Review

app.js

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

App.ProductsOnSaleRoute = Ember.Route.extend({
  model: function () {
    return this.modelFor('products').filterBy('isOnSale');
  }
});
```

We're getting products in multiple places

Review

index.html

```
<script type='text/x-handlebars' data-template-name='products/onsale'>
  {{#each}} … {{/each}}
</script>

<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}} … {{/each}}
</script>
```

Showing the same details about
each product in multiple places

Ember Component

Similar implementation to Web Components

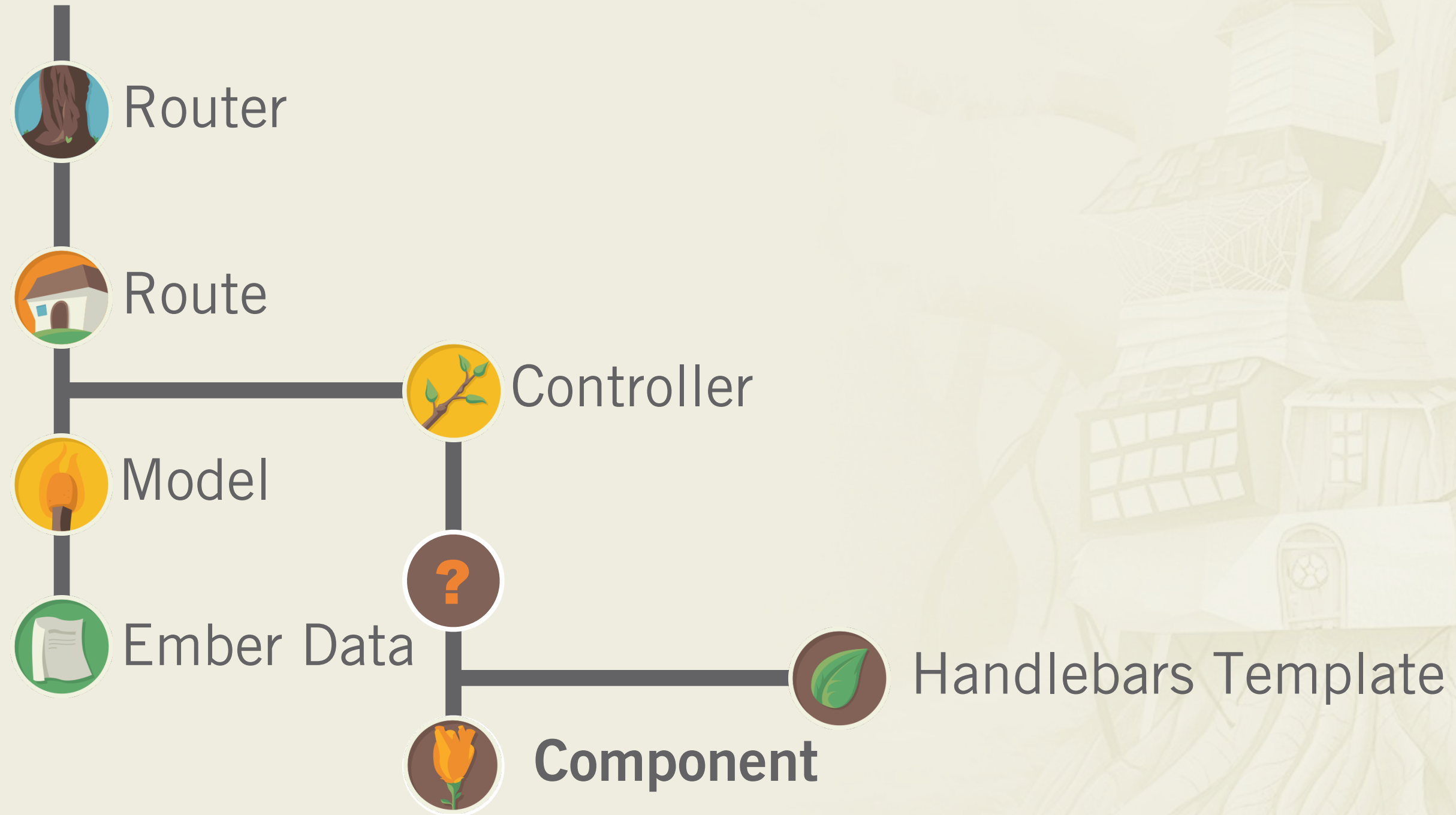Split out parts of your page to reusable components

Can wrap user interactivity

Each Ember Component will use this icon

Ember Component

Browser Request

- Router
- Route
- Model
- Ember Data
- Controller
- ?
- Component
- Handlebars Template

Splitting out to a Component

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}}
    <div class='col-sm-4'>
      <img {{bind-attr src='image'}} class='img-thumbnail col-md-5'/>
        <div class='col-sm-7'>
        <h2>{{title}}</h2>
        <p>{{description}}</p>
        <p>
          {{#link-to 'product' this classNames='btn btn-primary'}}
            Buy for ${{price}}
          {{/link-to}}
        </p>
      </div>
    </div>
  {{/each}}
</script>
```

