<SELECT>ING GOOD EMBER PATTERNS

Brenna O'Brien / @brnnbrn

front-end engineer / TED

<select>

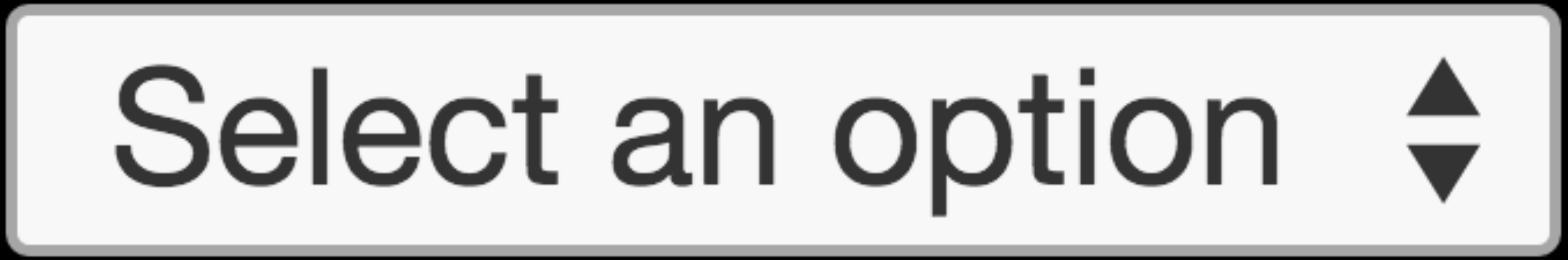
<select>

<SELECT> C. 2015

```
{{view “select”
  content=options
  value=selectedOption}}
```

