TOP TEN LIBRARIES

IN DRUPAL TEN CORE

Who I Am

Andy Blum

Sr Front End Developer
Lullabot

Subsystem Maintainer
Olivero

@andy_blum
drupal.community

@andy-blum
drupal.org
github.com
linkedin.com

Define: “Libraries”

Found in core.libraries.yml

Defined in core’s libraries yaml file

Defines one or more front-end assets

CSS or JavaScript files

May or may not depend on other libraries

Core libraries will only rely on other core libraries

Your theme or module can use them

You can depend on them, extend them, or override them

Define: “Libraries”

Avoid Internal Libraries

Some third-party libraries are marked “internal”. These might have major changes in minor version releases

Avoid Deprecated Libraries

There are some polyfills that aren’t needed with

01

Drupal

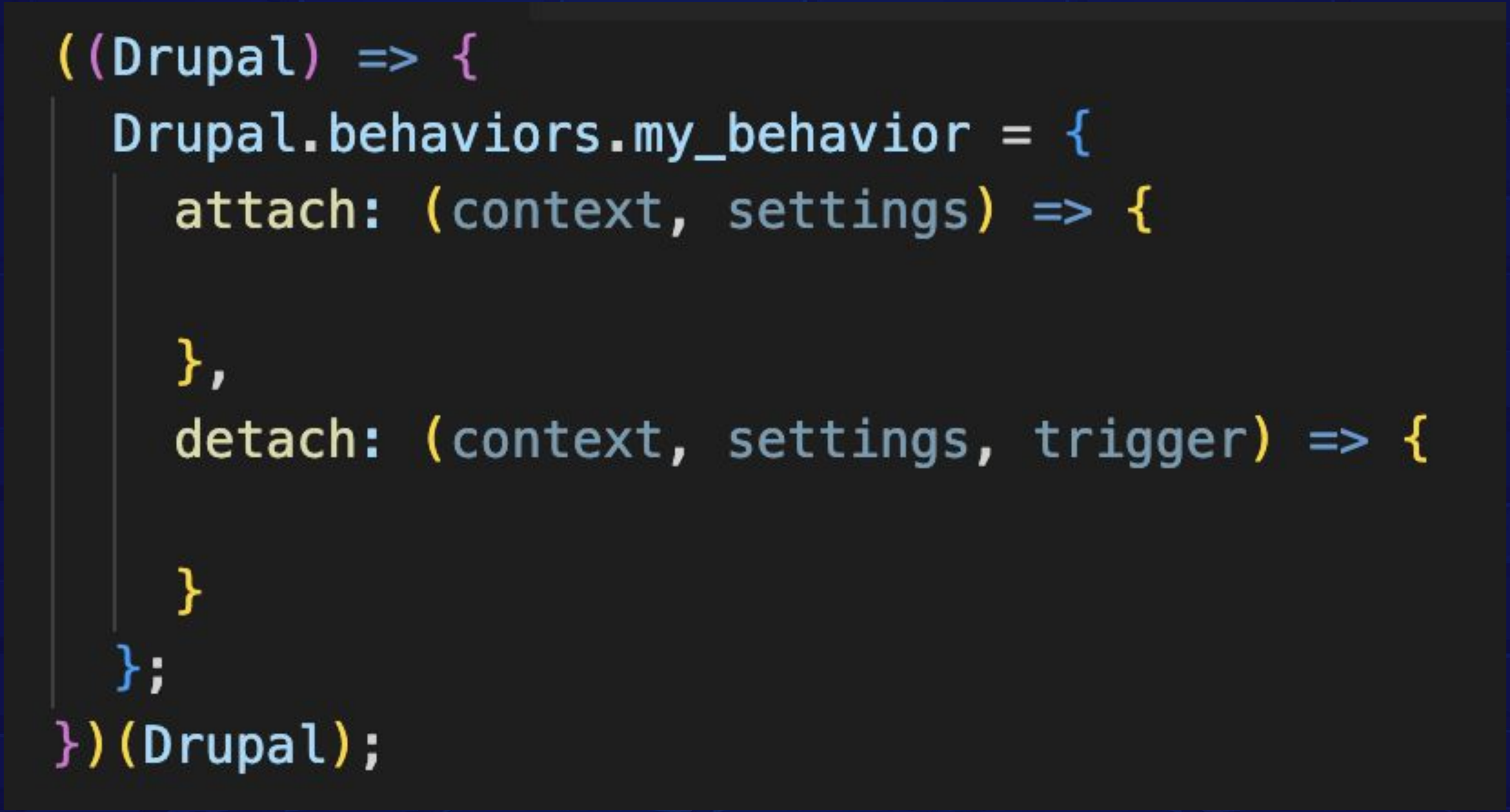
You should probably already be using this.

