

Warming Up With **ember.js**

Level 7 - Acting on Instinct

Data binding with Models



Review

index.html

```
<script type='text/x-handlebars' data-template-name='product'>
  <div class='new-review'>
    <h3>Review {{title}}</h3>
    {{#if text}}
      <p class='text-muted'>{{text}}</p>
    {{/if}}
    {{textarea valueBinding='text'}}
    <button {{action 'createReview'}} class='btn-primary'>Review</button>
  </div>
</script>
```



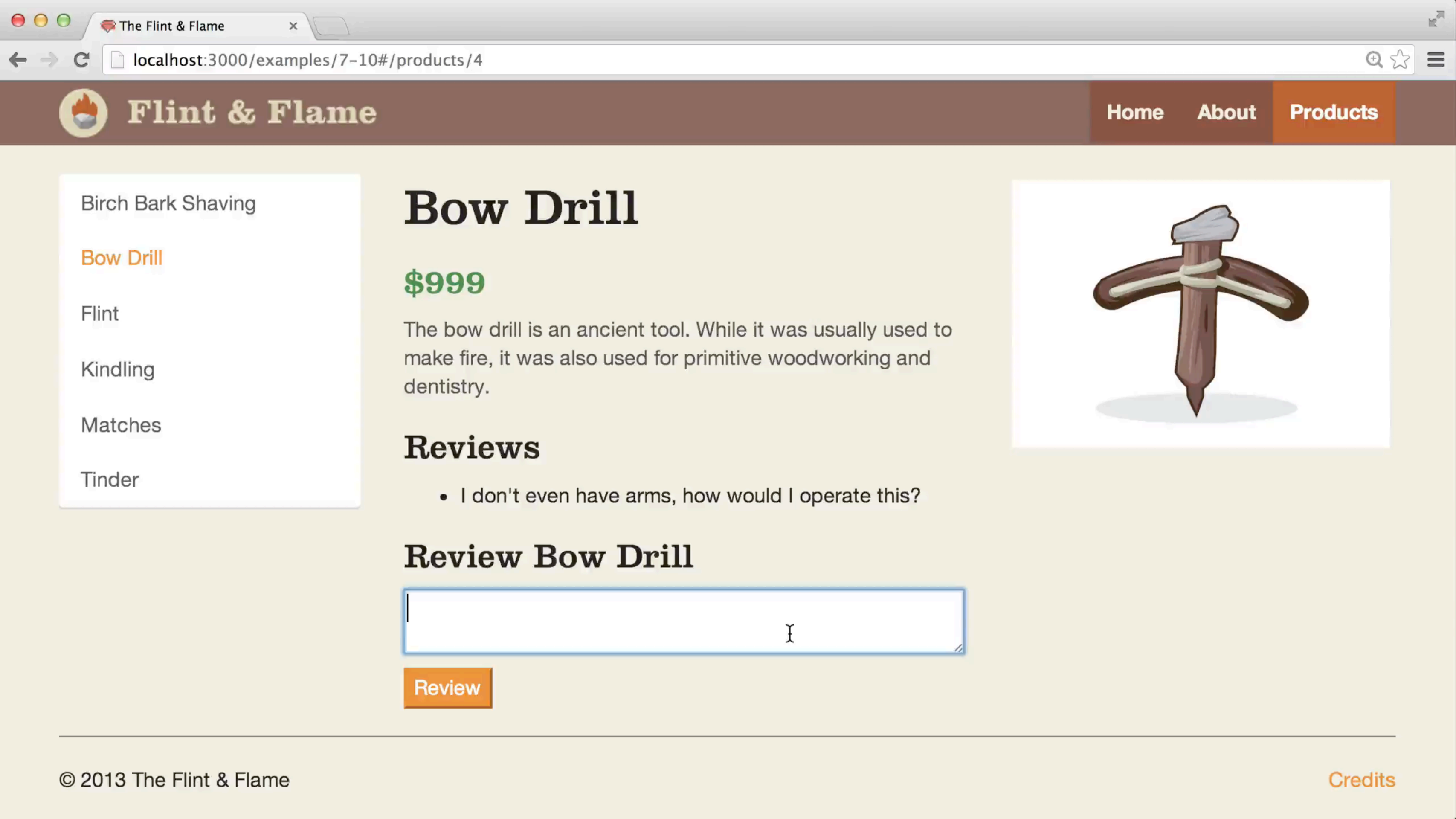
Review

app.js

```
App.ProductController = Ember.ObjectController.extend({
  text: '',
  actions: {
    createReview: function() {
      var review = this.store.createRecord('review', {
        text: this.get('text'),
        product: this.get('model'),
        reviewedAt: new Date()
      });
      var controller = this;

      review.save().then(function(review) {
        controller.set('text', '');
        controller.get('model.reviews').addObject(review);
      });
    }
  }
});
```





Birch Bark Shaving

Bow Drill

Flint

Kindling

Matches

Tinder

Bow Drill

\$999

The bow drill is an ancient tool. While it was usually used to make fire, it was also used for primitive woodworking and dentistry.



Reviews

- I don't even have arms, how would I operate this?

Review Bow Drill

Review

Controller Reuse

This same controller object is being reused!

app.js

```
App.ProductController = Ember.ObjectController.extend({  
  text: ''  
});
```


The only part that's changing is the model property, representing the product.



Using A New Model

index.html

```
<script type='text/x-handlebars' data-template-name='product'>
  <div class='new-review'>
    <h3>Review {{title}}</h3>
    {{#if text}}
      <p class='text-muted'>{{text}}</p>
    {{/if}}
    {{textarea valueBinding='text'}}
    <button {{action 'createReview'}} class='btn-primary'>Review</button>
  </div>
</script>
```



What if we bind this functionality to a
“new review” object?



Two Things to Fix

Create a new review object for this product

app.js

```
App.ProductController = Ember.ObjectController.extend({
  text: '',
  actions: {
    createReview: function() {
      ...
    }
  }
});
```

1. Create a 'review' property that returns a new Review model, and we'll set the text on the review model.
2. Create a new (blank) 'review' every time the product model changes.



A New Review

app.js

Create a new review object for this product