This could be moved to a component

Splitting out to a Component

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}}
    <div class='col-sm-4'>
     …
      <h2>{{title}}</h2>
     …
    </div>
  {{/each}}
</script>

<script type='text/x-handlebars' data-template-name=''>
</script>
```

Creating a Component Template

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}}
    <!-- Show the component -->
  {{/each}}
</script>
```

What should we name this new template?

```
<script type='text/x-handlebars' data-template-name=''>
  <div class='col-sm-4'>
    …
    <h2>{{title}}</h2>
    …
  </div>
</script>
```

Component Template Naming

index.html

```html
<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}}
    <!-- Show the component -->
  {{/each}}
</script>

<script type='text/x-handlebars' data-template-name='components/product-details'>
  <div class='col-sm-4'>
    …
    <h2>{{title}}</h2>
    …
  </div>
</script>
```

Must start with components/

Must have a dash in the component name

Using a Component

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}}
    {{product-details}}
  {{/each}}
</script>

<script type='text/x-handlebars' data-template-name='components/product-details'>
  <div class='col-sm-4'>
   …
    <h2>{{title}}</h2>
   …
  </div>
</script>
```

Same as the component name

How do we let it know about the product?

Passing Arguments to a Component

index.html

```handlebars
<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}}
    {{product-details product=this}}
  {{/each}}
</script>

<script type='text/x-handlebars' data-template-name='components/product-details'>
  <div class='col-sm-4'>
   …
    <h2>{{product.title}}</h2>
   …
  </div>
</script>
```

Setting product in the component to the local product in the each loop

Reference product

Component Wrapper Element

```
<script type='text/x-handlebars' data-template-name='index'>
  {{#each onSale}}
    {{product-details product=this classNames='col-sm-4'}}
  {{/each}}
</script>


