Warming Up With ember.js

Level 2 - Rendering the Flame

Controller Basics

Review

Mapped paths to route names.

app.js

```
App.Router.map(function() {
  this.route('about');
});
```

Link to other pages using route names, not paths.

index.html

```
{{#link-to 'about' tagName='li'}}About{{/link-to}}
```

Template Variables

You can use properties in your template

index.html

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

How do you set a property for use in a template?

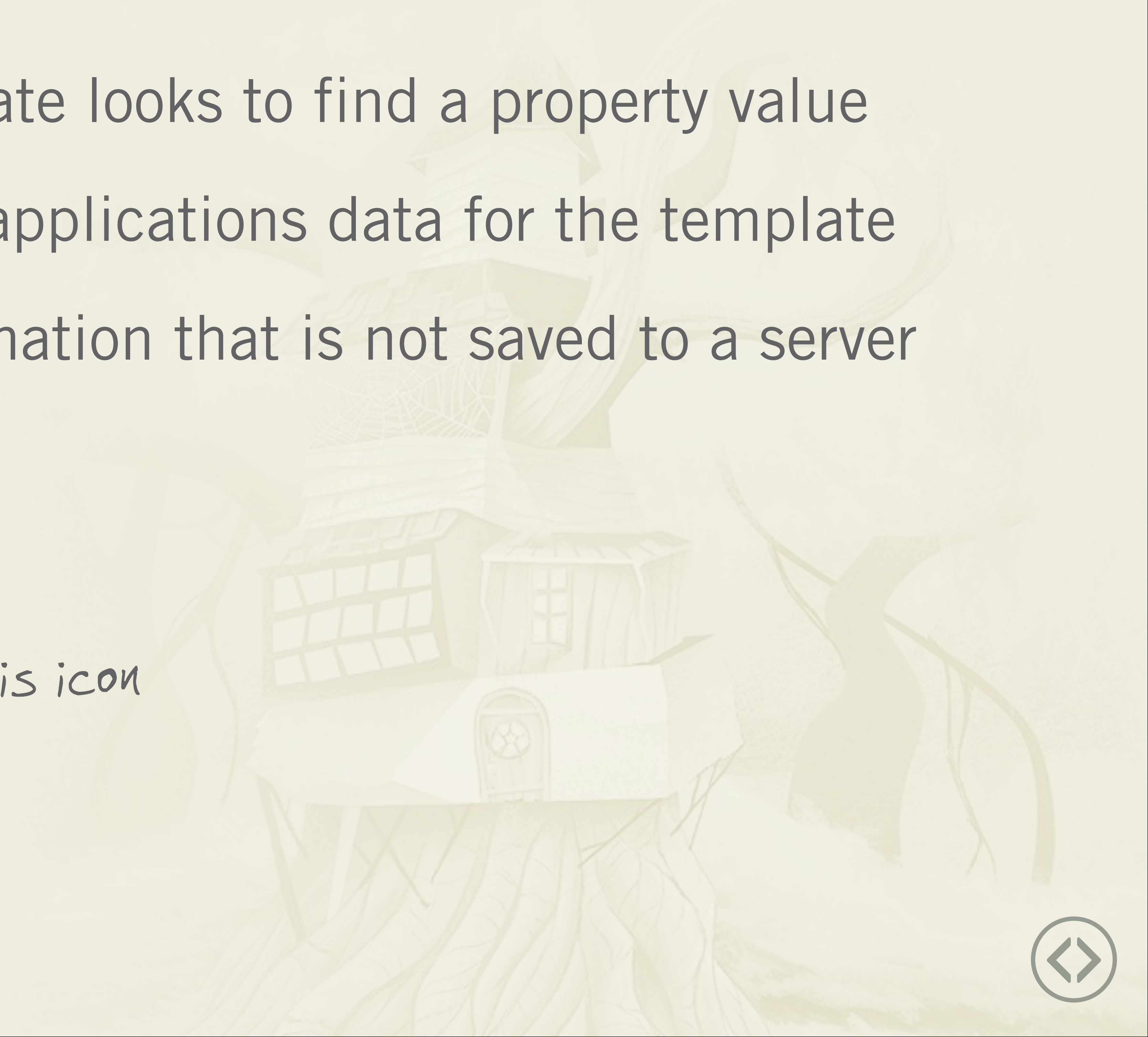
Ember Controller

Where a template looks to find a property value