<SELECT> C. 2015

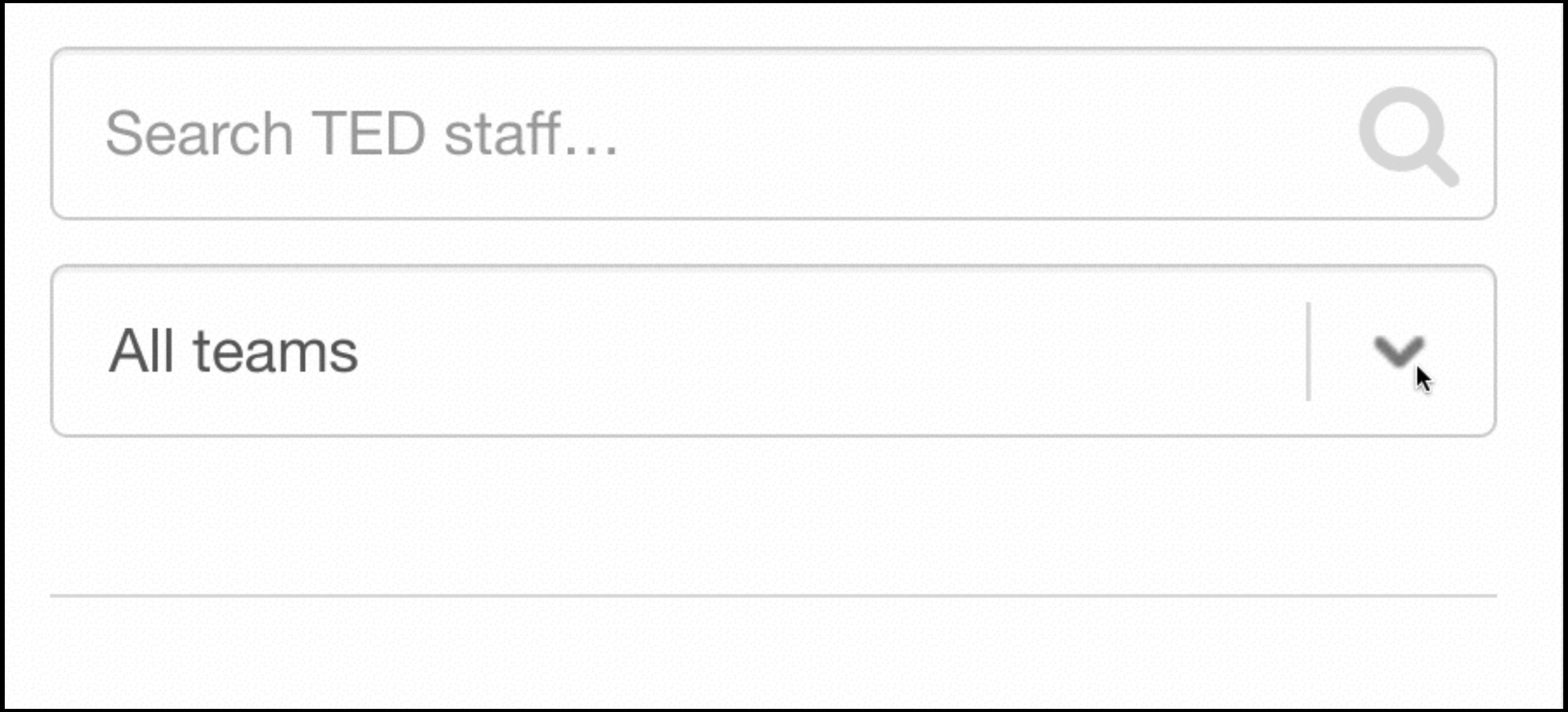
✨



BEWARE OF MAGIC

2-WAY BINDING

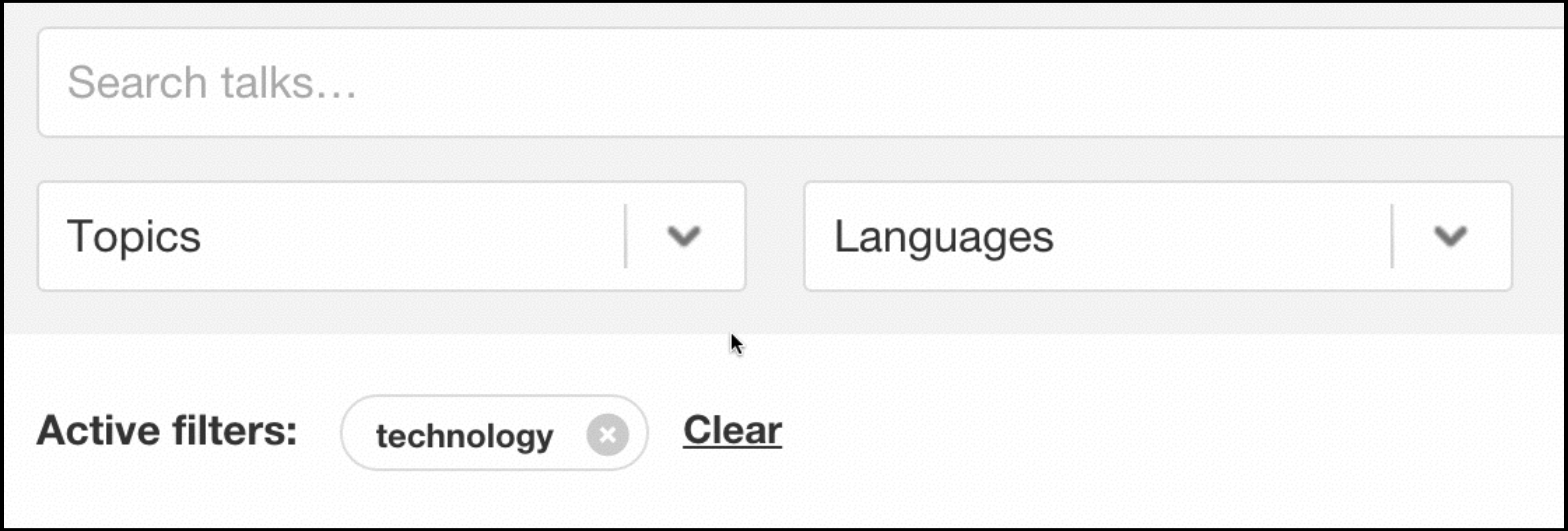
2-WAY BINDING



Search TED staff…

All teams

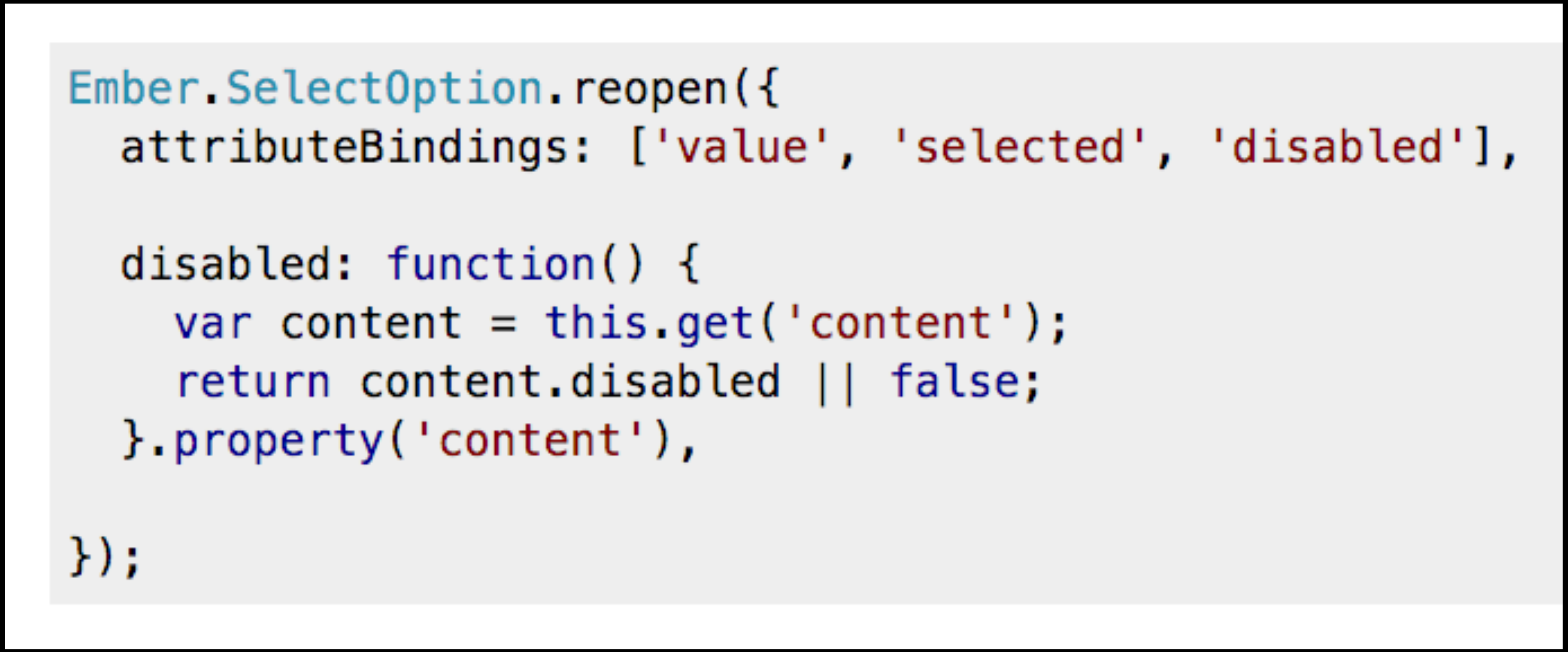
2-WAY BINDING?



Active filters: technology Clear

WHERE’S THE DOM?

“A SIMPLE APPROACH” FOR DISABLED OPTIONS



```
Ember.SelectOption.reopen({
  attributeBindings: ['value', 'selected', 'disabled'],

  disabled: function() {
    var content = this.get('content');
    return content.disabled || false;
  }.property('content'),

});
```

http://stackoverflow.com/questions/17287988/how-to-add-disabled-attribute-to-options-in-ember-select

> “Select elements will be the death of me.”

JUNE 2015

“🔥 ceremonial burning of the {{view "select"}} helper 🔥”

OCTOBER 2015

<SELECT> THE BETTER WAY

1. REAL HTML
2. DATA DOWN, ACTIONS UP
3. COOL NEW EMBER FEATURES

NOW WITH MORE HTML

```
<select>
  <option value=“”>select an option</option>

  {{#each languages as |language|}}
    <option value={{language.code}}>
      {{language.name}}
    </option>
  {{/each}}

</select>
```

DATA DOWN, ACTIONS UP

(DDAU)