core/drupal

- Creates a globally-scoped object “`Drupal`”

- Allows Drupal core and contrib to keep all JS objects under a single namespace to avoid collisions

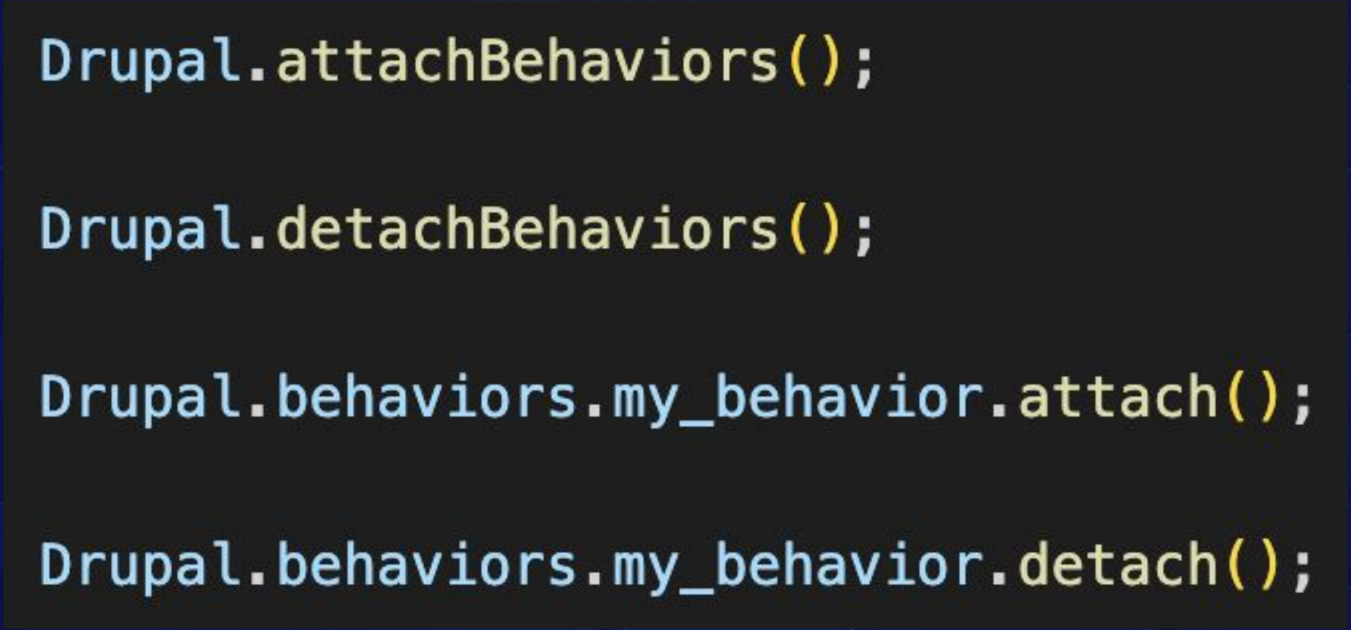
- Drupal behaviors (JavaScript functionality)
  - Behaviors are *attached* on page load and when content is added or changed
  - Behaviors can also be *detached* before content is pruned, moved, or serialized



```
((Drupal) => {
  Drupal.behaviors.my_behavior = {
    attach: (context, settings) => {

    },
    detach: (context, settings, trigger) => {

    }
  };
})(Drupal);
```



```
Drupal.attachBehaviors();

Drupal.detachBehaviors();

Drupal.behaviors.my_behavior.attach();

Drupal.behaviors.my_behavior.detach();
```