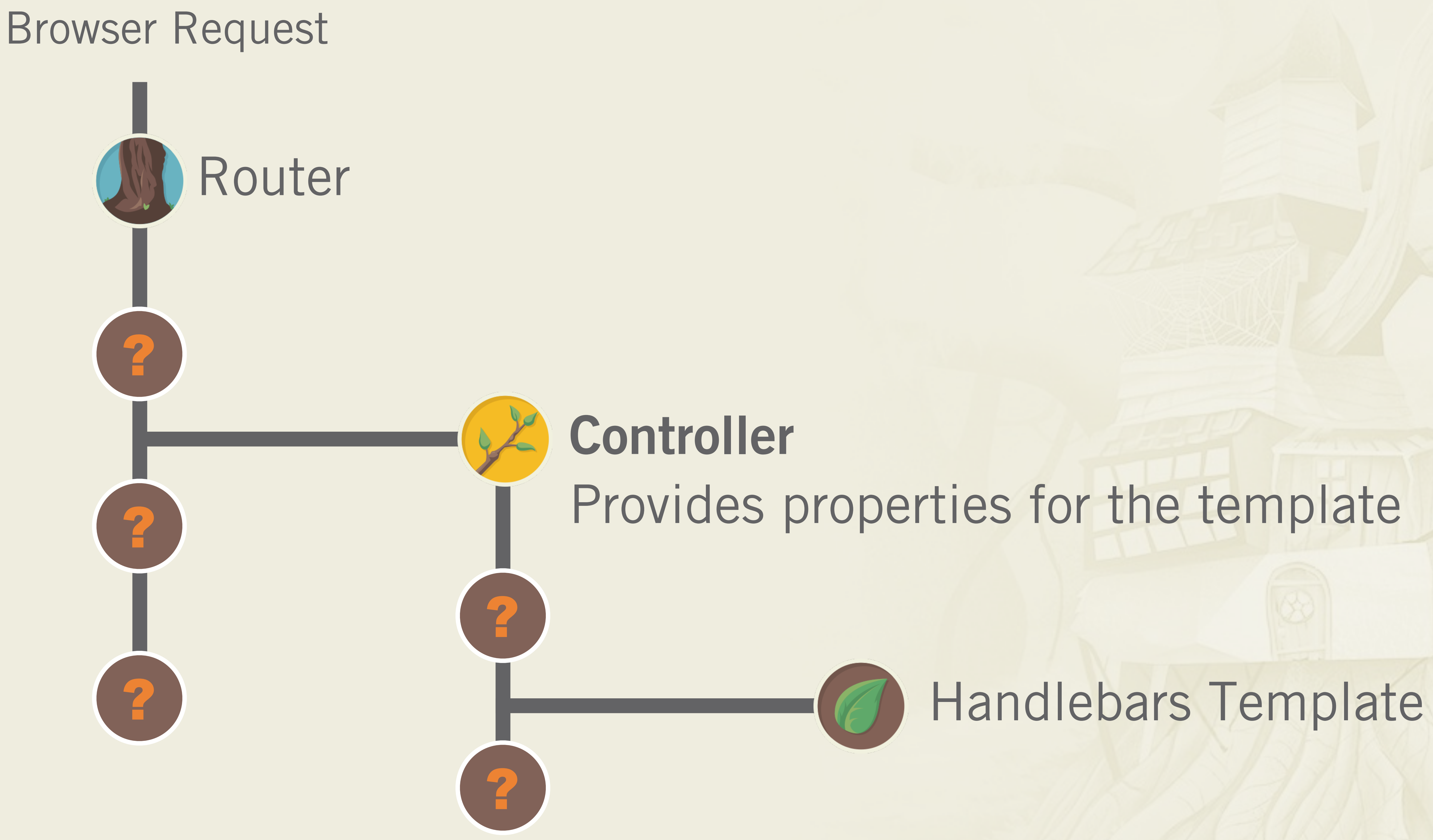
Decorate your applications data for the template

Contains information that is not saved to a server

Each Ember Controller will use this icon



Routing Our Way In



Asking the Controller

index.html

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

looks for a controller named "IndexController", and the property inside it.

app.js

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

```
There are 6 products
```

Every Route has a Default Controller

```
App.Router.map(function() {
  this.route('about');
});
```

Behind the scenes, Ember creates this About controller.