```
App.ProductController = Ember.ObjectController.extend({  
  review: function() {  
    return this.store.createRecord('review', {  
      product: this.get('model')  
    });  
  }.property('model'),  
  actions: {  
    createReview: function() {  
      ...  
    }  
  }  
});
```

Watch the model, and when it changes immediately invoke this function.



Now let's use the Review model

index.html

```
<script type='text/x-handlebars' data-template-name='product'>
  <div class='new-review'>
    <h3>Review {{title}}</h3>
    {{#if text}}
      <p class='text-muted'>{{text}}</p>
    {{/if}}
    {{textarea valueBinding='text'}}
    <button {{action 'createReview'}} class='btn-primary'>Review</button>
  </div>
</script>
```



Template Updates for Review

index.html

```
<script type='text/x-handlebars' data-template-name='product'>
  <div class='new-review'>
    <h3>Review {{title}}</h3>
    {{#if review.text}}
      <p class='text-muted'>{{review.text}}</p>
    {{/if}}
    {{textarea valueBinding='review.text'}}
    <button {{action 'createReview'}} class='btn-primary'>Review</button>
  </div>
</script>
```

Bind directly to the text
property on our review object



Let's Update the createReview

app.js

```
App.ProductController = Ember.ObjectController.extend({
  review: function() { ... }.property('model'),
  actions: {
    createReview: function() {
      var review = this.store.createRecord('review', {
        text: this.get('text'),
        product: this.get('model'),
        reviewedAt: new Date()
      });

      var controller = this;

      review.save().then(function(review) {
        controller.set('text', '');
        controller.get('model.reviews').addObject(review);
      });
    }
  }
});
```

We no longer need to
create a model.



Updating createReview

app.js

```
App.ProductController = Ember.ObjectController.extend({
  review: function() { ... }.property('model'),
  actions: {
    createReview: function() {
      var controller = this;
      this.get('review').set('reviewedAt', new Date());
      this.get('review').save().then(function(review) {
        controller.get('model.reviews').addObject(review);
      });
    }
  }
});
```



Birch Bark Shaving

Bow Drill

Flint

Kindling

Matches

Tinder

Flint

\$99

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

Reviews

- Not the brightest flame, but warm!
- Started a fire in no time!