DATA-DOWN, ACTIONS UP

initial selection

onchange event

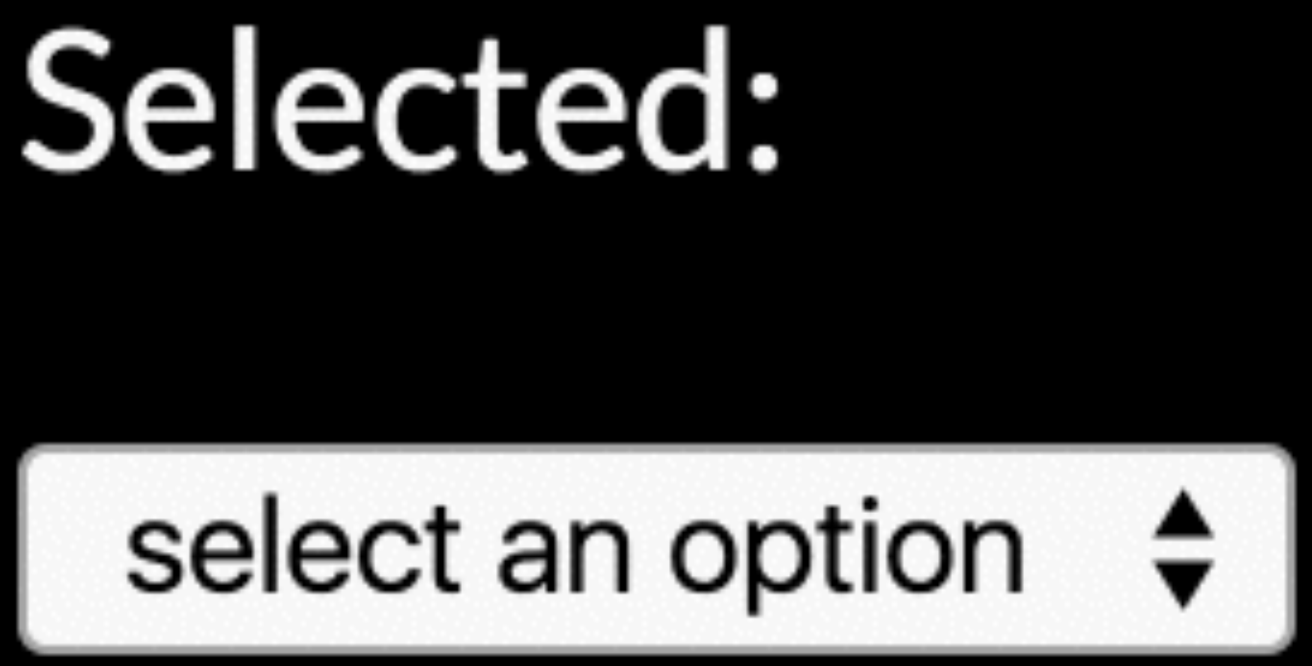
ACTIONS-UP

```
<select onchange={{action “languageDidChange”
value=“target.value”}}>

  <option value=“”>select an option</option>

  {{#each languages as |language|}}
    <option value={{language.code}}>
      {{language.name}}
    </option>
  {{/each}}

</select>
```

ACTIONS-UP

```
// controller.js

actions: {
  languageDidChange(value) {
    this.set('selectedLanguageCode', value);
  }
},

selectedLangaugeCode: null
```

ACTIONS UP



Selected:

DATA-DOWN

```
<select onchange={{action “languageDidChange”
value=“target.value”}}>
  <option value=“”>select an option</option>

  {{#each languages as |language|}}
    <option
       value={{language.code}}
       selected={{is-equal language selectedLanguage}}>
      {{language.name}}
    </option>
  {{/each}}

</select>
```

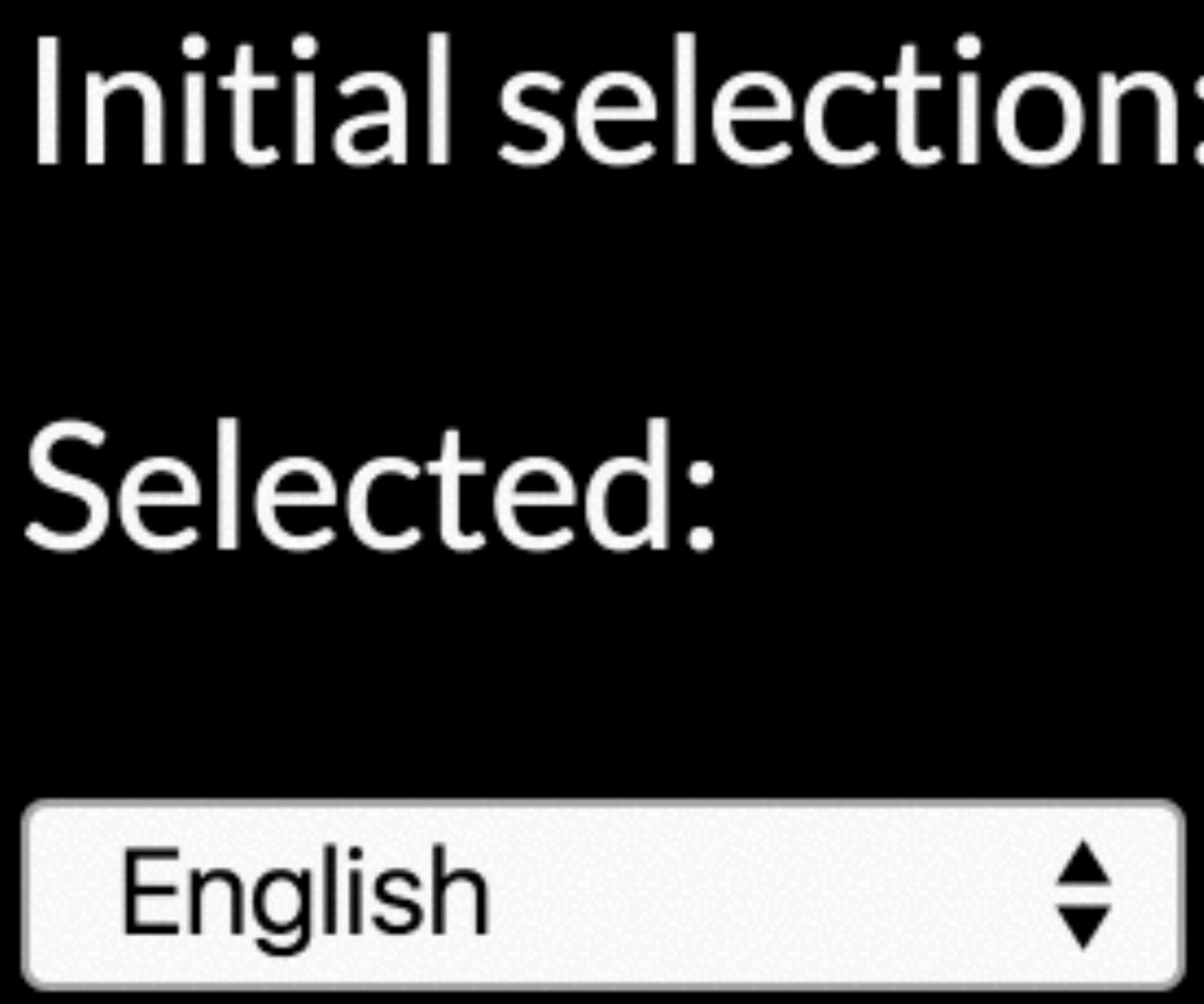
DATA-DOWN

```
<select onchange={{action “languageDidChange”
value=“target.value”}}>
  <option value=“”>select an option</option>

  {{#each languages as |language|}}
    <option
       {{bind-attr
         value=“language.code”
         selected=“¯\_(ツ)_/¯”}}>
      {{language.name}}
    </option>
  {{/each}}
```

