Warming Up With

ember.js

Level 5 - Controlling our Growth

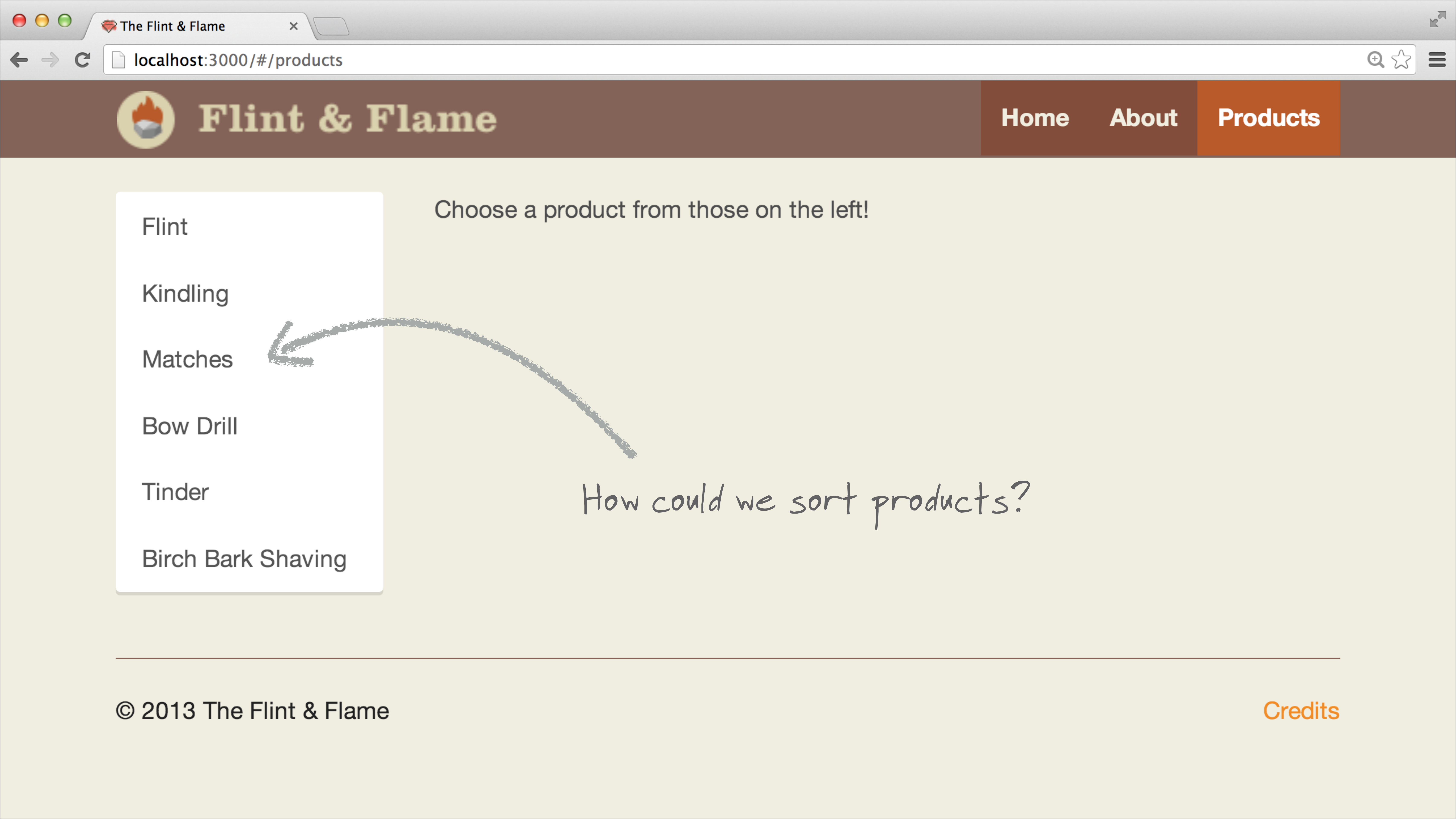
Array Controllers

Review

app.js

```
App.ProductsRoute = Ember.Route.extend({
  model: function() {
    return this.store.findAll('product');
  }
});
```

What about Sorting or Filtering?