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

If we want to declare properties that get rendered inside a template, then we can declare this Controller.

Binding with Metamorph!

index.html

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

```
<p>There are
  <script id="metamorph-2-start" type="text/x-placeholder"></script>
  6
  <script id="metamorph-2-end" type="text/x-placeholder"></script>
  products
</p>
```

Ember needs to wrap properties so it can potentially update it later. (AKA. **Binding**)

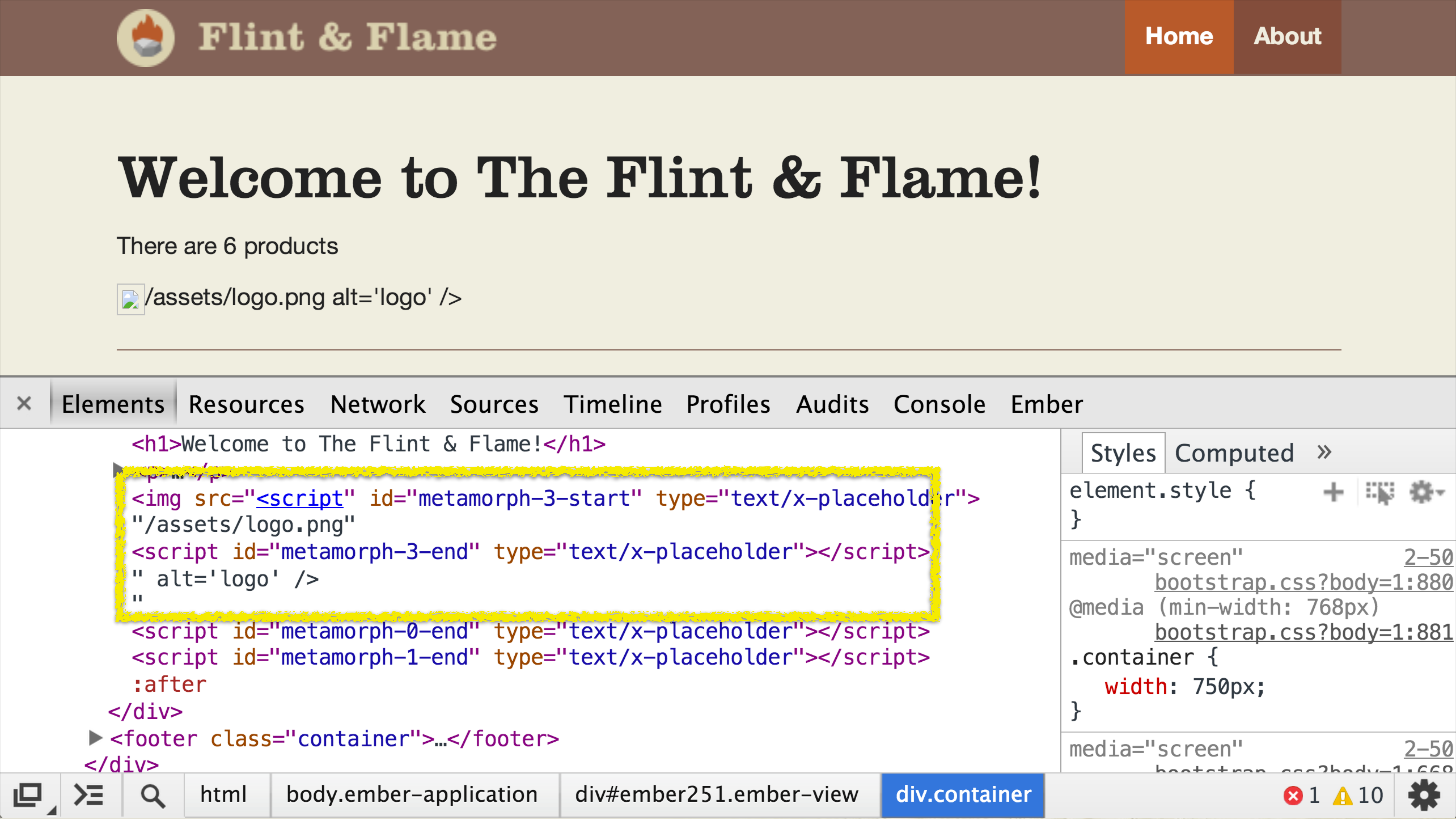
Binding Attributes?