DYNAMIC ATTRIBUTES

as of Ember 1.10, `bind-attr` deprecated in 1.13

DATA-DOWN, ACTIONS UP



Initial selection: en

Selected:

English

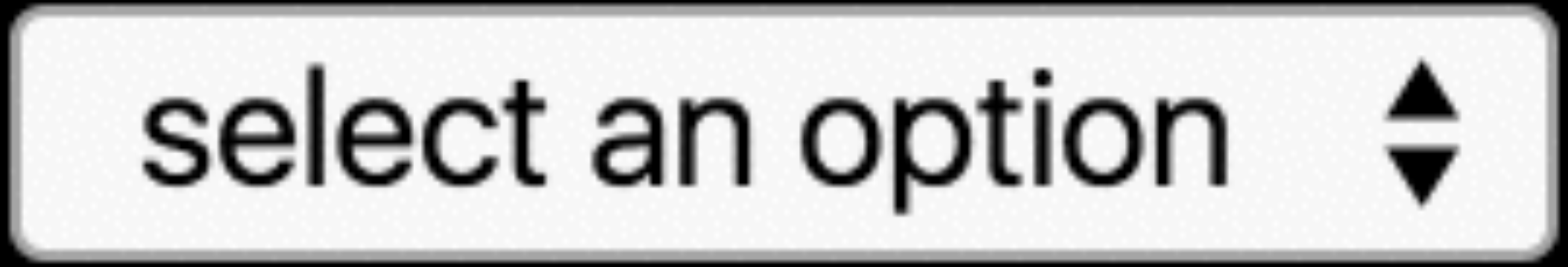
DISABLED OPTIONS

```
// models/fruit.js

export default DS.Model.extend({
  name: DS.attr('string'),
  colour: DS.attr('string'),

  isYellow: Ember.computed.equal('colour', 'yellow')
});
```

DISABLED OPTIONS

```
<select onchange={{action 'fruitDidChange' value='target.value'}}>
  <option value=“”>select an option</option>

  {{#each fruits as |fruit|}}
    <option
      disabled={{fruit.isYellow}}
      selected={{is-equal fruit initialSelection}}
      value={{fruit.id}}>
      {{fruit.name}}
    </option>
  {{/each}}

</select>
```

DISABLED OPTIONS



select an option

💁🚫

FLEXIBLE :)

