



TRP.

407

try ember



Level 5.2

# Properties and Components

.....  
Components









# Populating the Cost Percentages

The percentages need to be implemented and styled based on their values.

localhost:4200/orders/2345

Order details for Olivia Okapi (#2345)

| Item  | Qty | Unit    | Cost     | Cost%     |
|---|-----|---------|----------|-----------|
|  Tent           | 0   | \$10/ea | \$0/\$10 | XX% = 0%  |
|  Sleeping Bag  | 1   | \$5/ea  | \$5/\$10 | XX% = 50% |
|  Flashlight    | 1   | \$2/ea  | \$2/\$10 | XX% = 20% |
|  First-Aid Kit | 1   | \$3/ea  | \$3/\$10 | XX% = 30% |
| TOTAL   |     |         | \$10     |           |

Make bold when at or above 50%.





# Considering the Cost Percentage

The Lineltem cost percentage has a dynamic presentation and it's isolated.

| Cost%      |
|------------|
| <b>XX%</b> |
| <b>XX%</b> |
| <b>XX%</b> |

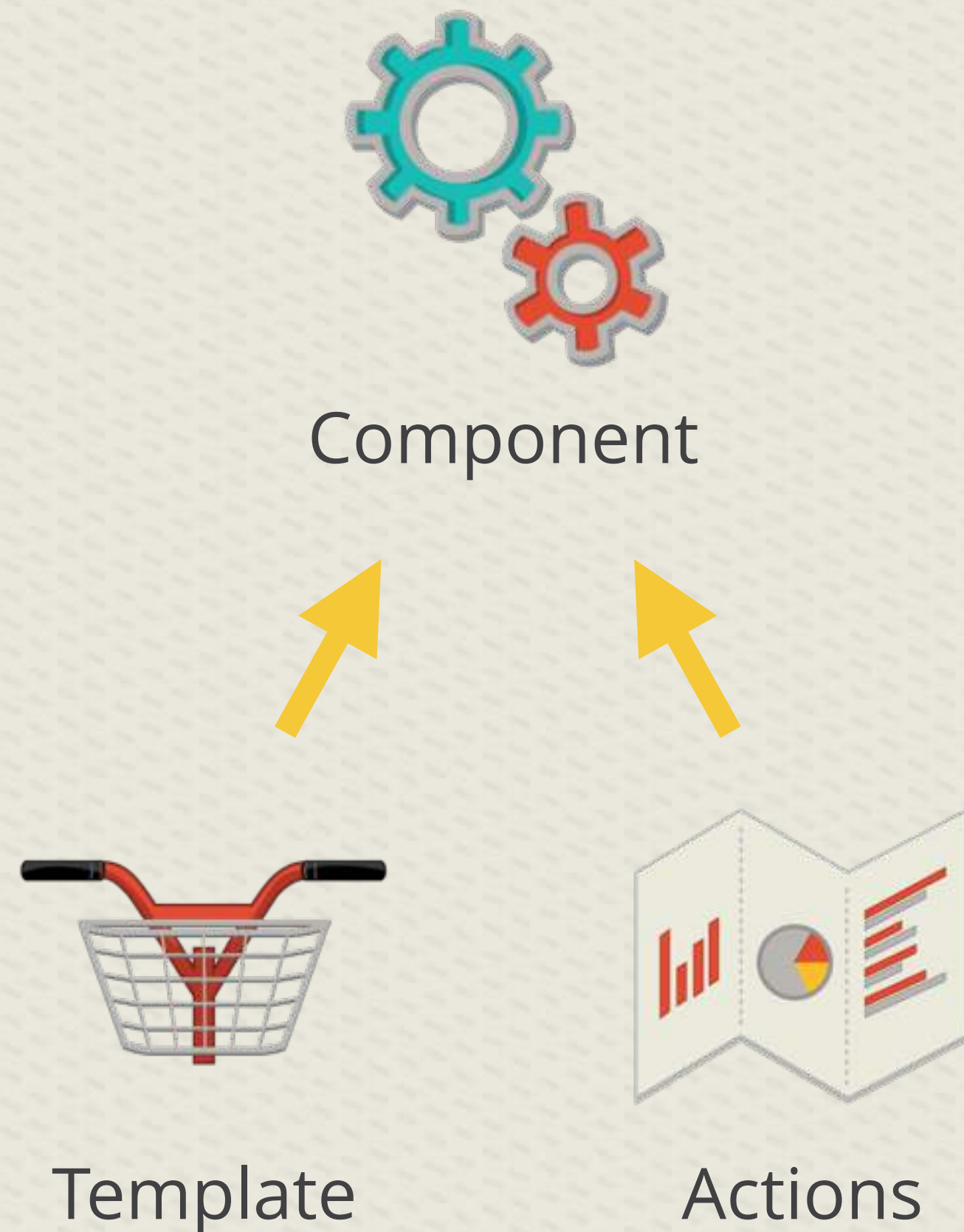
1. The presentation changes if it's at or above 50% (it's bolded)
2. The value isn't used anywhere else in the app
3. If a user clicks on the percentage, it'd be nice to show how it is calculated





# Introducing Components

Components are a reusable way to combine a template with action handling and behavior.



Used for:

- Charts
- Tabs
- Tree widgets
- Buttons with confirmations
- Date range selectors
- Wrap libraries and services

`{{link-to}}` and  
`{{input}}` are  
components!