Choose a product from those on the left!

- Flint
- Kindling
- Matches
- Bow Drill
- Tinder
- Birch Bark Shaving

How could we sort products?

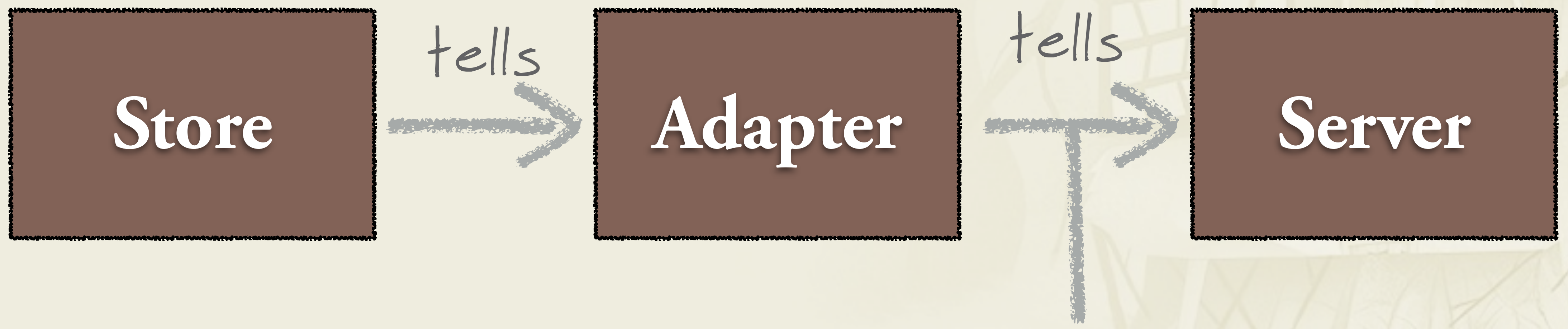
© 2013 The Flint & Flame

Sorting Models By Title

app.js

```
App.ProductsRoute = Ember.Route.extend({
  model: function() {
    return this.store.find('product', { order: 'title'});
 }
});
```

Tells our store how we want the product sorted



http://example.com/products?order=title

Sorting at the Server vs Client

To have the server do the sorting

```
this.store.find('product', { order: 'title'});
```

This will send our browser products already sorted.

To have the client (or browser) do the sorting

we will have to sort in the **controller**.

Controller

Decorate your applications data for the template.