core/drupal

demo

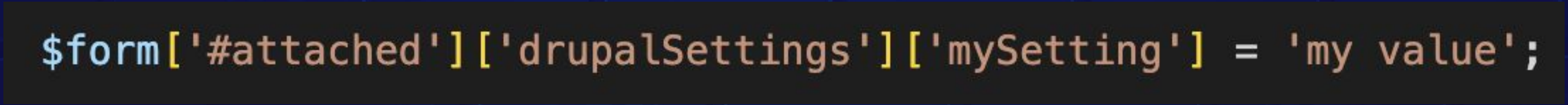
02

DrupalSettings

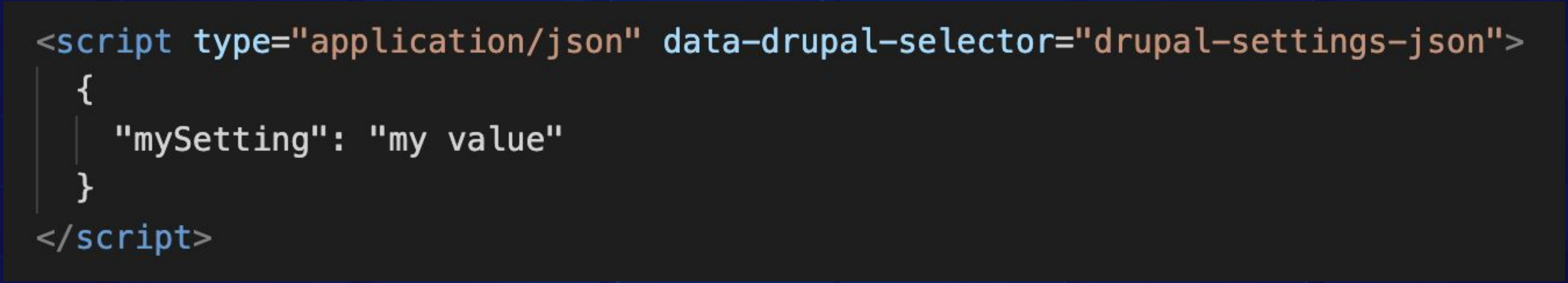
Getting backend data
to the frontend

core/drupalSettings

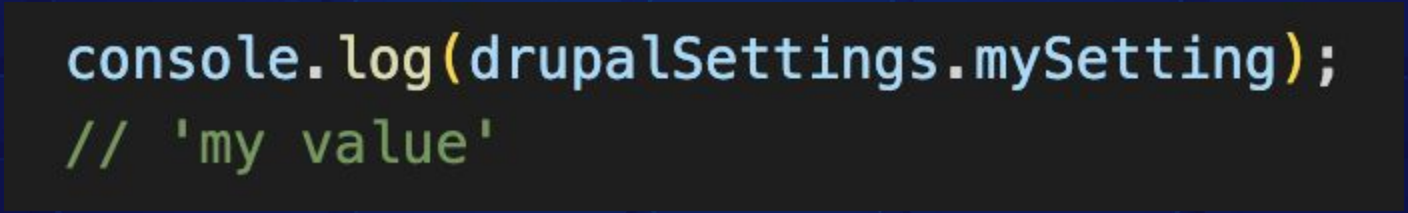
- Creates a second globally-scoped object
- On page load, searches for a specific script element
- `<script data-drupal-selector="drupal-settings-json">`
- Parses the JSON within, and stores it in `drupalSettings`
- Serves as a way to pass data from Drupal's backend to the frontend



```php
$form['#attached']['drupalSettings']['mySetting'] = 'my value';
```



```html
<script type="application/json" data-drupal-selector="drupal-settings-json">
  {
    "mySetting": "my value"
  }
</script>
```



```javascript
console.log(drupalSettings.mySetting);
// 'my value'
```

core/drupalSettings

demo

03

Normalize

Starting Styles
Off Right.

core/normalize

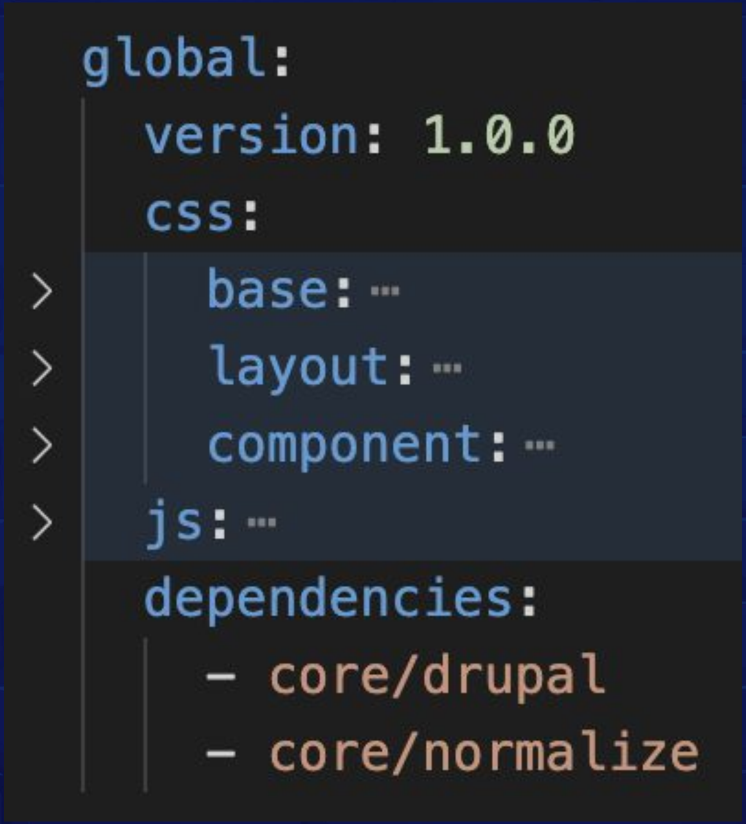
- Implements the widely-used normalize.css file

- Used in Claro, Umami, Stable9, and Starterkit themes

- “Makes browsers render all elements more consistently and in line with modern standards. It precisely targets only the styles that need normalizing.”