Review Flint

Review

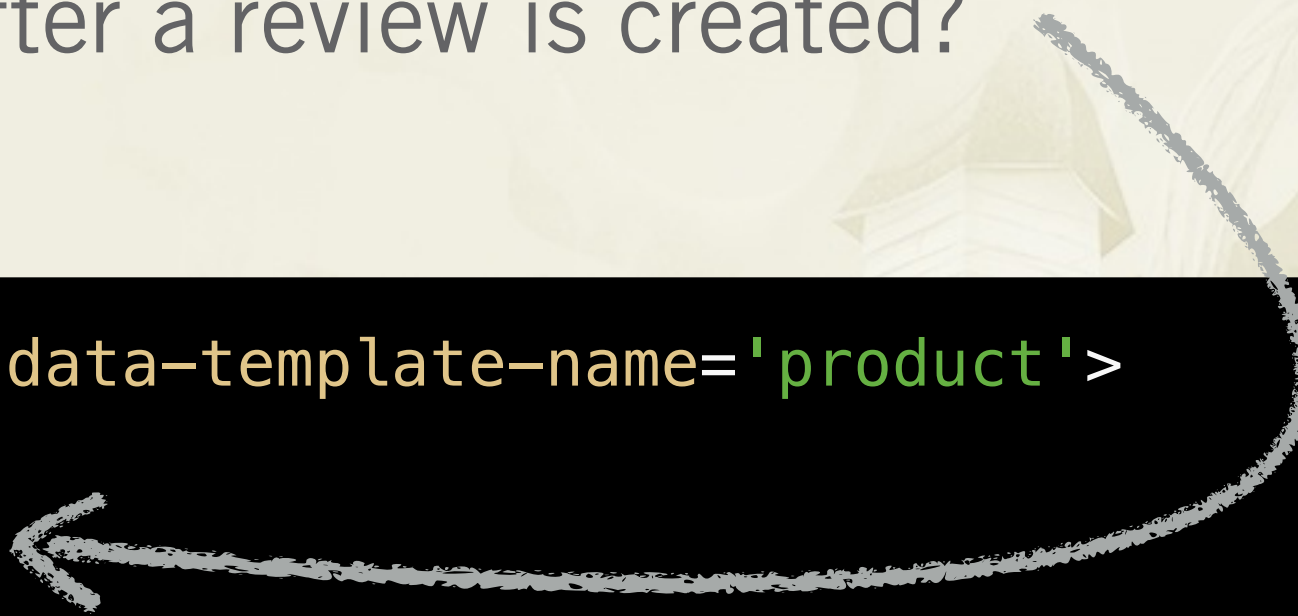


Hiding after Review Creation

How do we hide this form after a review is created?

index.html

```
<script type='text/x-handlebars' data-template-name='product'>
  ...
  <div class='new-review'>
    <h3>Review {{title}}</h3>
    {{#if review.text}}
      <p class='text-muted'>{{review.text}}</p>
    {{/if}}
    {{textarea valueBinding='review.text'}}
    <button {{action 'createReview'}} class='btn-primary'>Review</button>
  </div>
</script>
```



Hiding the Form

We'll need to create this property on our ProductController

index.html

```
<script type='text/x-handlebars' data-template-name='product'>
  ..
  {{#if isNotReviewed}}
    <div class='new-review'>
      <h3>Review {{title}}</h3>
      {{#if review.text}}
        <p class='text-muted'>{{review.text}}</p>
      {{/if}}
      {{textarea valueBinding='review.text'}}
      <button {{action 'createReview'}} class='btn-primary'>Review</button>
    </div>
  {{/if}}
</script>
```



isNotReviewed Property

app.js

```
App.ProductController = Ember.ObjectController.extend({  
  review: function() { ... }.property('model'),  
  actions: { ... },  
  isNotReviewed: Ember.computed.alias('review.isNew')  
});
```