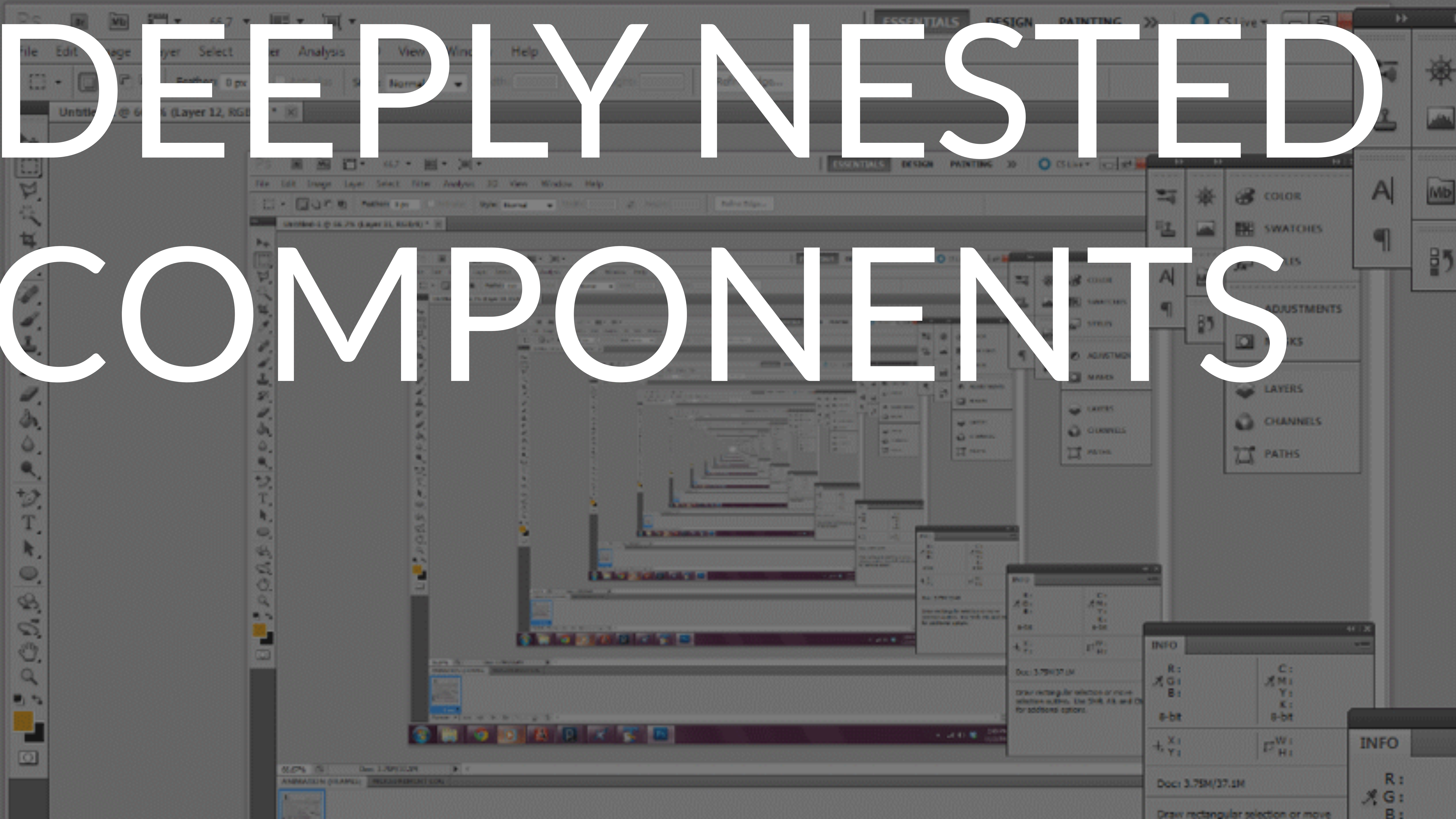
REUSABLE :(

COMPONENTS

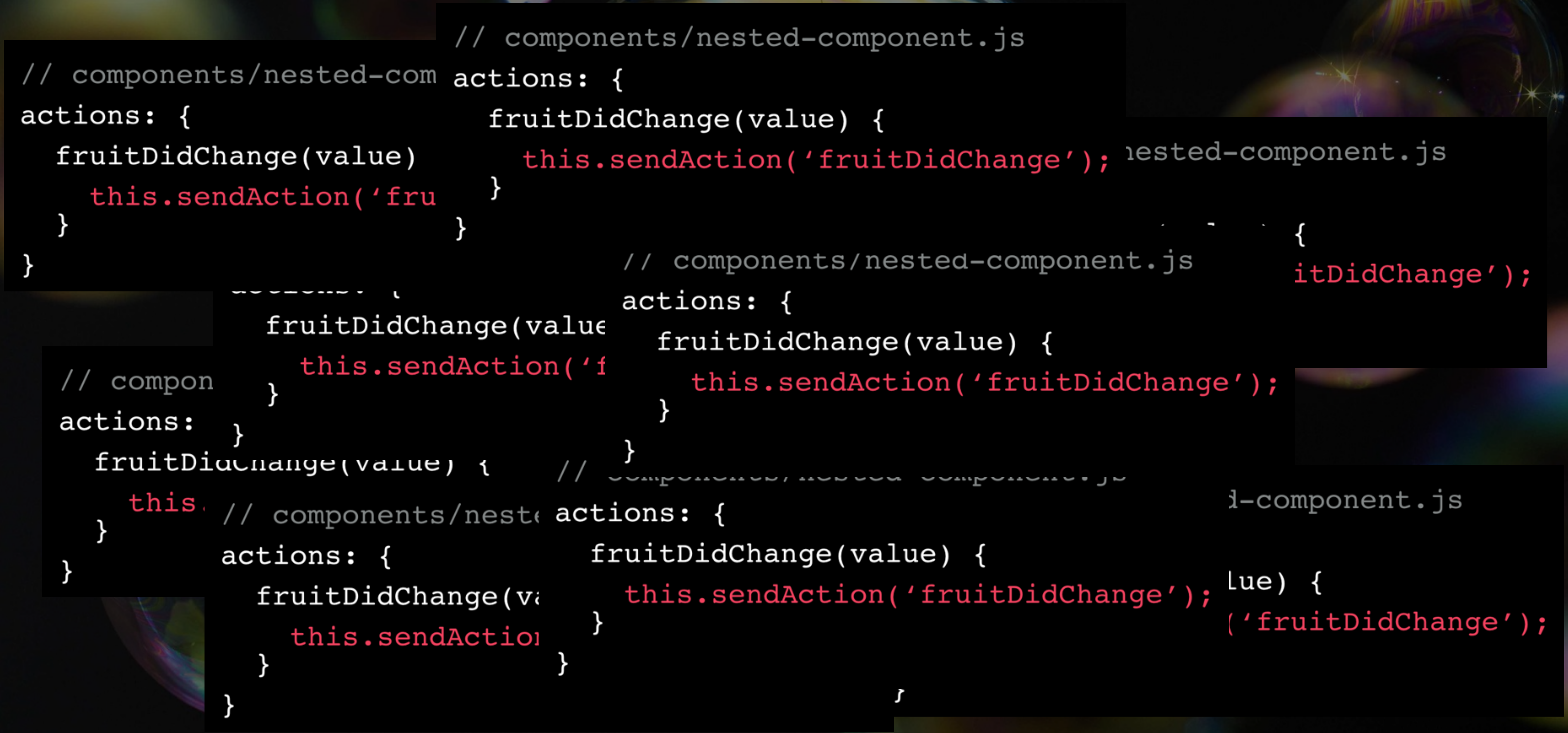
<SELECT> COMPONENT, ACTIONS-UP

```
// components/fruit-select.hbs
<select onchange={{action ‘fruitDidChange' value=‘target.value’}}>

// components/fruit-select.js
actions: {
 fruitDidChange(value) {
   this.sendAction(‘fruitDidChange’, selection);
 }
}

// templates/some-route.hbs
{{fruit-select fruitDidChange=“fruitDidChange”}}

// controllers/some-route.js
actions: {
  fruitDidChange(value) { // deal with it }
}
```

DEEPLY NESTED COMPONENTS



BUBBLING ACTIONS



```
// components/nested-com
actions: {
  fruitDidChange(value)
    this.sendAction(‘fru
  }
}
```

```
// components/nested-component.js
actions: {
  fruitDidChange(value) {
    this.sendAction(‘fruitDidChange’);
  }
}
```

```
ıested-component.js

                    {
itDidChange’);
```

```
  fruitDidChange(value
    this.sendAction(‘f
  }
}
```

```
// components/nested-component.js
actions: {
  fruitDidChange(value) {
    this.sendAction(‘fruitDidChange’);
  }
}
```

```
// compon
actions:
  fruitDidChange(value) {
    this
  }
}
```

```
// components/neste
actions: {
  fruitDidChange(va
    this.sendActio
  }
}
```

```
actions: {
  fruitDidChange(value) {
    this.sendAction(‘fruitDidChange’);
  }
}
```

```
d-component.js

lue) {
(‘fruitDidChange’);
```

CLOSURE ACTIONS

available in Ember 1.13 and up

CLOSURE ACTIONS

```
// components/fruit-select.hbs
<select onchange={{action onchange value=‘target.value’}}>

// components/fruit-select.js
actions: {
  // nothing to see here
}

// templates/some-route.hbs
{{fruit-select onchange=(action “updateFruit”)}}

// controllers/some-route.js
actions: {
  updateFruit(fruit) { // deal with it }
}
```