- Make it a dependency of your theme’s global styles
  - Avoid putting it in modules

core/normalize



```
global:
  version: 1.0.0
  css:
    base: ⋯
    layout: ⋯
    component: ⋯
  js: ⋯
  dependencies:
    - core/drupal
    - core/normalize
```

core/normalize

demo

04

Debounce

Handling events performantly

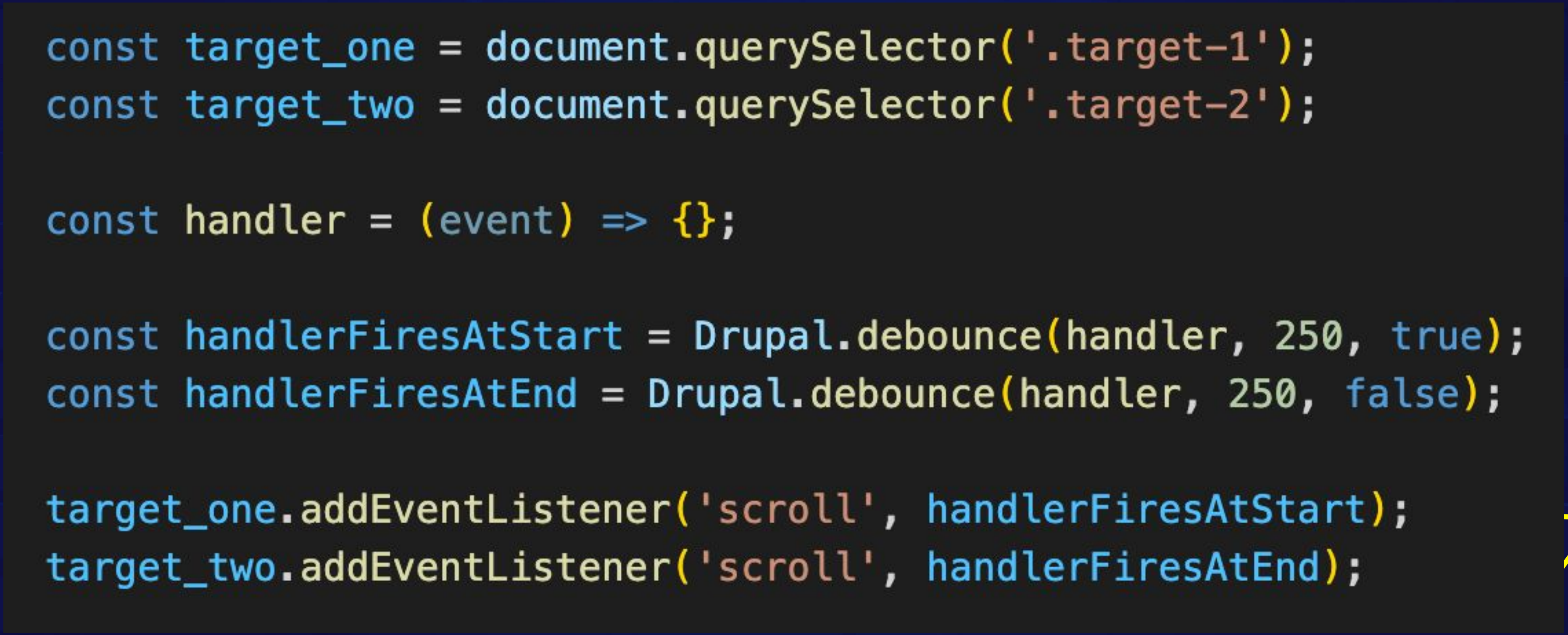
core/drupal.debounce

- Some events can happen very rapidly

- More performant to act when a rapid sequence starts or ends instead of on every event.

