

Knockout Bindings

Steve Michelotti

www.pluralsight.com

<http://geekswithblogs.net/michelotti>



Knockout Key Features

Observables (3 types)

Built-in Bindings (4 types)

Custom Bindings

KO Utilities

Data Features

Knockout Bindings

Built-in Bindings

- Text and Appearance
- Forms
- Control Flow
- Templates

Custom Bindings

Knockout Key Features

Observables (3 types)

Built-in Bindings (4 types)

Custom Bindings

KO Utilities

Data Features

Built-in Bindings – Type 1: Text and Appearance

visible

- Toggle visible/invisible of DOM element

text

- Text value of DOM element

html

- Raw HTML of DOM element

css

- CSS class(es) of DOM element

style

- Raw style attribute of DOM element

attr

- Any arbitrary attribute of DOM element

Built-in Bindings – Type 2: Forms

click	• Handler invoked when DOM element clicked
event	• Handler invoked for arbitrary event on DOM element
submit	• Handler invoked when form submitted
enable	• DOM element enabled if true
disable	• DOM element disabled if true
value	• Value of DOM element
checked	• Attached to checkbox and radio button
options	• Collection of elements in dropdown or multi-select
selectedOptions	• Currently selected item(s) of dropdown or multi-select
uniqueName	• Ensures DOM element has “name” attribute

Built-in Bindings – Type 3: Control Flow

if

- Executes if condition is true

ifnot

- Executes if condition is false

foreach

- Executes for each item in collection

with

- Executes for object specified (child models)

Built-in Bindings – Type 4: Templates

JavaScript templates

- Traditional JavaScript template in `<script>` tag

Containerless Control Flow (Anonymous)

- Template-less, comment-based syntax

DEMO

Built-in Bindings

Knockout Key Features

Observables (3 types)

Built-in Bindings (4 types)

Custom Bindings

KO Utilities

Data Features

Custom Bindings

- Do not limit yourself to built-in bindings!
- Custom Bindings: the most useful extensibility point of Knockout

```
ko.bindingHandlers.yourBindingName = {  
  init: function (element, valueAccessor, allBindingsAccessor, viewModel) {  
    // This will be called when the binding is first applied to an element  
    // Set up any initial state, event handlers, etc. here  
  },  
  update: function (element, valueAccessor, allBindingsAccessor, viewModel) {  
    // This will be called once when the binding is first applied to an element,  
    // and again whenever the associated observable changes value.  
    // Update the DOM element based on the supplied values here.  
  }  
};
```

DEMO

Custom Bindings

Summary

Built-in Bindings

- Text and Appearance
- Forms
- Control Flow
- Templates

Custom Bindings