

Knockout.

By : Vijay Shivakumar

<http://learnjs.in>





What is Knockout ?

A JavaScript MVVM Framework / Library

Model : Business Data

View : User Interface that is visible to user

View Model : how the data needs to be rendered (View Model is similar to model but will have presentation concerns)

Works with other libraries like
jQuery, Zepto etc.

<http://learnjs.in>





Why Knockout ?

Declarative Binding

Binding and 2 way or reverse data binding

Automatic UI Refresh

The UI gets updated when the model gets updated

Dependency Tracking

Where one property is dependent on one or more properties

Templating

Inbuilt support for templating





Getting started

Download the latest lib from
knockoutjs.com

CDN

<http://ajax.aspnetcdn.com/ajax/knockout/knockout-3.4.0.js>

<http://cdnjs.cloudflare.com/ajax/libs/knockout/3.4.0/knockout-min.js>

Any many to choose from

<http://learnjs.in>





Lets begin

Create view model

can be function or an object

And bind them to the view

with data-bind attribute followed by
appropriate binding

Start knockout by calling

```
ko.applyBindings(myViewModel);
```



Observables are JavaScript functions

3 Types of observables

Regular Observables :

strings, boolean or any other simple values

Array Observables

Used for collections and can trigger events when value is added removed or modified in an array.

Computed Observables

You can take advantage of a value change event of more than one observable





Regular observables

Declare a view model

```
function viewModel() {  
    this.message = "welcome to your life";  
}
```

Start knockout

```
ko.applyBindings( new viewModel() );
```





Regular observables

Declare a view model

```
function viewModel() {  
  this.message = ko.observable("message")  
  ;  
}
```

Start knockout

```
ko.applyBindings( new viewModel() );
```

Use the Observable Model

```
var msg = viewModel.message( ); // read  
viewModel.message("new message"); //  
write
```





Observable Array

To be used with collections

Can detect changes on and in the collections

Advantages

- Works across browser

- Dependency tracking

- Cleaner syntax (simple api)





Observable Array methods

`indexOf("value")` – returns the index position of the value or -1

`slice(2,4)` returns values between start and end values

`push` – add a new value at the end

`pop` – remove the last value

`unshift` – add a new value in the start

`shift` – remove the first value

`reverse` – reverses the order

`sort` – sorts the item

`remove("value")` – removes the specified value

`removeAll` – removes all the values



Knockout. Computed Observable

Can use the existing values and build on it...

Dependent observable or computed observable
is a method that takes 2 parameters

- First parameter is function with computed return

- Second parameter is the viewmodel or the this keyword
to set the scope



A command in knockout to assign a value
Binding can be classified in to two parts

Built-in bindings :

types of built-in binding

4

Text and Appearance

Form binding

Control Flow

Templates

Custom bindings :





Text and Appearance

visible	: toggle visible or invisible of DOM elements
text	: textContent of DOM elements
html	: innerHTML content of DOM elements
CSS	: css classes of DOM elements
style	: raw / native styles (inline) of DOM elements
attr	: any other attributes of DOM element



- click : invoked when the DOM is clicked
- event : any other events of DOM like mouseover, mouseout etc.,
- submit : invoked when form is submitted
- enable : enables the DOM element
- disable : disable the DOM element
- value : value of input tags
- checked : used with checked property of radio & checkbox
- options : when working with dropdown list
- optionsText : text nodes in option element
- optionsValue : value attributes in option element
- selectedOptions : the current selected value in dropdown list
- uniqueName : assign unique name to DOM elements



Knockout.

valueUpdate binding

valueUpdate : 'keyup';

valueUpdate : 'keypress';

valueUpdate : 'change';

valueUpdate : 'afterkeydown';



Also called as anonymous templates

if : executes binding if returns true

ifnot : executes binding if returns false

foreach : executes binding for each item in collection

with : executes for a specific object
(used with child models)





Template Binding

Named Templates
Control Flow Template
Binding Context
Inline Templates
Dynamic Template
Template Binding Helpers
Containerless Bindings