<script type='text/x-handlebars' data-template-name='components/product-details'>
  …
  <h2>{{product.title}}</h2>
  …
</script>
```
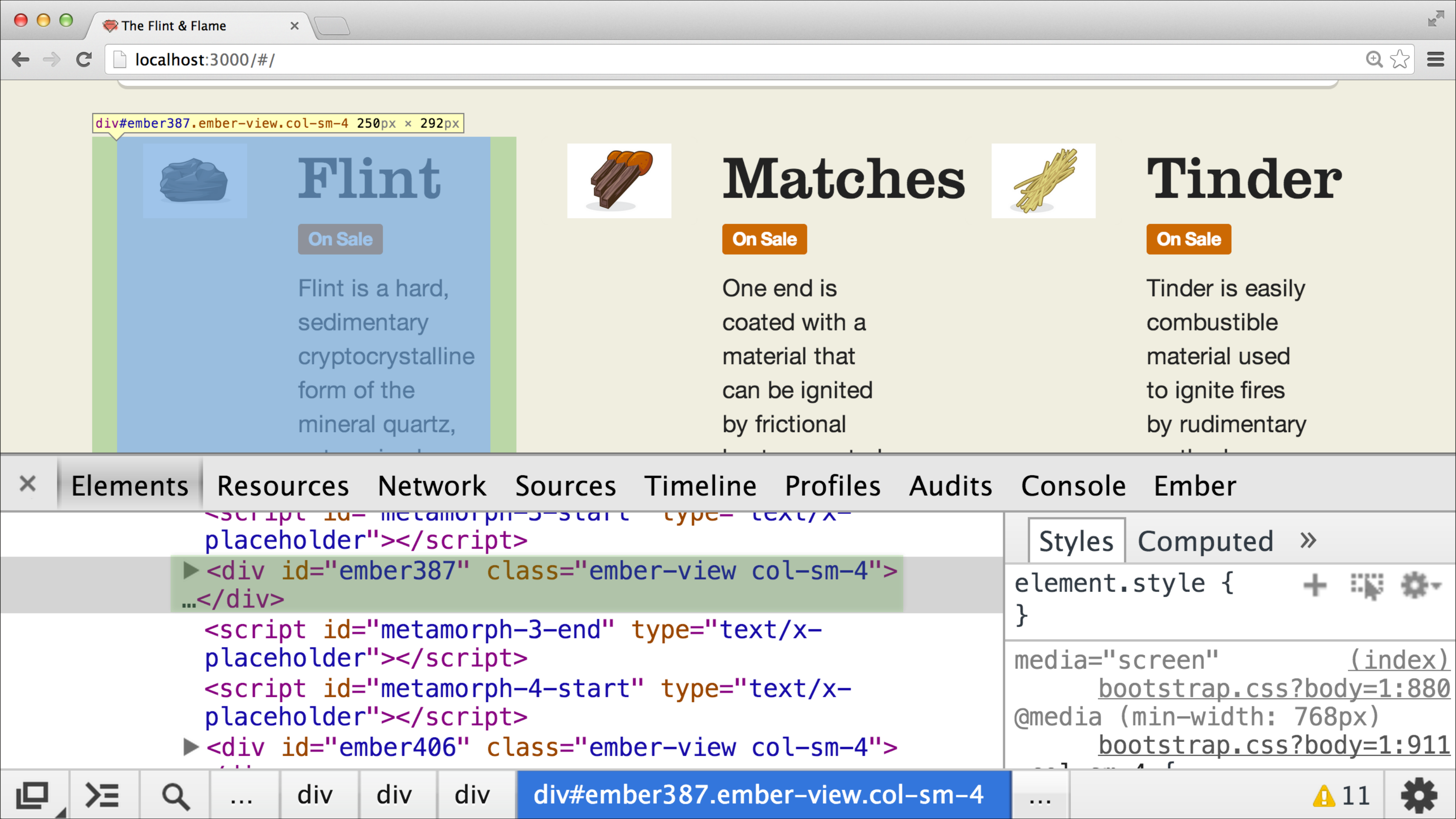
We can pass in a class name for our wrapper div

The component will render into a new

div element with our given class

localhost:3000/#/

div#ember387.ember-view.col-sm-4 250px × 292px

Flint

On Sale

Flint is a hard, sedimentary cryptocrystalline form of the mineral quartz,

Matches

On Sale

One end is coated with a material that can be ignited by frictional

Tinder

On Sale

Tinder is easily combustible material used to ignite fires by rudimentary

---

✕ | Elements | Resources | Network | Sources | Timeline | Profiles | Audits | Console | Ember

```
<script id="metamorph-3-start" type="text/x-
placeholder"></script>
▶ <div id="ember387" class="ember-view col-sm-4">
…</div>
<script id="metamorph-3-end" type="text/x-
placeholder"></script>
<script id="metamorph-4-start" type="text/x-
placeholder"></script>
▶ <div id="ember406" class="ember-view col-sm-4">
```

Styles | Computed »

```
element.style {
}
```

media="screen"   (index)
        bootstrap.css?body=1:880
@media (min-width: 768px)
        bootstrap.css?body=1:911

div#ember387.ember-view.col-sm-4

⚠ 11

Updating products/onsale

index.html

```
<script type='text/x-handlebars' data-template-name='products/onsale'>
  <ul>
    {{#each}}
      <li class='row'>
        <!-- product details -->
      </li>
    {{/each}}
  </ul>
</script>
```

We'll need to use the component here too

Passing in the Product

index.html

