Binding context

A *binding context* is an object that holds data that you can reference from your bindings. While applying bindings, Knockout automatically creates and manages a hierarchy of binding contexts. The root level of the hierarchy refers to the `viewModel` parameter you supplied to `ko.applyBindings(viewModel)`. Then, each time you use a control flow binding such as `with` or `foreach`, that creates a child binding context that refers to the nested view model data.

Bindings contexts offer the following special properties that you can reference in any binding:

- `$parent`

  This is the view model object in the parent context, the one immeditely outside the current context. In the root context, this is undefined. Example:

```
<h1 data-bind="text: name"></h1>

<div data-bind="with: manager">
    <!-- Now we're inside a nested binding context -->
    <span data-bind="text: name"></span> is the
    manager of <span data-bind="text: $parent.name"></span>
</div>
```

- `$parents`

  This is an array representing all of the parent view models:

  `$parents[0]` is the view model from the parent context (i.e., it's the same as `$parent`)

  `$parents[1]` is the view model from the grandparent context

  `$parents[2]` is the view model from the great-grandparent context

  … and so on.

- `$root`

  This is the main view model object in the root context, i.e., the topmost parent context. It's usually the object that was passed to `ko.applyBindings`. It is equivalent to `$parents[$parents.length - 1]`.

- `$component`

  If you're within the context of a particular component template, then `$component` refers to the viewmodel for that component. It's the component-specific equivalent to `$root`. In the case of nested components, `$component` refers to the viewmodel for the closest component.

  This is useful, for example, if a component's template includes one or more `foreach` blocks in which you wish to refer to some property or function on the component viewmodel rather than on the current data item.

- `$data`

  This is the view model object in the current context. In the root context, `$data` and `$root` are equivalent. Inside a nested binding context, this parameter will be set to the current data item (e.g., inside a `with: person` binding, `$data` will be set to `person`). `$data` is useful when you want to reference the viewmodel itself, rather than a property on the viewmodel. Example:

```
<ul data-bind="foreach: ['cats', 'dogs', 'fish']">
    <li>The value is <span data-bind="text: $data"></span></li>
</ul>
```

- `$index` (only available within `foreach` bindings)

  This is the zero-based index of the current array entry being rendered by a `foreach` binding. Unlike the other binding context properties, `$index` is an observable and is updated whenever the index of

the item changes (e.g., if items are added to or removed from the array).

- `$parentContext`

  This refers to the binding context object at the parent level. This is different from `$parent`, which refers to the *data* (not binding context) at the parent level. This is useful, for example, if you need to access the index value of an outer `foreach` item from an inner context (usage: `$parentContext.$index`). This is undefined in the root context.

- `$rawData`

  This is the raw view model value in the current context. Usually this will be the same as `$data`, but if the view model provided to Knockout is wrapped in an observable, `$data` will be the unwrapped view model, and `$rawData` will be the observable itself.

- `$componentTemplateNodes`

  If you're within the context of a particular component template, then `$componentTemplateNodes` is an array containing any DOM nodes that were passed to that component. This makes it easy to build components that receive templates, for example a grid component that accepts a template to define its output rows. For a complete example, see passing markup into components.

The following special variables are also available in bindings, but are not part of the binding context object:

- `$context`

  This refers to the current binding context object. This may be useful if you want to access properties of the context when they might also exist in the view model, or if you want to pass the context object to a helper function in your view model.

- `$element`

  This is the element DOM object (for virtual elements, it will be the comment DOM object) of the current binding. This can be useful if a binding needs to access an attribute of the current element. Example:

```
<div id="item1" data-bind="text: $element.id"></div>
```

Controlling or modifying the binding context in custom bindings

Just like the built-in bindings `with` and `foreach`, custom bindings can change the binding context for their descendant elements, or provide special properties by extending the binding context object. This is described in detail under creating custom bindings that control descendant bindings.