THE UNIVERSITY
of ADELAIDE

Web and Database Computing

Client-side Frameworks and APIs: Introduction to Client JS Frameworks

What are Client Side Frameworks?

(and why you want to use them)

JavaScript is great ...

- Allows us to have dynamic content on client side
  - Interactive page elements
  - Respond to events e.g. mouse over menu, clicks, etc.
  - Content changes without reloading page
  - Content changes based on user, browser, location etc.

... but writing DOM code can be a pain.

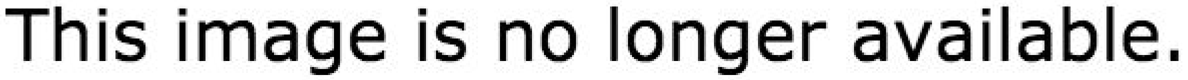
```
// Store results here
var values = [];

// Get all checkboxes
var checkboxes = document.getElementsByClassName('box');

// Check if ticked and if so add value to array
for(var i=0; i<checkboxes.length; i++){
      if(checkboxes[i].checked){
           value.push(checkboxes[i].value);
           break;
      }
}
```

Getting the values of checkboxes shouldn't be this hard!

There's gotta be a better way!



This image is no longer available.

jQuery; the original solution

```
// Store results here
var values = [];

// jQuery
$('.box:checked').each(function() { value.push($( this ).val()); });
```

- A Javascript library, designed to simplify repetetive and common Javascript tasks.
- Full of shortcuts to easily access and manipulate the DOM tree and make iteratative tasks easier.
- Fixed browser incompatibilities.
- Based on CSS selectors, similar to `document.querySelector();`

Simply load the library as a script:

```
<script src="https://code.jquery.com/jquery-3.4.0.min.js"></script>
```

jQuery; the original solution

```
$( 'css selector' )                             // Select elements

$( 'css selector' ).hide();                     // Hide/Show elements
$( 'css selector' ).show();                     // (display style)

$( 'css selector' ).text();                     // Get/set text/html content
$( 'css selector' ).html();

$( 'css selector' ).css('property');            // Get CSS content of element,
$( 'css selector' ).css('property','value');    // set for multiple

$( 'css selector' ).width();                    // Change common CSS properties
$( 'css selector' ).width('10px');
$( 'css selector' ).outerWidth('10px');

$( '<div><p>some html</p></div>' );             // Create new elements

$( 'css selector' ).append( '<p>html</p>' );    // Modify the DOM tree
```

What could we do better?

- Minimise directly changing DOM
    - Keep page elements that we want to change accessible as JavaScript objects.
    - Treat complex components composed of multiple HTML elements as a single object that can be easily manipulated.
    - Change parts of components by updating the properties of the object.
- Store page data as state information
    - Update page elements automatically when state changes.
- Use templates/placeholders to improve code reusabiility.

Introducing Client-side JS Frameworks

Client-side frameworks ...

- Provide a method of connecting elements on our page with data in our JavaScript.
    - Data is stored in a data model.
    - The page's HTML is defined using a template syntax.
    - Data from the data model is rendered into the page's HTML at runtime.
    - Changes to the data model trigger 'reactive' changes in the HTML.

Current Client-side JS Frameworks

Angular

- Developed by Google
- Original Front-end Framework
- Google, Micorsoft, PayPal, The Guardian, Nike, HBO, Sony
  Source: https://www.madewithangular.com/
- Complex to learn and use (but has gotten better), heavy (144K)

React

- Developed by Facebook
- Currently most popular framework
- Used by Facebook, Airbnb, Dropbox, Netflix, Reddit
  Source: https://madewithreact.com/
- Easier to use but still a challenge to learn, medium (117K)

Current Client-side JS Frameworks

Vue.js

- Open Source
- Gaining popularity
- Used by Baidu, Tencent,Xiaomi, DJI, Nintendo, Sainsbury's
  Source: https://madewithvuejs.com/
- Easy to learn/use & lightweight (88K)

We will be using Vue.js

Summary

- JavaScript and DOM provide allow us to build dynamic webpages, but can be cumbersome for large and complex websites.
- jQuery attempted to improve on this by providing shortcuts to common tasks.
- Reactive frameworks have become popular as a way to connect a webpage and its underlying data together.
- We will be looking at Vue.js in the next couple of lectures.

SUB CRUCE LUMEN

THE UNIVERSITY of ADELAIDE