Components follow closely  
with the W3C Custom  
Elements spec.

<http://w3c.github.io/webcomponents/spec/custom/>





# Generating a Component

Ember CLI provides a generator for creating a component and its template.

```
ember generate component <component-name>
```

Console

```
$ ember generate component item-percentage
```

```
installing component
```

```
  create app/components/item-percentage.js
```

```
  create app/templates/components/item-percentage.hbs
```

```
...
```

The component file

The component's template





# Creating the Component

Components are defined in `app/components` and extend `Ember.Component`.

`app/components/item-percentage.js` ←

```
import Ember from 'ember';
```

```
export default Ember.Component.extend({  
});
```

Components must have at least one hyphen in their name. This conforms with the Custom Elements spec.





# Defining the Component Logic

Components have their own properties, functions, and behaviors.

app/components/item-percentage.js

```
import Ember from 'ember';

export default Ember.Component.extend({
  percentage: Ember.computed('itemPrice', 'orderPrice', function() {
    return this.get('itemPrice') / this.get('orderPrice') * 100;
  })
});
```

We'll provide the `itemPrice` and `orderPrice` values from the calling template.





# Customizing the Component's Template

Component templates are defined in `app/templates/components`.

```
app/templates/components/item-percentage.hbs
```

```
<span>
  {{percentage}}%
</span>
```

The template name matches  
the component name.






# Rendering a Component From a Template

Components are called by name and may be passed additional data.

```
app/templates/orders/order.hbs
```

```
{{#each model.items as |lineItem|}}  
  <tr>  
    <td>{{lineItem.title}}</td>  
    <td>{{lineItem.quantity}}</td>  
    <td>{{lineItem.unitPrice}}</td>  
    <td>${{lineItem.price}}</td>  
    <td>  
      {{item-percentage itemPrice=lineItem.price orderPrice=model.price}}  
    </td>  
  </tr>  
{{/each}}
```







This renders the `item-percentage` component into this location.

This passes the `item` and `order` prices to the component.



# Viewing the Un-styled Percentage

The correct percentage is calculated and displayed, but the style does not yet change.

| Item  | Qty | Unit    | Cost | Cost% |
|---|-----|---------|------|-------|
|  Tent           | 0   | \$10/ea | \$0  | 0%    |
|  Sleeping Bag  | 1   | \$5/ea  | \$5  | 50%   |
|  Flashlight    | 1   | \$2/ea  | \$2  | 20%   |
|  First-Aid Kit | 1   | \$3/ea  | \$3  | 30%   |
| TOTAL   |     |         | \$10 |       |

But we're missing the bold styling.





# Adding an “isImportant” Property

Using the percentage value, the component determines if it's an important (high) value.

app/components/item-percentage.js

```
import Ember from 'ember';

export default Ember.Component.extend({
  isImportant: Ember.computed.gte('percentage', 50),

  percentage: Ember.computed('itemPrice', 'orderPrice', function() {
    return this.get('itemPrice') / this.get('orderPrice') * 100;
  })
});
```

The greater than or equal to (gte) macro returns true if the property value is greater than or equal to the value given.





# Adding the Template Styling

Use the Handlebars `{{if}}` helper to dynamically set or unset an “important” class.

```
app/templates/components/item-percentage.hbs
```

```
<span class="{{if isImportant 'important'}}">
  {{percentage}}%
</span>
```



The span gets an “important” class if the component’s `isImportant` property is true.









# Viewing the Dynamic Styling

The percentages are now dynamically styled based on their value.

localhost:4200/orders/2345

Order details for Olivia Okapi (#2345)

| Item  | Qty | Unit    | Cost | Cost% |
|---|-----|---------|------|-------|
|  Tent           | 0   | \$10/ea | \$0  | 0%    |
|  Sleeping Bag  | 1   | \$5/ea  | \$5  | 50%   |
|  Flashlight    | 1   | \$2/ea  | \$2  | 20%   |
|  First-Aid Kit | 1   | \$3/ea  | \$3  | 30%   |
| TOTAL   |     |         | \$10 |       |

How about  
clicking to show  
the details?





# Adding a Component Action

We'll handle a "toggleDetails" action to toggle an "isShowingDetails" component property.

app/components/item-percentage.js

```
import Ember from 'ember';

export default Ember.Component.extend({
  actions: {
    toggleDetails() {
      this.toggleProperty('showDetails');
    }
  },

  isImportant: Ember.computed.gte('percentage', 50),
  percentage: ...
});
```

toggleProperty() switches a property's value between true and false each time it is called.



Actions are handled in a component's actions block, just like routes.



# Mapping the toggleDetails Action

Clicking on the span should toggle the details, so the action is mapped on the span element.

```
app/templates/components/item-percentage.hbs
```

```
<span class="{{if isImportant 'important'}}" {{action "toggleDetails"}}>
  {{percentage}}%
</span>
```

This action is using the default  
on="click" trigger.





# Conditionally Showing Details

The component's template is now dynamically filled with either percentage or detail contents.

app/templates/components/item-percentage.hbs

```
<span class="{{if isImportant 'important'}}" {{action "toggleDetails"}}>
  {{#if showDetails}}
    ${{itemPrice}} / ${{orderPrice}}
  {{else}}
    {{percentage}}%
  {{/if}}
</span>
```





# Application Complete!





Level 5.2

# Properties and Components

.....  
Components