CLOSURE ACTIONS

```
// components/fruit-select.hbs
<select onchange={{action onchange value=‘target.value’}}>

// components/fruit-select.js
actions: {
  // nothing to see here
}

// components/grocery-order-form.hbs
{{fruit-select onchange=onchange}}

// templates/some-route.hbs
{{grocery-order-form onchange=(action “updateFruit”)}}

// controllers/some-route.js
actions: {
  updateFruit(fruit) { // deal with it }
}
```

<SELECT> COMPONENT, DATA DOWN

```
// components/fruit-select.hbs

{{#each fruits as |fruit|}}

  <option

    disabled={{fruit.isYellow}}

    selected={{is-equal fruit selectedFruit}}

    value={{fruit.id}}>

    {{fruit.name}}

  </option>

{{/each}}
```

```
// components/grocery-order-form.hbs

{{fruit-select

  onchange=(action ‘updateFruit’)

  fruits=fruits

  selectedFruit=selectedFruit }}
```

<SELECT> COMPONENT, DATA DOWN

```
// components/fruit-select.hbs
{{#each fruits as |fruit|}}
  <option
    disabled={{fruit.isYellow}}
    selected={{is-equal fruit selectedFruit}}
    value={{fruit.id}}>
    {{fruit.name}}
  </option>
{{/each}}
```

```
// components/grocery-order-form.hbs
{{fruit-select
  onchange=(action ‘updateFruit’)
  fruits=fruits
  selectedFruit=selectedFruit
  disabledKey=“isRed” }}
```

DYNAMIC KEY LOOKUP?

```
// components/fruit-select.hbs

{{#each fruits as |fruit|}}

  {{fruit-select-option

    fruit=fruit

    disabledKey=disabledKey

    selectedFruit=selectedFruit}}

{{/each}}
```

```
// components/fruit-select-option.js

tagName: ‘option’,

attributeBindings: [‘value’, ‘disabled’,’selected’],

value: Ember.computed.alias(‘fruit.id’),

disabled: Ember.computed('fruit', 'disabledKey',
  function(){
    return Ember.get(this.get(‘fruit’),this.get('disabledKey'));
  }
),

selected: Ember.computed('selectedFruit', ‘fruit',
  function(){
    return this.get('selectedFruit') === this.get('fruit');
  }
)
```

{{GET}} HELPER

available in Ember 2.1 and up

<SELECT> COMPONENT, DATA DOWN

```
// components/fruit-select.hbs

{{#each fruits as |fruit|}}

  <option

    disabled={{get fruit disabledKey}}

    selected={{is-equal fruit selectedFruit}}

    value={{fruit.id}}>

    {{fruit.name}}

  </option>

{{/each}}
```

```
// components/grocery-order-form.hbs

{{fruit-select

  onchange=(action ‘updateFruit’)

  fruits=fruits

  selectedFruit=selectedFruit

  disabledKey=“isRed” }}
```

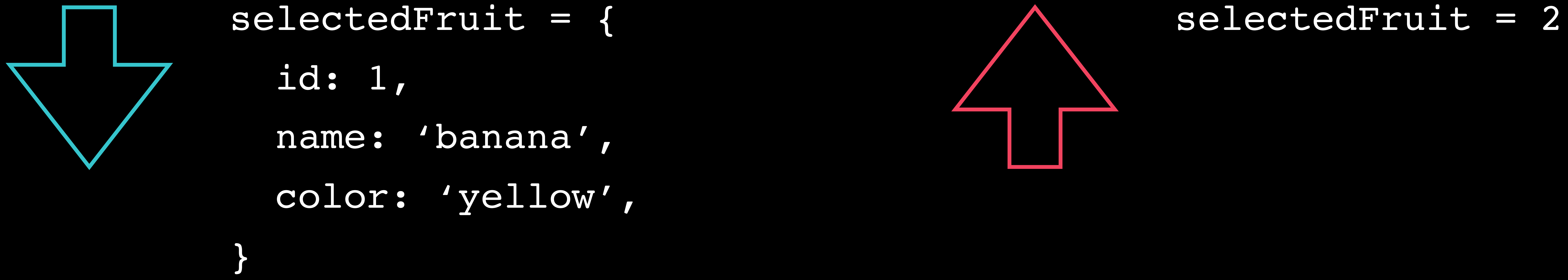
I WANT MY 2-WAY BINDING

MUT HELPER

available in Ember 1.13 and up, 2-way bindings with closure actions

MUT HELPER

```
{{fruit-select
  onchange=(action (mut selectedFruit))
  fruits=fruits
  selectedFruit=selectedFruit}}
```



```
selectedFruit = {
  id: 1,
  name: ‘banana’,
  color: ‘yellow’,
}
```

```
selectedFruit = 2
```

MUT HELPER

```
{{fruit-select
  onchange=(action (mut selectedFruit))
  fruits=fruits
  selectedFruit=selectedFruit}}
```

↓

```
selectedFruit = {
  id: 1,
  name: ‘banana’,
  color: ‘yellow’,
}
```

↑

```
selectedFruit = {
  id: 2,
  name: ‘blueberry’,
  color: ‘blue
}
```

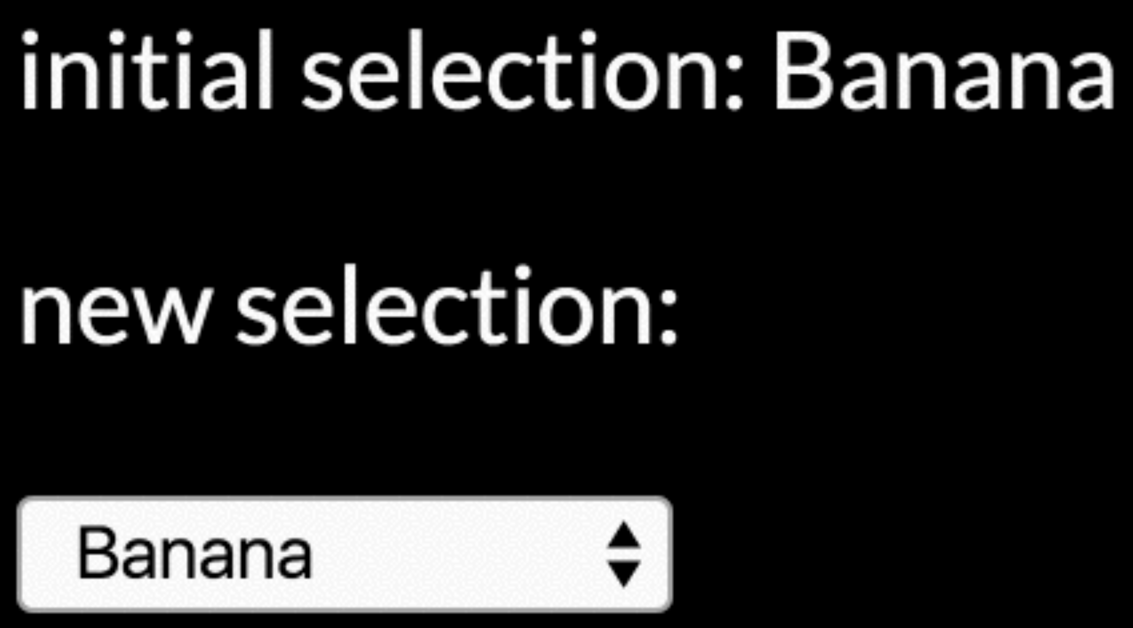
MUT HELPER

```
{{fruit-select
  onchange=(action (mut selectedFruit))
  fruits=fruits
  selectedFruit=selectedFruit}}

// components/fruit-select.hbs
<select onchange={{action ‘fruitDidChange’ value=‘target.value’}}>

// components/fruit-select.js
actions: {
  fruitDidChange(value) {
    let selection = this.get('fruits').findBy('id', +value);
    this.get(‘onchange’)(selection);
  }
}
```