- Drupal.debounce() returns a function that you can use as an event handler
    - Still runs on every event, but we can prevent thread-blocking operations

core/drupal.debounce



```
const target_one = document.querySelector('.target-1');
const target_two = document.querySelector('.target-2');

const handler = (event) => {};

const handlerFiresAtStart = Drupal.debounce(handler, 250, true);
const handlerFiresAtEnd = Drupal.debounce(handler, 250, false);

target_one.addEventListener('scroll', handlerFiresAtStart);
target_two.addEventListener('scroll', handlerFiresAtEnd);
```

core/drupal.debounce

demo

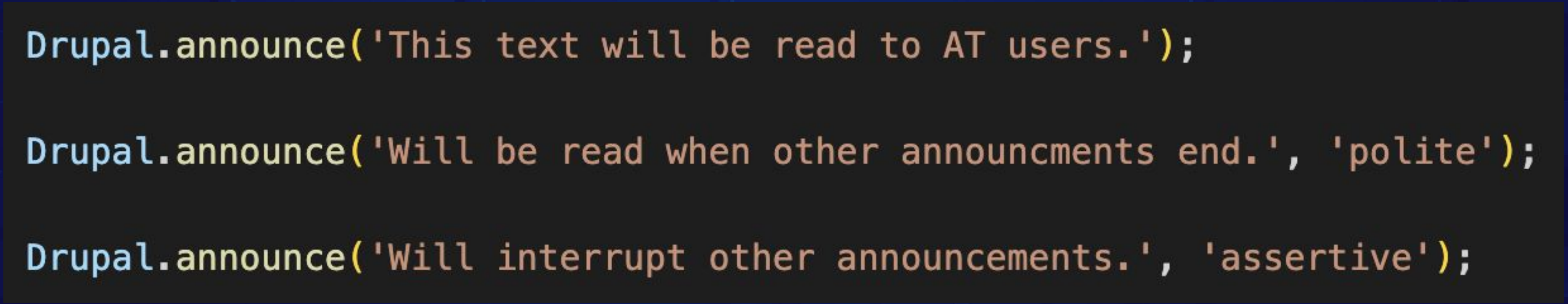
05

Announce

Making client-side
updates accessible

core/drupal.announce

- Provides a simple-to-use function that creates & updates an aria-live region
    - aria-live regions are read out by screen readers when their content changes
    - Provides a way to share content changes with AT users they might otherwise miss



```
Drupal.announce('This text will be read to AT users.');

Drupal.announce('Will be read when other announcments end.', 'polite');

Drupal.announce('Will interrupt other announcements.', 'assertive');
```

core/drupal.announce

demo

06

Displace

Organizing layouts with fixed content

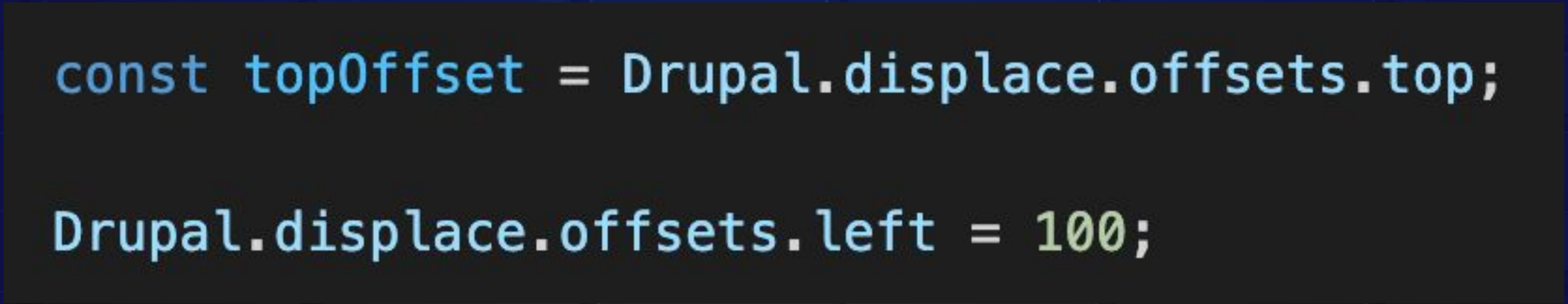
core/drupal.displace

- Creates an API to create, store, and retrieve offset values

- Helps prevent hiding content under fixed-position elements docked to edge of viewport
  - e.g. Admin Toolbar

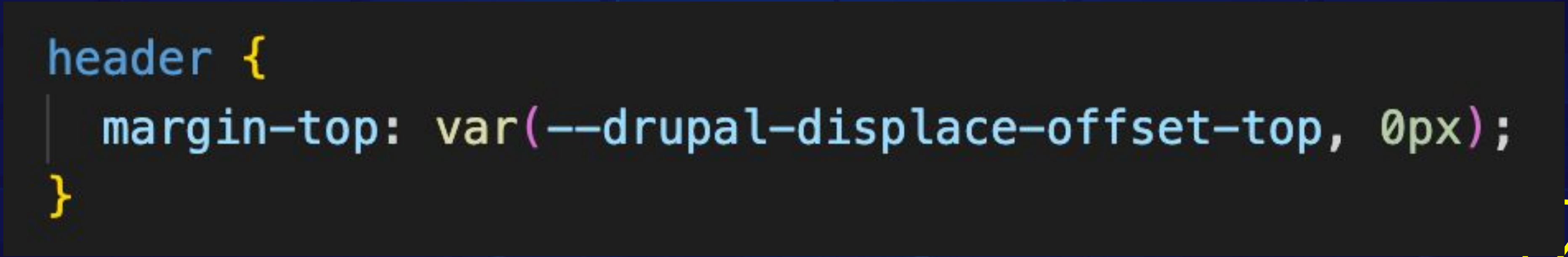
- In addition to storing values in JS, creates a CSS variable
  - `--drupal-displace-offset-top: 100px;`