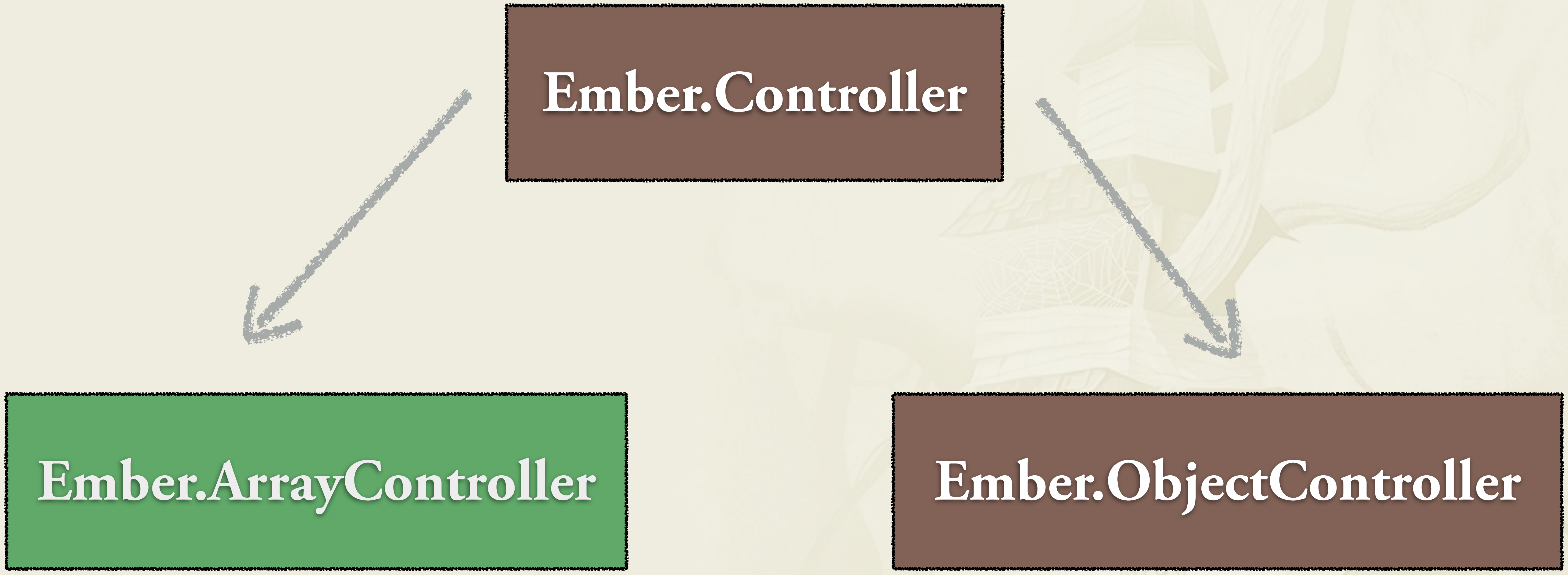
We Need a Special Controller

app.js

```
App.ProductsController = Ember.Controller.extend({});
```

We need a controller that knows how to deal with Arrays of objects.

EmberController Variants



Decorates an Array

Decorates a single Object

Array Controller

If our model is an array, we should use an ArrayController.

app.js

```
App.ProductsController = Ember.ArrayController.extend({});
```

This gives our Controller some special abilities.

Sorting by a Property

app.js

```
App.ProductsController = Ember.ArrayController.extend({
  sortProperties: ['title']
});
```

Sorts by the title of each product (in the browser)

