**Exercises**

**Load External Content**

Open the file /exercises/index.html in your browser. Use the file /exercises/js/load.js. Your task is to load the content of a blog item when a user clicks on the title of the item.

1. Create a target div after the headline for each blog post and store a reference to it on the headline element using $.fn.data.
2. Bind a click event to the headline that will use the $.fn.load method to load the appropriate content from /exercises/data/blog.html into the target div. Don't forget to prevent the default action of the click event.

Note that each blog headline in index.html includes a link to the post. You'll need to leverage the href of that link to get the proper content from blog.html. Once you have the href, here's one way to process it into an ID that you can use as a selector in $.fn.load:

var href = 'blog.html#post1';

var tempArray = href.split('#');

var id = '#' + tempArray[1];

Remember to make liberal use of console.log to make sure you're on the right path!

**Load Content Using JSON**

Open the file /exercises/index.html in your browser. Use the file /exercises/js/specials.js. Your task is to show the user details about the special for a given day when the user selects a day from the select dropdown.

1. Append a target div after the form that's inside the #specials element; this will be where you put information about the special once you receive it.
2. Bind to the change event of the select element; when the user changes the selection, send an Ajax request to /exercises/data/specials.json.
3. When the request returns a response, use the value the user selected in the select (hint: $.fn.val) to look up information about the special in the JSON response.
4. Add some HTML about the special to the target div you created.
5. Finally, because the form is now Ajax-enabled, remove the submit button from the form.

Note that we're loading the JSON every time the user changes their selection. How could we change the code so we only make the request once, and then use a cached response when the user changes their choice in the select?