CHOOSE YOUR OWN BINDING ADVENTURE

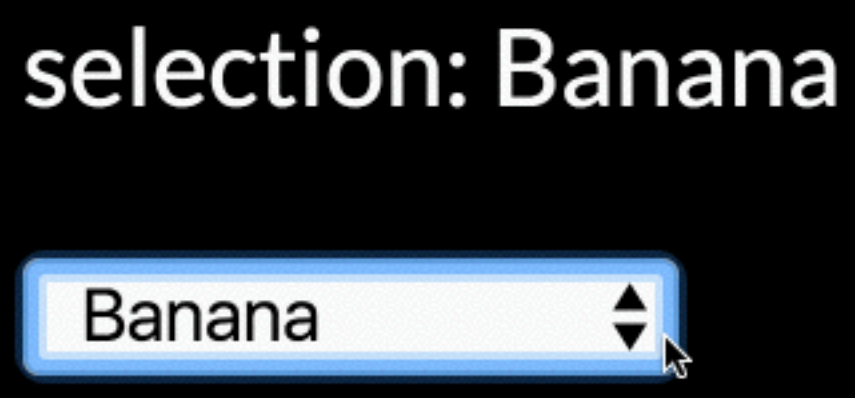
```
{{fruit-select
   onchange=(action ‘updateFruit’)
   fruits=fruits
   selectedFruit=selectedFruit}}
```

```
{{fruit-select
   onchange=(action (mut selectedFruit))
   fruits=fruits
   selectedFruit=selectedFruit}}
```



initial selection: Banana

new selection:



selection: Banana

- ✓ CUSTOM HTML
- ✓ CUSTOM DATA FLOW
- ✓ DYNAMIC ATTRS, CLOSURE ACTIONS, {{GET}}, MUT

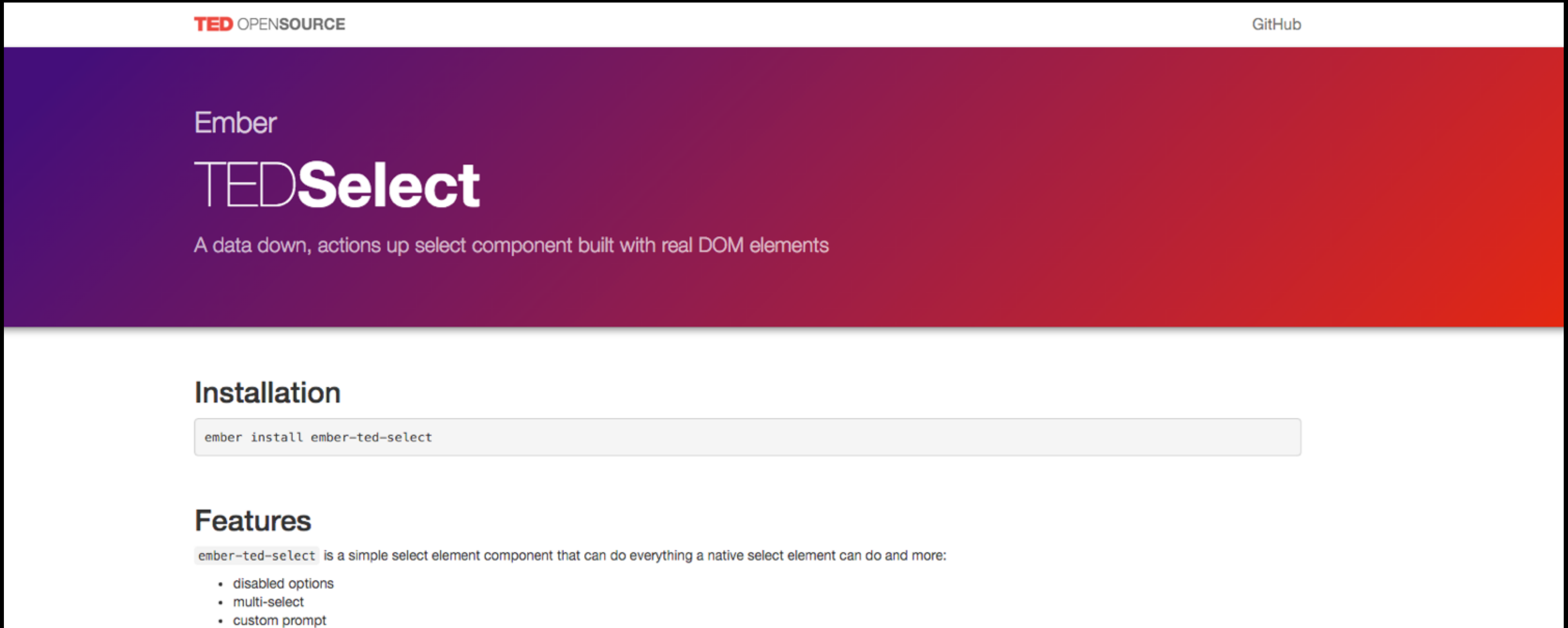
FLEXIBLE.

REUSABLE.