app.js

```
App.IndexController = Ember.Controller.extend({
  productsCount: 6,
  logo: '/images/logo.png'
});
```

index.html

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



Welcome to The Flint & Flame!

There are 6 products

/assets/logo.png alt='logo' />

```
<h1>Welcome to The Flint & Flame!</h1>
<img src="<script" id="metamorph-3-start" type="text/x-placeholder">
"/assets/logo.png"
<script id="metamorph-3-end" type="text/x-placeholder"></script>
" alt='logo' />
"
<script id="metamorph-0-end" type="text/x-placeholder"></script>
<script id="metamorph-1-end" type="text/x-placeholder"></script>
:after
</div>
<footer class="container">…</footer>
</div>
```

```
element.style {
}
media="screen"                    2-50
        bootstrap.css?body=1:880
@media (min-width: 768px)
        bootstrap.css?body=1:881
.container {
   width: 750px;
}
media="screen"                    2-50
```

Property Binding

index.html

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

```
<img src="
  <script id="metamorph-2-start" type="text/x-placeholder"></script>
     /images/logo.png
  <script id="metamorph-2-end" type="text/x-placeholder"></script>
>
```

We need to bind attributes differently.

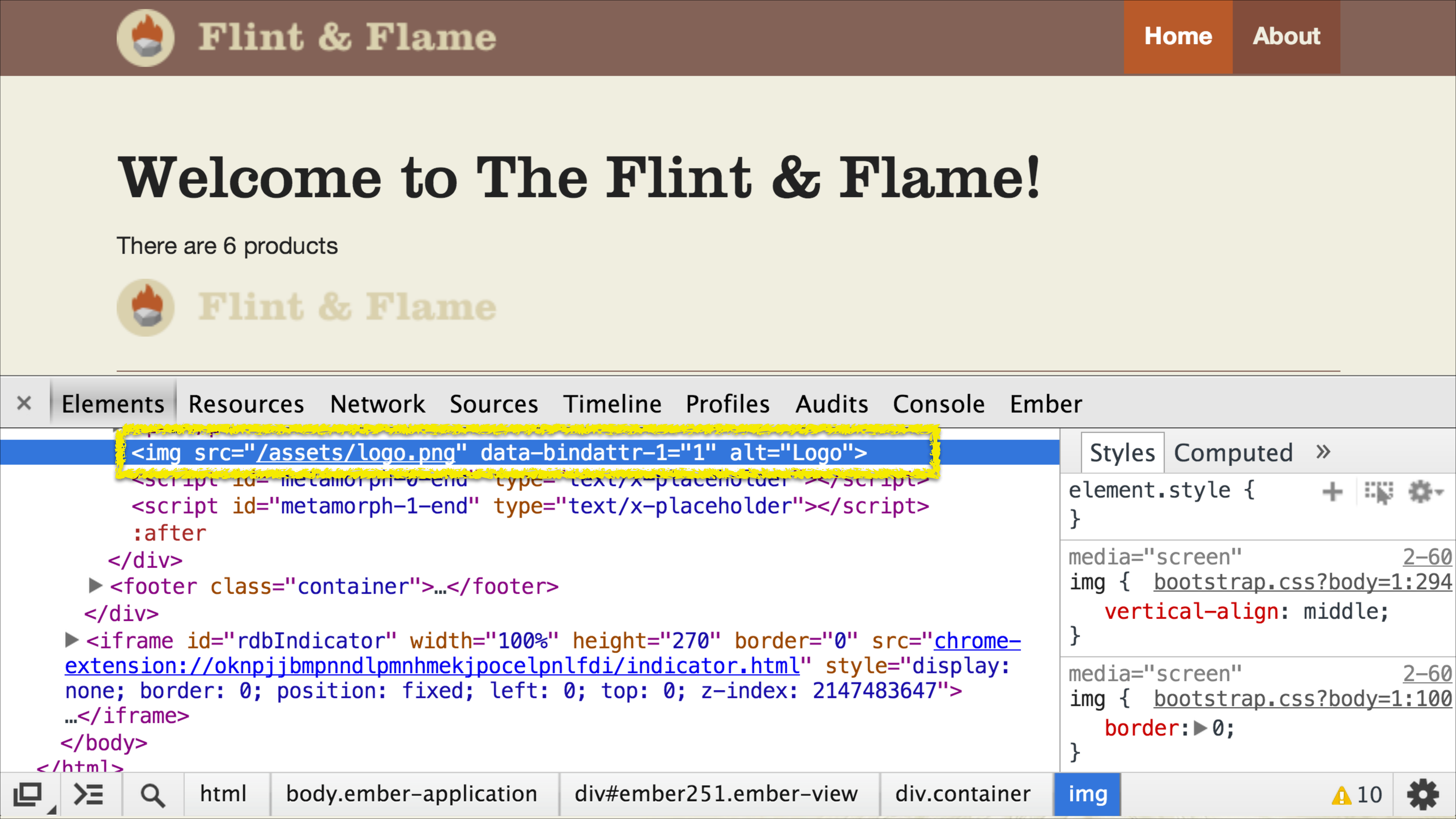
How to Bind Attributes

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  <p>There are {{productsCount}} products</p>