core/drupal.displace



```javascript
const topOffset = Drupal.displace.offsets.top;

Drupal.displace.offsets.left = 100;
```



```css
header {
  margin-top: var(--drupal-displace-offset-top, 0px);
}
```

core/drupal.displace

demo

07

Message

Alerting users in
a uniform way

core/drupal.message

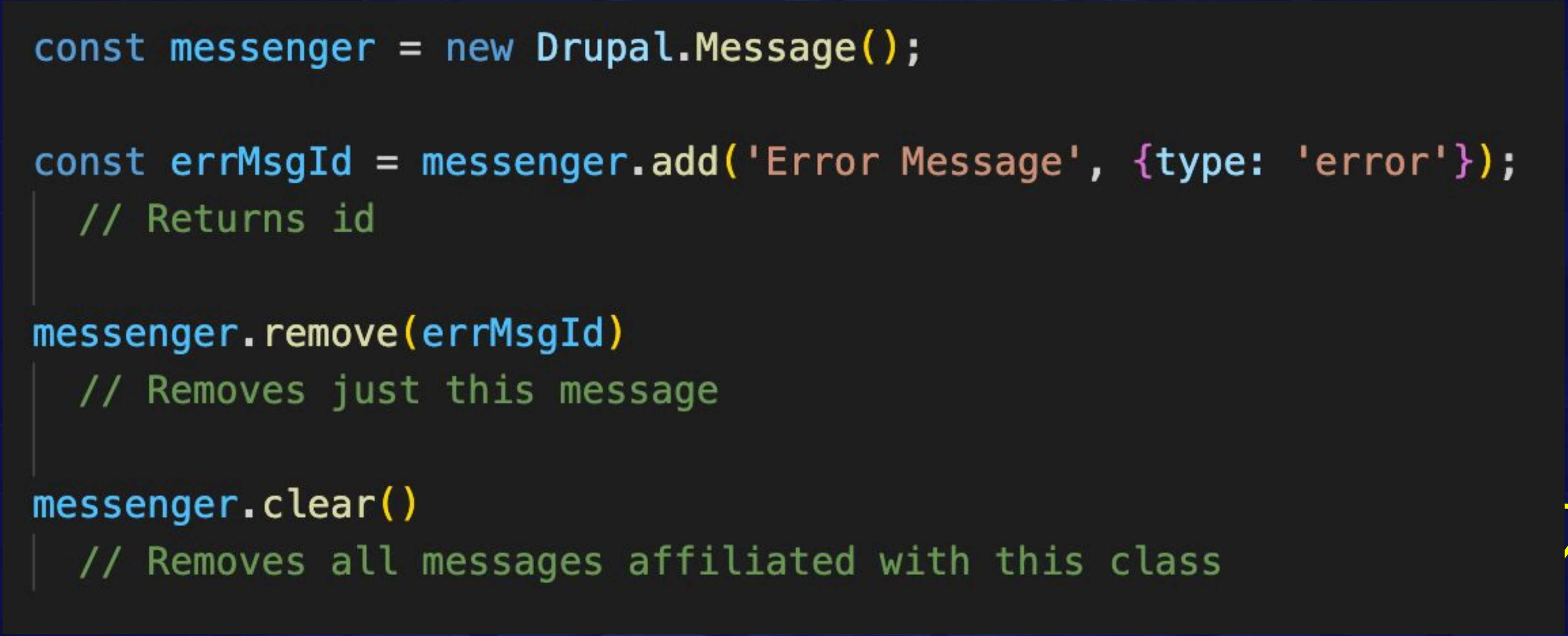
- Creates a JS class that can be used to insert messages to the DOM that match the current theme