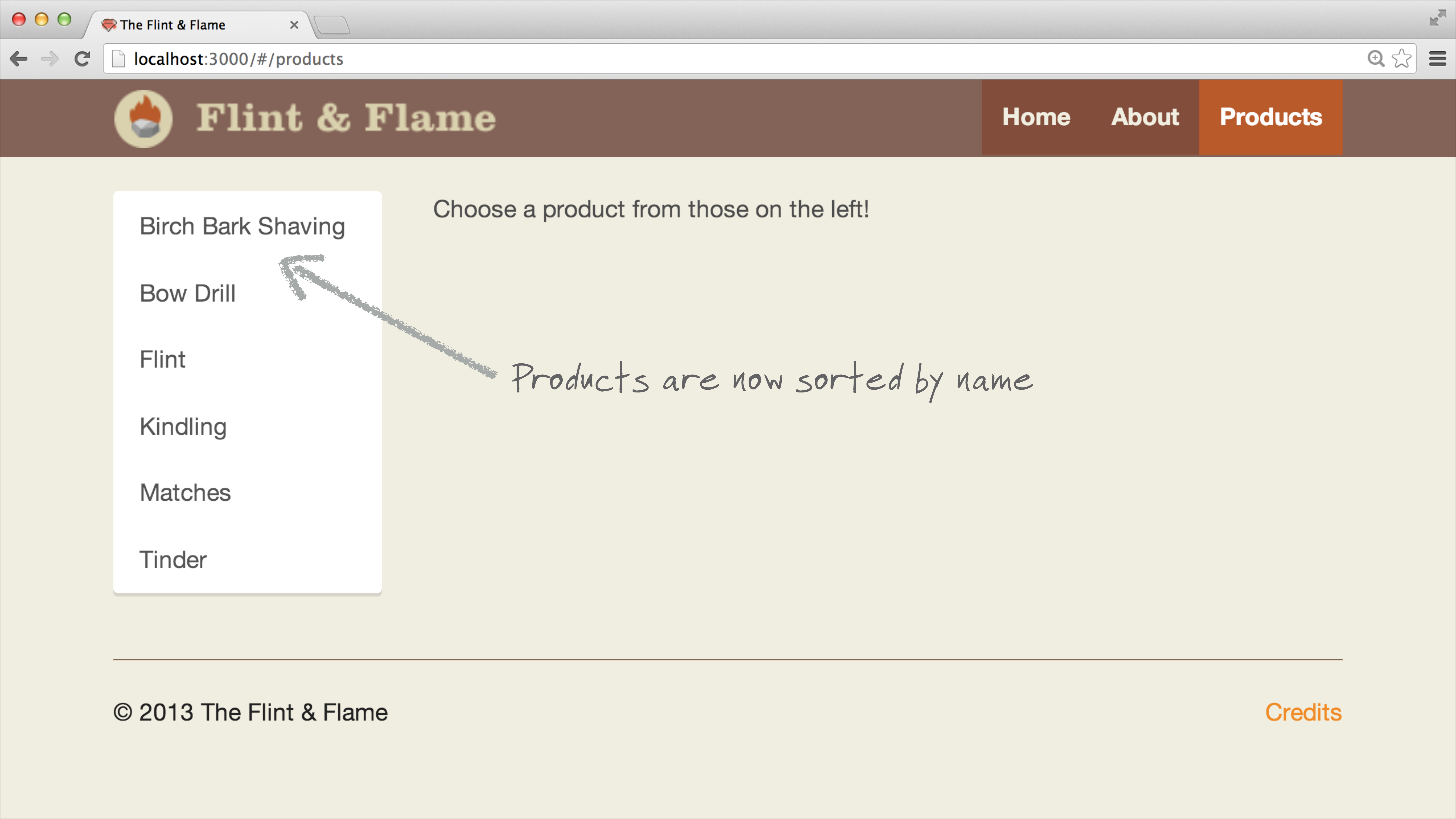
Sort Direction

By default this will sort ascending A-Z

```
App.ProductsController = Ember.ArrayController.extend({
  sortProperties: ['title']
});
```

To sort descending

```
App.ProductsController = Ember.ArrayController.extend({
  sortProperties: ['title'],
  sortAscending: false
});
```



The Flint & Flame

localhost:3000/#/products

Flint & Flame

Home
About
Products

Birch Bark Shaving
Bow Drill
Flint
Kindling
Matches
Tinder

Choose a product from those on the left!