See the Ember API Docs for all states at <http://emberjs.com/>



Ember Data Model States



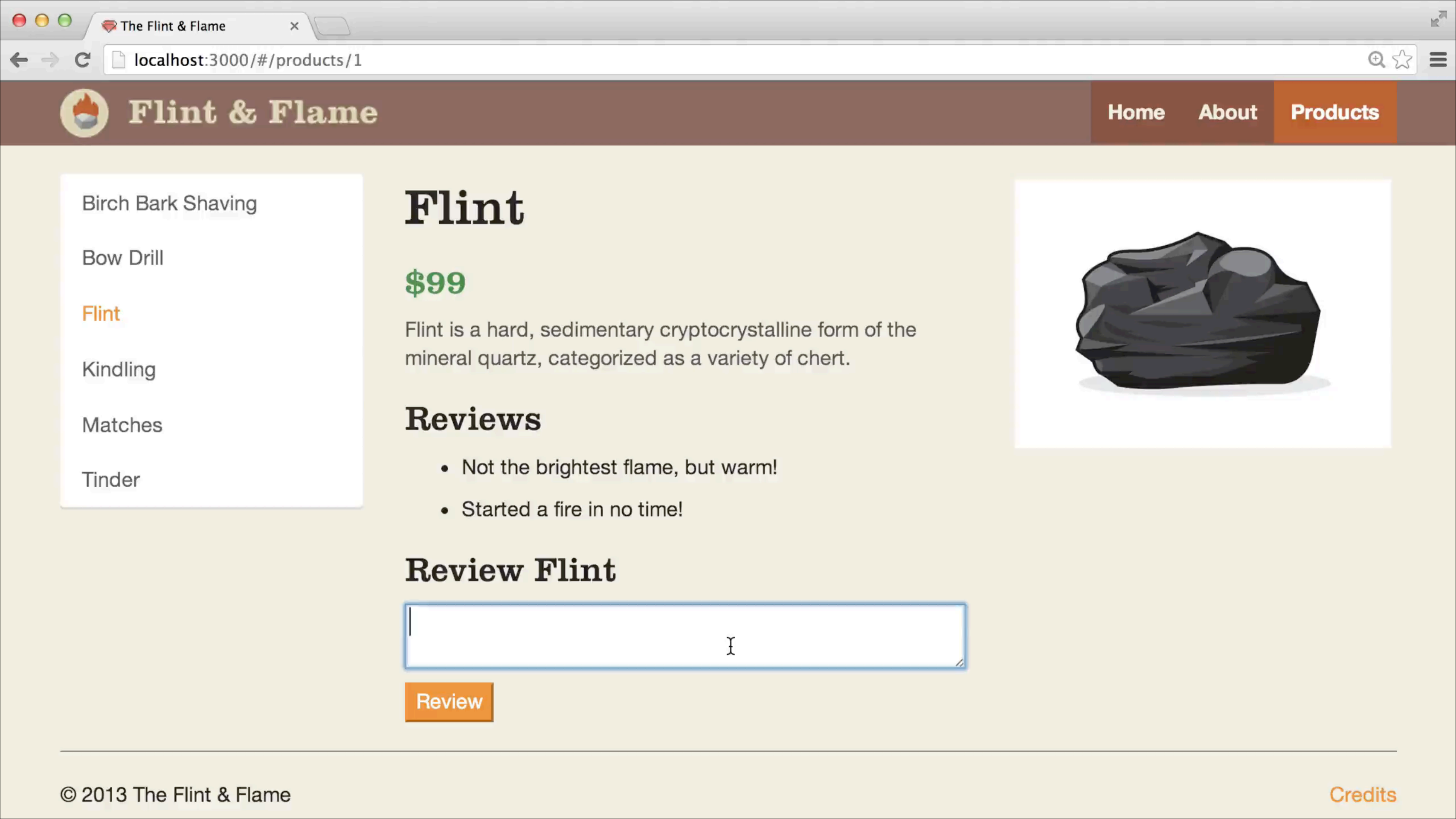
isNew Has not been saved

isLoading While fetching from a server

isDirty Has unsaved changes

isLoading After fetching from a server





Birch Bark Shaving

Bow Drill

Flint

Kindling

Matches

Tinder

Flint

\$99

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

Reviews

- Not the brightest flame, but warm!
- Started a fire in no time!

Review Flint

Review