  <img {{bind-attr src='logo'}} alt='Logo' />
</script>
```

```
<img src="/images/logo.png" data-bindattr-1="1">
```

Ember meta data is still added to enable binding



Welcome to The Flint & Flame!

There are 6 products

```
<img src="/assets/logo.png" data-bindattr-1="1" alt="Logo">
<script id="metamorph-0-end" type="text/x-placeholder"></script>
<script id="metamorph-1-end" type="text/x-placeholder"></script>
:after
</div>
▶ <footer class="container">…</footer>
</div>
▶ <iframe id="rdbIndicator" width="100%" height="270" border="0" src="chrome-
extension://oknpjjbmpnndlpmnhmekjpocelpnlfdi/indicator.html" style="display:
none; border: 0; position: fixed; left: 0; top: 0; z-index: 2147483647">
…</iframe>
</body>
```

```
element.style {
}
media="screen"                    2–60
img {  bootstrap.css?body=1:294
   vertical-align: middle;
}
media="screen"                    2–60
img {  bootstrap.css?body=1:100
   border:▶0;
}
```

Dynamic Content

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  <p>Rendered on {{time}}</p>
</script>
```

How do we fetch the current time?

We need to create a property that calls this function:

```
(new Date()).toDateString()
```

This is more then just a value, we need to execute some code to get the value.

Controller Functions

index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  <p>Rendered on {{time}}</p>
</script>
```

app.js

