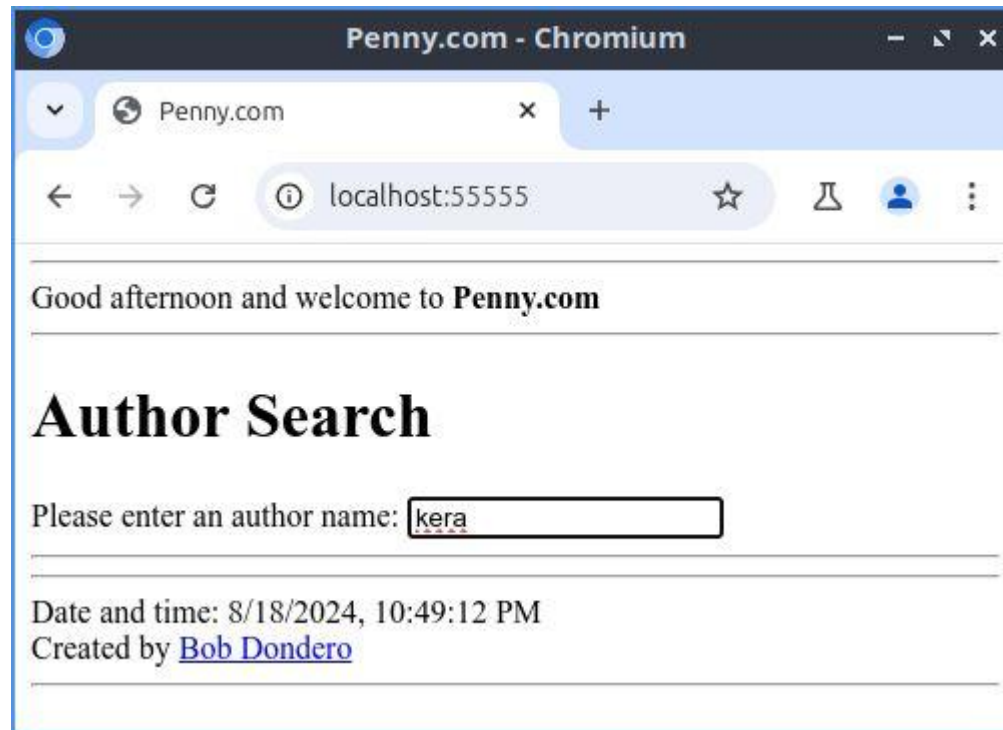
jQuery

- 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 makes accessing the DOM easier

Without jQuery:

```
let authorInput =  
    document.getElementById('authorInput');  
let author = authorInput.value;
```

With jQuery:

```
let author = $('#authorInput').val();
```

=> access by id

jQuery

- jQuery makes updating the DOM easier

Without jQuery:

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

With jQuery:

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

=> access by id

jQuery

- jQuery makes installing event handlers easier

Without jQuery:

```
let authorInput =  
  document.getElementById('authorInput');  
authorInput.addEventListener('input',  
  debouncedGetResults);
```

With jQuery:

```
$('#authorInput').on('input',  
  debouncedGetResults);
```

=> access by id

jQuery

- jQuery makes installing event handlers easier

Without jQuery:

```
window.addEventListener('beforeunload',  
    function(e) {window.clearInterval(apInterval);})
```

With jQuery:

```
$(window).on('beforeunload',  
    function(e) {window.clearInterval(apInterval);})
```

=> access by id

jQuery

- jQuery makes setup easier

Without jQuery:

```
document.addEventListener('DOMContentLoaded', setup);
```

With jQuery:

```
$( 'document' ).ready ( setup ) ;
```

=> 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 = html;
}
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)

Summary

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