- Messages can be added to the messages block, or to another location in the DOM

- Messages can be added/removed/cleared

- Messages can be typed as status/warning/error

core/drupal.message



```
const messenger = new Drupal.Message();

const errMsgId = messenger.add('Error Message', {type: 'error'});
  // Returns id

messenger.remove(errMsgId)
  // Removes just this message

messenger.clear()
  // Removes all messages affiliated with this class
```

core/drupal.message

demo

08

Once

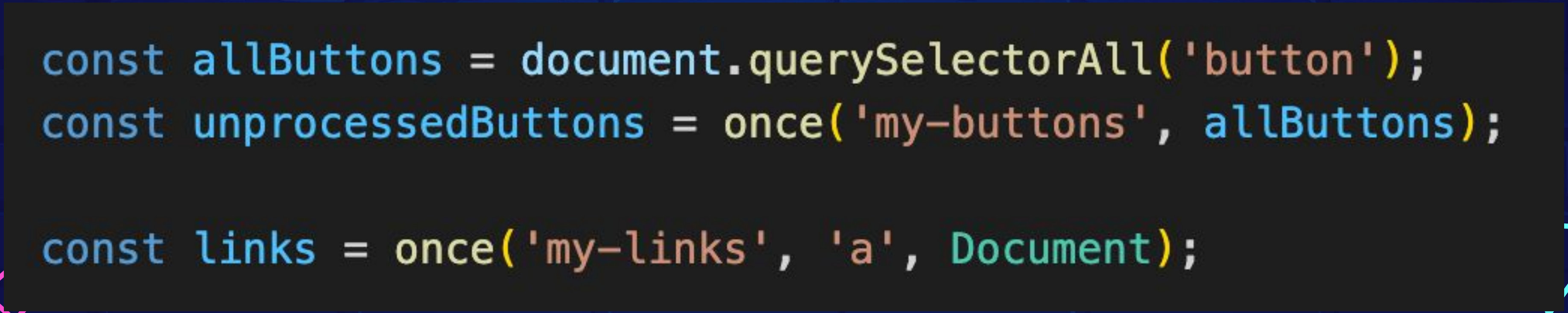
Doing things and then never doing them again

core/once

- Creates function that can be used to select items that have not been selected before
- Returns an array of selected items
- After an item has been selected, it is marked with the attribute “data-once”
- Items can be selected from “context”

core/once

- Multiple ways to select
    - Using a selector string
    - Passing reference to a Node (querySelector())
    - Passing reference to a NodeList (querySelectorAll())
    - Passing reference to an Array of Nodes ($())



```
const allButtons = document.querySelectorAll('button');
const unprocessedButtons = once('my-buttons', allButtons);

const links = once('my-links', 'a', Document);
```

core/once

demo

09

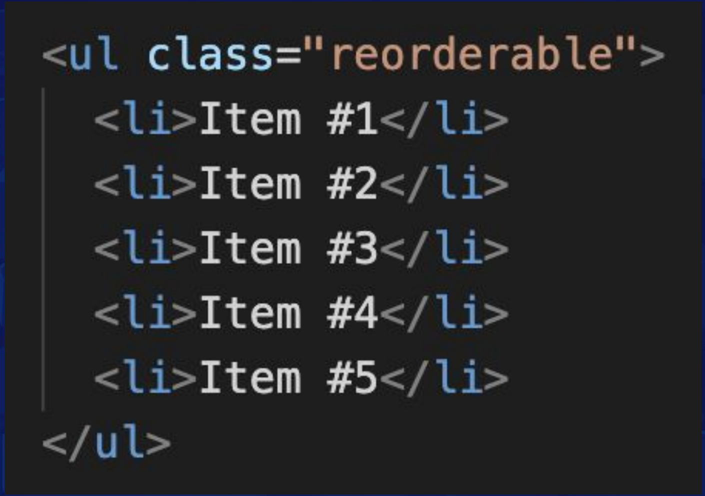
Sortable

Move things around

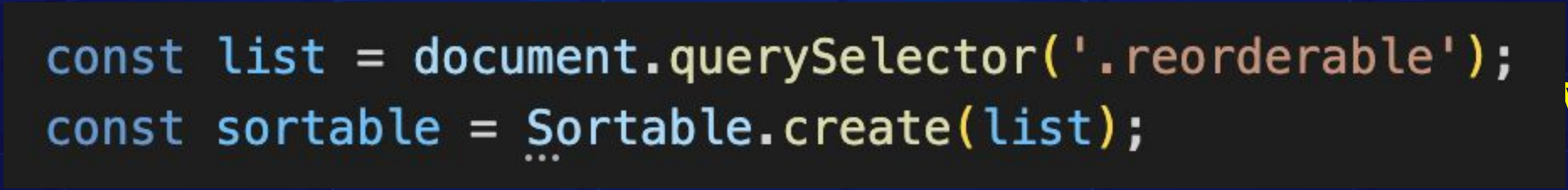
core/sortable

- Imports the SortableJS library for drag-and-drop lists
- Used by CKEditor5, Layout Builder, and the Media Library
- Has a huge variety of options & configurations
  - SortableJS
  - Better documentation in the repository