```html
<script type='text/x-handlebars' data-template-name='products/onsale'>
  <ul>
    {{#each}}
      <li class='row'>
        {{product-details product=this}}
      </li>
    {{/each}}
  </ul>
</script>
```

Changing the Root Element

index.html
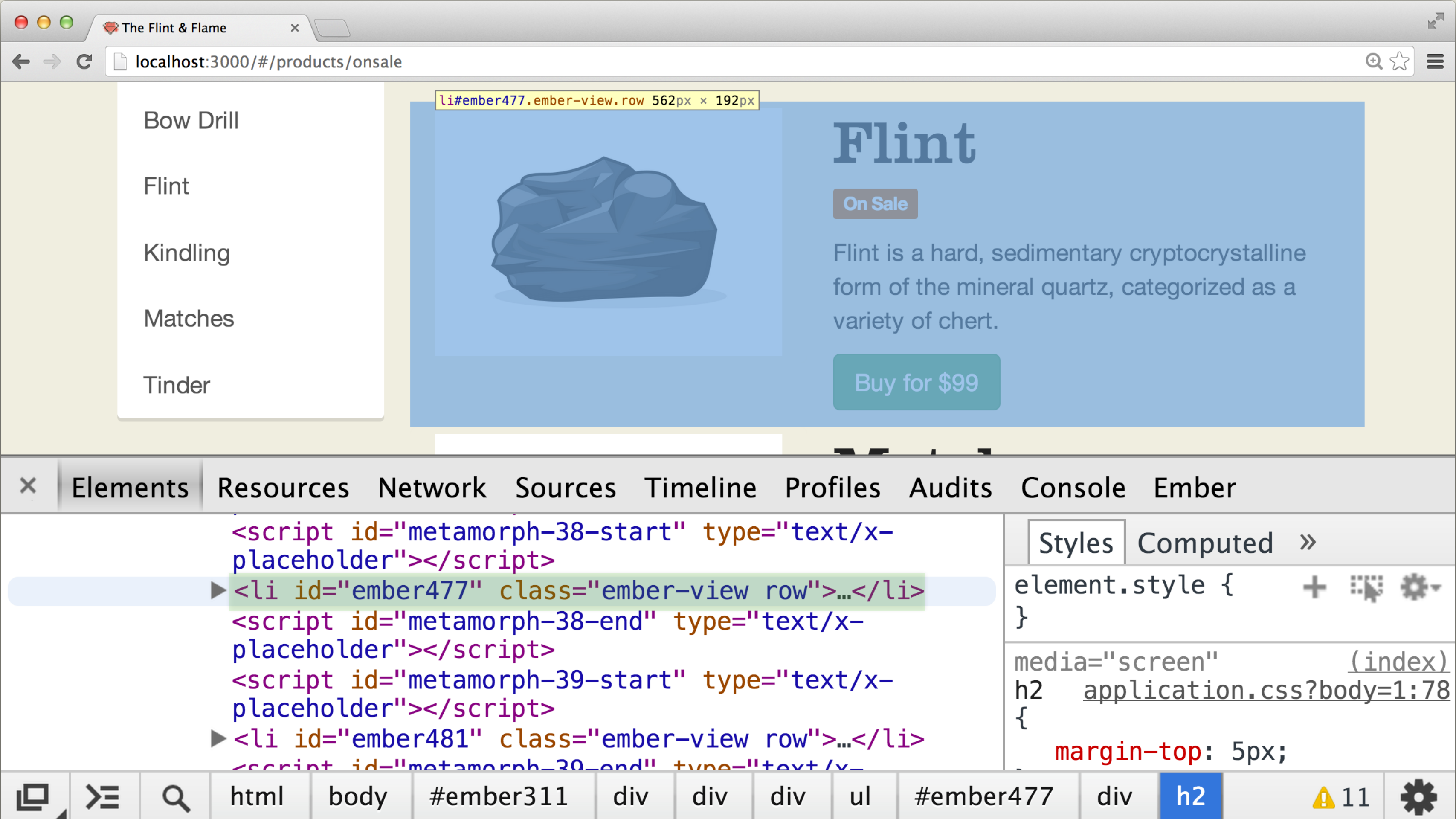
```
<script type='text/x-handlebars' data-template-name='products/onsale'>
  <ul>
    {{#each}}
      {{product-details product=this tagName='li' classNames='row'}}
    {{/each}}
  </ul>
</script>
```

Pass in tag, like we did with link-to

Add the class for the wrapper element

The Flint & Flame

localhost:3000/#/products/onsale

Bow Drill

Flint

Kindling

Matches

Tinder

li#ember477.ember-view.row 562px × 192px

Flint

On Sale

Flint is a hard, sedimentary cryptocrystalline form of the mineral quartz, categorized as a variety of chert.

Buy for $99

Elements   Resources   Network   Sources   Timeline   Profiles   Audits   Console   Ember

```
<script id="metamorph-38-start" type="text/x-placeholder"></script>
<li id="ember477" class="ember-view row">…</li>
<script id="metamorph-38-end" type="text/x-placeholder"></script>
<script id="metamorph-39-start" type="text/x-placeholder"></script>
<li id="ember481" class="ember-view row">…</li>
<script id="metamorph-39-end" type="text/x-
```

Styles   Computed   »

element.style {
}

media="screen"   (index)
h2   application.css?body=1:78
{
    margin-top: 5px;

html   body   #ember311   div   div   div   ul   #ember477   div   h2   ⚠ 11