Products are now sorted by name

© 2013 The Flint & Flame

Credits

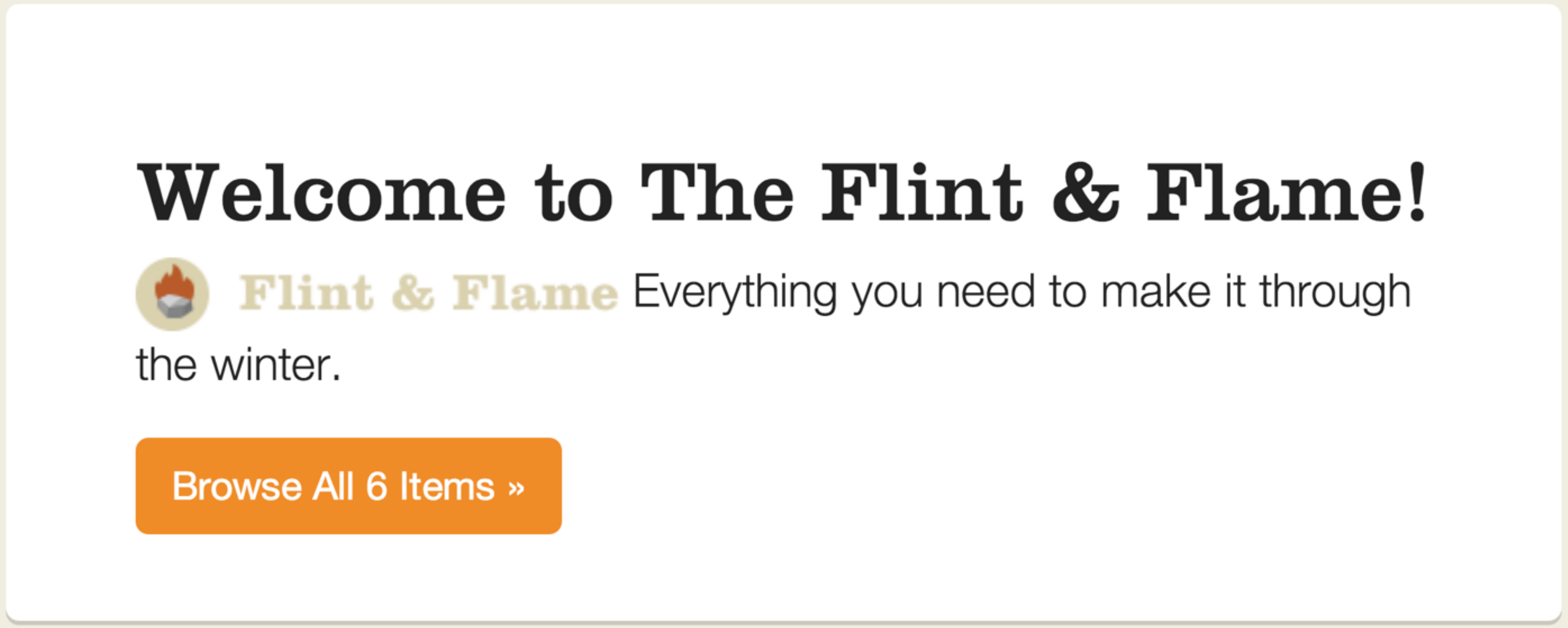
Refactoring IndexController

We need to convert this to actually count the length of our models.

Using Ember Data

app.js

```
App.IndexController = Ember.Controller.extend({
  productsCount: 6
});
```



Welcome to The Flint & Flame!

Flint & Flame Everything you need to make it through the winter.

Browse All 6 Items »

Step 1 - Associate a Model

Set the appropriate model that the IndexController will use.

app.js

```javascript
App.IndexController = Ember.Controller.extend({
  productsCount: 6
});

App.IndexRoute = Ember.Route.extend({
  model: function() {
    return this.store.findAll('product');
  }
});
```

Switching Controller Type

The model is an array

app.js

```
App.IndexController = Ember.ArrayController.extend({
  productsCount: 6
});
```

Computed Properties

app.js

```
App.IndexController = Ember.ArrayController.extend({
  productsCount: function() {
    return 6;
  }.property()
});
```