core/sortable



```html
<ul class="reorderable">
  <li>Item #1</li>
  <li>Item #2</li>
  <li>Item #3</li>
  <li>Item #4</li>
  <li>Item #5</li>
</ul>
```



```javascript
const list = document.querySelector('.reorderable');
const sortable = Sortable.create(list);
```

core/sortable

demo

10

Tabbable

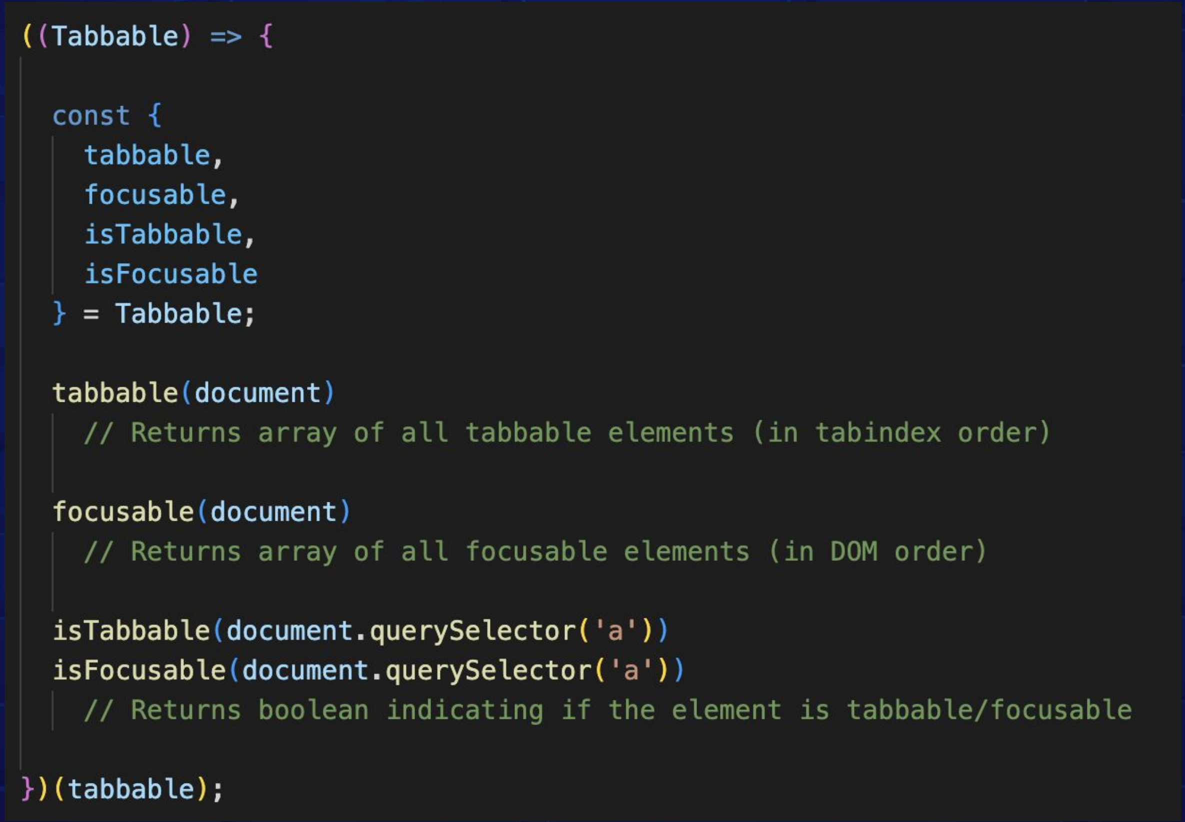
Finding items that
are keyboard focusable

core/tabbable

- Provides 4 utility functions
  - tabbable()
  - focusable()
  - isTabbable()
  - isFocusable()

- There are config options available in the repo’s readme

core/tabbable



```
((Tabbable) => {

  const {
    tabbable,
    focusable,
    isTabbable,
    isFocusable
  } = Tabbable;

  tabbable(document)
    // Returns array of all tabbable elements (in tabindex order)

  focusable(document)
    // Returns array of all focusable elements (in DOM order)

  isTabbable(document.querySelector('a'))
  isFocusable(document.querySelector('a'))
    // Returns boolean indicating if the element is tabbable/focusable

})(tabbable);
```

core/tabbable

demo

Thank You!

Andy Blum

Sr Front End Developer
Lullabot

Subsystem Maintainer
Olivero

@andy_blum
drupal.community

@andy-blum
drupal.org
github.com
linkedin.com