MODERN. 🙌

{{ted-select}}

http://tedconf.github.io/ember-ted-select



TED OPENSOURCE

GitHub

Ember

TEDSelect

A data down, actions up select component built with real DOM elements

Installation

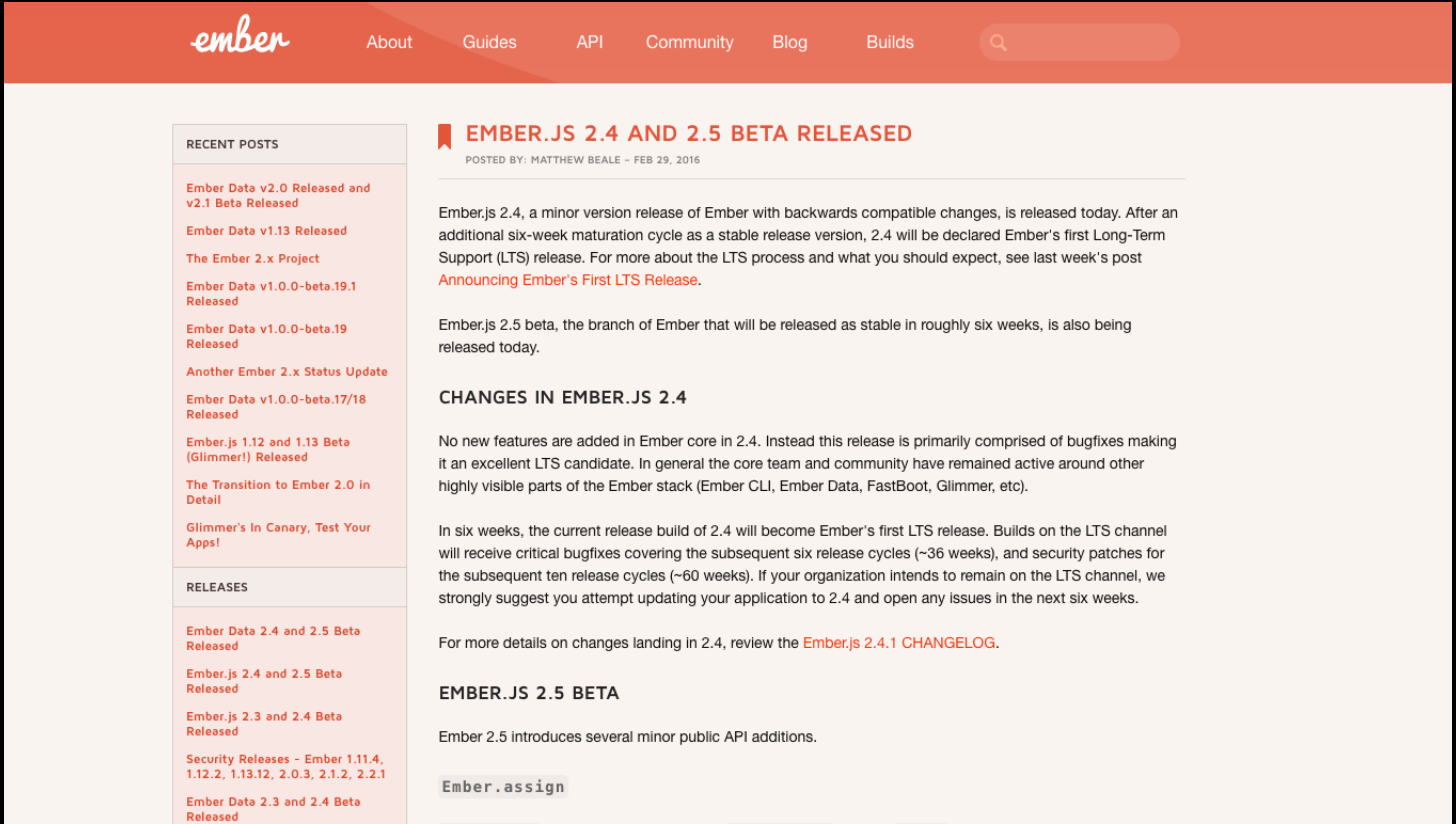
```
ember install ember-ted-select
```

Features

`ember-ted-select` is a simple select element component that can do everything a native select element can do and more:

- disabled options
- multi-select
- custom prompt

EMBERJS.COM/BLOG



RECENT POSTS

- Ember Data v2.0 Released and v2.1 Beta Released
- Ember Data v1.13 Released
- The Ember 2.x Project
- Ember Data v1.0.0-beta.19.1 Released
- Ember Data v1.0.0-beta.19 Released
- Another Ember 2.x Status Update
- Ember Data v1.0.0-beta.17/18 Released
- Ember.js 1.12 and 1.13 Beta (Glimmer!) Released
- The Transition to Ember 2.0 in Detail
- Glimmer's In Canary, Test Your Apps!

RELEASES

- Ember Data 2.4 and 2.5 Beta Released
- Ember.js 2.4 and 2.5 Beta Released
- Ember.js 2.3 and 2.4 Beta Released
- Security Releases - Ember 1.11.4, 1.12.2, 1.13.12, 2.0.3, 2.1.2, 2.2.1
- Ember Data 2.3 and 2.4 Beta Released

EMBER.JS 2.4 AND 2.5 BETA RELEASED

POSTED BY: MATTHEW BEALE – FEB 29, 2016

Ember.js 2.4, a minor version release of Ember with backwards compatible changes, is released today. After an additional six-week maturation cycle as a stable release version, 2.4 will be declared Ember's first Long-Term Support (LTS) release. For more about the LTS process and what you should expect, see last week's post Announcing Ember's First LTS Release.

Ember.js 2.5 beta, the branch of Ember that will be released as stable in roughly six weeks, is also being released today.

CHANGES IN EMBER.JS 2.4

No new features are added in Ember core in 2.4. Instead this release is primarily comprised of bugfixes making it an excellent LTS candidate. In general the core team and community have remained active around other highly visible parts of the Ember stack (Ember CLI, Ember Data, FastBoot, Glimmer, etc).

In six weeks, the current release build of 2.4 will become Ember's first LTS release. Builds on the LTS channel will receive critical bugfixes covering the subsequent six release cycles (~36 weeks), and security patches for the subsequent ten release cycles (~60 weeks). If your organization intends to remain on the LTS channel, we strongly suggest you attempt updating your application to 2.4 and open any issues in the next six weeks.

For more details on changes landing in 2.4, review the Ember.js 2.4.1 CHANGELOG.

EMBER.JS 2.5 BETA

Ember 2.5 introduces several minor public API additions.

`Ember.assign`

<SELECT> GOOD EMBER PATTERNS!

Thank you!

@brnnbrn

talks.brennaobrien.com/ember-select