

An Introduction to Knockout.js

with Daniel Doezema

dan.doezema.com

Knockout.js

- Standalone / Pure JavaScript
- Open Source (MIT license)
- Wide Browser Support
 - (IE 6+ / FF 2+ / Chrome / Opera / Safari - Mobile)
- Framework Agnostic

Best Use Cases

- Web Apps
- User Interaction
- UI / UX Priority
- AJAX / API Backed

Benefits

- Reusability (DRY)
- View Consistency
- Concise Code
- Logic / Responsibility Encapsulation

Model-View-View Model (MVVM)

- Model - Persistent Data Store
- View - UI / Layout
- View Model - The Staging Area

View Model (JavaScript)

```
var ViewModel = function(){  
    this.name    = "Daniel Doezema";  
    this.site    = "http://dan.doezema.com";  
    this.age     = 28;  
};
```

```
var view_model = new ViewModel();
```

View (HTML)

```
<h1>Presenter</h1>
```

```
<h3>Name</h3>
```

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

```
<h3>Site</h3>
```

```
<p data-bind="text:site"></p>
```

```
<h3>Age</h3>
```

```
<p data-bind="text:age"></p>
```

- Bind Attribute

- Knockout "Binding"

- View Model Property

Activation / Page Bootstrap

```
$(function($){  
    var view_model = new ViewModel();  
  
    ko.applyBindings(view_model);  
});
```


Rendered View

Presenter

Name

Daniel Doezema

Site

<http://dan.doezema.com>

Age

28

Observables

// Static Properties

```
var ViewModel = function(){  
    this.name = "Daniel Doezema";  
    this.site = "http://dan.doezema.com";  
    this.age = 28;  
};
```

// Observable Properties

```
var ViewModel = function(){  
    this.name = ko.observable("Daniel Doezema");  
    this.site = ko.observable("http://dan.doezema.com");  
    this.age = ko.observable(28);  
};
```

Observables

- Method, acting as property...

```
view_model.name();           // Getter  
view_model.name("John Doe"); // Setter
```

- Notifies Subscribers
- Detects Dependencies
- Reflects Changes of Bindings

Subscribers

Bi-Directional Bindings

```
<input data-bind="value:name" />
```

```
<input type="checkbox" data-bind="checked:is_admin" />
```

Subscribers

Computed Observables

```
var ViewModel = function(){  
    var self = this;  
  
    self.signature = ko.computed(function(){  
        return self.name() + " - " + self.site();  
    });  
  
};
```

```
<p data-bind="text:signature"></p>
```

Observable Arrays

```
var view_model = new ViewModel();
```

```
view_model.links = ko.observableArray([  
    {name:"Me",    url:"http://dan.doezema.com"},  
    {name:"Apple", url:"http://apple.com"},  
    {name:"Twitter", url:"http://twitter.com"}  
]);
```

```
ko.applyBindings(view_model);
```

ObservableArray + `foreach` Binding

```
<ul data-bind="foreach:links">
  <li>
    <a data-bind="attr:{href:$data.url}, text:$data.name"></a>
  </li>
</ul>
```

- Me
- Apple
- Twitter

ObservableArray Methods

- .push
- .pop
- .shift
- .unshift
- .reverse
- .sort
- .splice
- .slice
- .remove
- .removeAll
- .destroy
- .destroyAll

In Action...

<http://kojs.dan.doezema.com>