```
App.IndexController = Ember.Controller.extend({
  productsCount: 6,
  logo: '/images/logo.png',
  time: function() {
    return (new Date()).toDateString()
  }
});
```

We need to tell Ember this is a property and to call it

Controller Functions as Properties

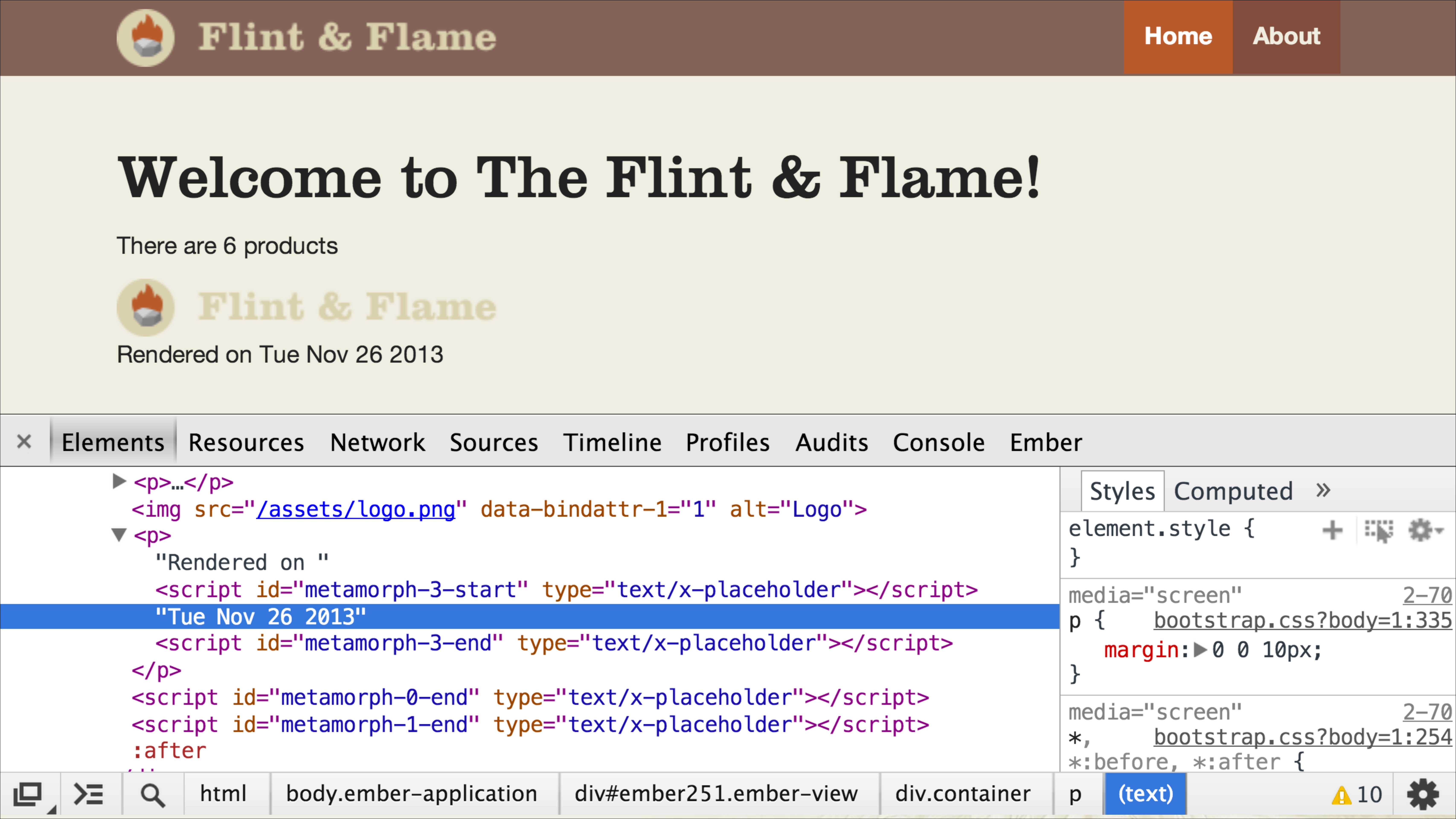
index.html

```
<script type='text/x-handlebars' data-template-name='index'>
  <p>Rendered on {{time}}</p>
</script>
```

app.js

```
App.IndexController = Ember.Controller.extend({
  productsCount: 6,
  logo: '/images/logo.png',
  time: function() {
    return (new Date()).toDateString()
  }.property()
});
```

time will be called, and that value used as a property



Welcome to The Flint & Flame!

There are 6 products

Rendered on Tue Nov 26 2013

```
▶ <p>…</p>
  <img src="/assets/logo.png" data-bindattr-1="1" alt="Logo">
▼ <p>
    "Rendered on "
    <script id="metamorph-3-start" type="text/x-placeholder"></script>
    "Tue Nov 26 2013"
    <script id="metamorph-3-end" type="text/x-placeholder"></script>
  </p>
  <script id="metamorph-0-end" type="text/x-placeholder"></script>
  <script id="metamorph-1-end" type="text/x-placeholder"></script>
  :after
```

```
element.style {
}
media="screen"                    2-70
p {        bootstrap.css?body=1:335
  margin:▶0 0 10px;
}
media="screen"                    2-70
*,         bootstrap.css?body=1:254
*:before, *:after {
```