Convert productsCount to a function

Getting the Model Length

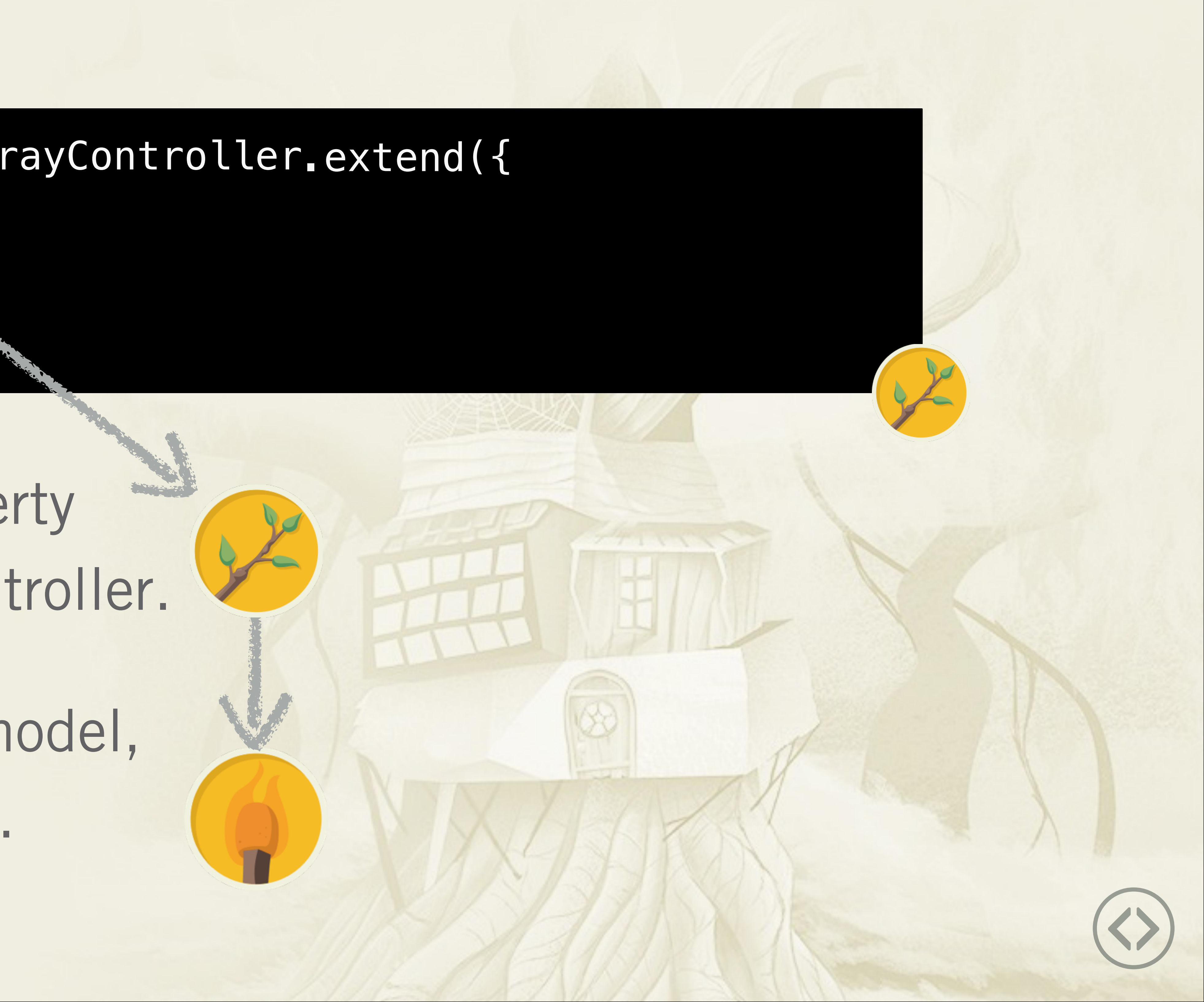
app.js

```
App.IndexController = Ember.ArrayController.extend({
  productsCount: function() {
    return this.get('length');
  }.property()
});
```

This will first look for a property called length in the ArrayController.

Then it will delegate to the model, and call length on the model.

```
model.length
```



Handlebars Property Binding

app.js

```
App.IndexController = Ember.ArrayController.extend({
  productsCount: function() {
    return this.get('length');
  }.property('length')
});
```

This will keep a watch on the 'length' property of the controller, and if it changes update productsCount

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  <p>There are {{productsCount}} products</p>
</script>
```

Computed Alias

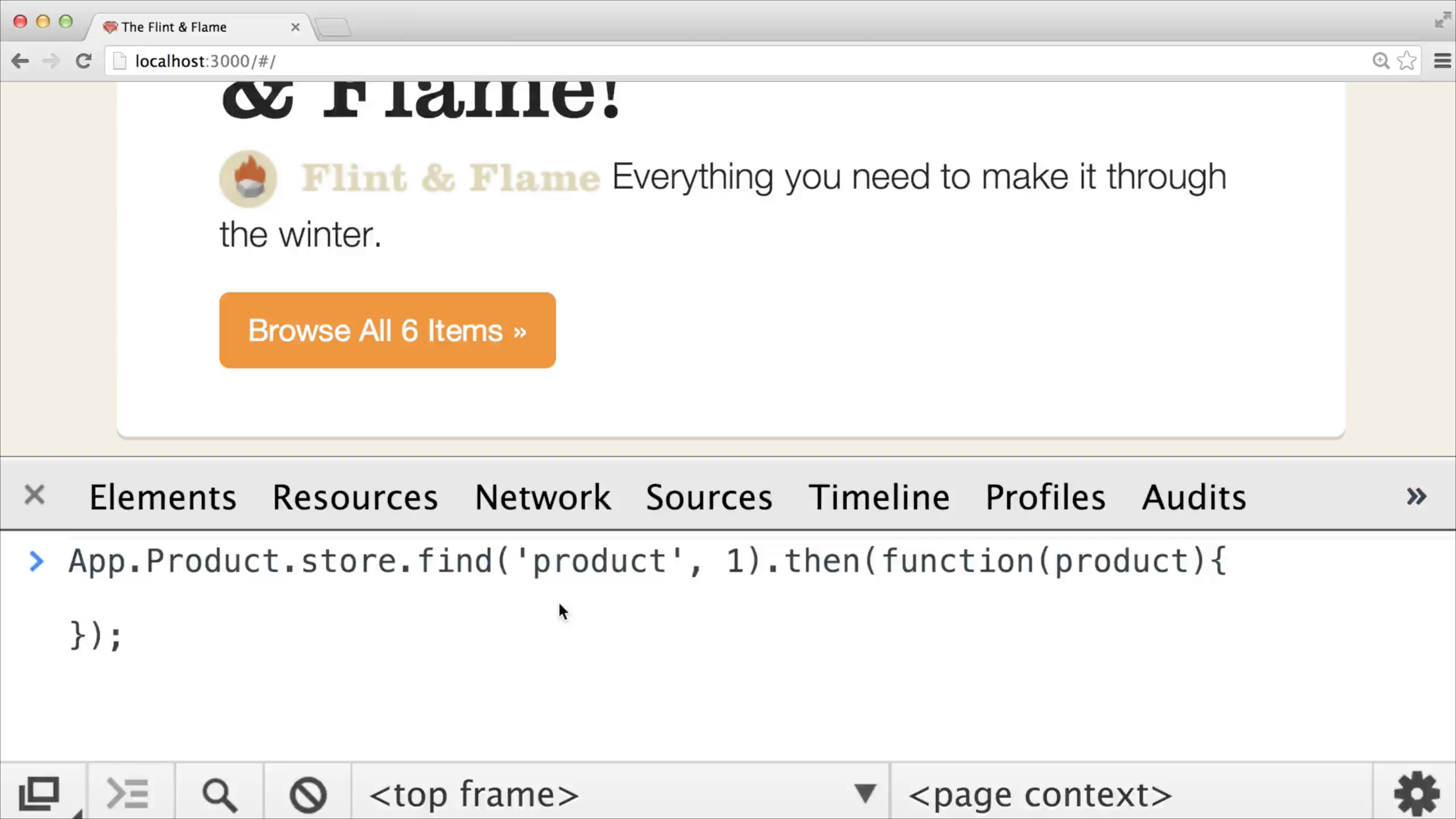
app.js

```
App.IndexController = Ember.ArrayController.extend({
  productsCount: function() {
    return this.get('length');
  }.property('length')
});
```

Functionally the same

app.js

```
App.IndexController = Ember.ArrayController.extend({
  productsCount: Ember.computed.alias('length')
});
```



```
> App.Product.store.find('product', 1).then(function(product){

  });
```