

Annotations

- See elsewhere ¹ See footnote
- ? Optional ^R Ruby on Rails

Viewmodel (JavaScript)

```
var vm = function(value) {
    this.variable = ko.observable(value);
};
ko.applyBindings(new vm('...'), DOM2);
```

Templates (HTML)

```
<div data-bind="bindings" >html</div>
<!-- ko bindings -->html<!-- /ko -->
<script type="text/html" id="T">html</script>
```

^{MSIE} Always quote if in { 'if': ... }

bindings.

Observables

```
.isObservable .isWritableObservable
() .writeable()
.isComputed() .valueHasMutated()
.subscribe(function(v) {}, trg1,2, event?) : s
```

```
.notifySubscribers(currentvalue)
```

```
s.dispose()
```

¹ Default to this.

² change (default) or before Change.

ko.observable(value)

```
() : value (value) : vm
```

ko.observableArray(array)

```
() : 1 [] ([]) : vm
```

```
.pop() : item .push(item)
```

```
.shift() : item .unshift(item)
```

```
.remove(item) .removeAll()
```

```
.indexOf(item) : .reverse()
```

¹ /#

```
.slice(s, e2) : .splice(s, l2, ...2)
```

```
[] : []
```

```
.sort(function(a, b)?: -0/+2)
```

```
.destroy(item)R .destroyAll()R
```

ko.computed(function() {})¹

```
() : value
```

¹ Use ko.pureComputed if only observables.

Appearance bindings

```
text: string html: string
```

```
css: { class: bool, ... } style: { css: value, ... }
```

```
attr: { attribute: value, ... }
```

```
visible: bool template: string
```

"template" appearance binding

```
name: string nodes: [DOM, ...]
```

```
data: [] if: bool
```

```
foreach: [] as: string
```

```
afterRender: function([DOM], context)
```

```
afterAdd: function before Remove: function
n
```

Control flow bindings

```
foreach: []/{}→ with: observable
```

```
if: bool ifnot: bool
```

```
component: observable
```

"foreach" control flow binding

```
data: [] as: string
```

```
afterRender: function
```

```
afterAdd: function before Remove: function
n
```

```
before Move: function afterMove: function
n
```

```
includeDestroyedR: bool
```

Form bindings

```
click: function event: { event: function }
```

```
submit: function
```

```
value: string textinput:1 string
```

```
enable: bool disable: bool
```

```
hasFocus: bool checked: bool
```

```
options: []/{}→ uniqueName: bool
```

¹ Updates value on keypress, clipboard, etc.

"options" form bindings

```
optionCaption: string
```

```
optionText: string optionValue
```

```
optionAfterRender: function
```

```
selectedOptions1: valueAllow
```

```
optionIncludeDestroyedR: bool
```

Only for multiple selects.

Context variables

```
$root: vm $component: vm
```

```
$parent: vm $parents[index]: vm
```

```
$data: vm $rawData: vm
```

```
$element: $index1: integer
```

DOM

```
$componentTemplateNodes:
```

DOM array

```
$context: $parentContext:
```

context context

¹ Only available in foreach loops.

Writable ko.computed(...)

```
this.value = ko.observable();
this.doubled = ko.pureComputed({
    read: function () { return
this.value * 2; },
    write: function(v) {
this.value(v / 2); },
    owner: this
});
```

Builtin extenders

```
.extend({ rateLimit: { timeout: milliseconds, method: method1,2 } })
```

```
.extend({ notify: 'always' });
```

¹ notifyAtFixedRate/notifyWhenChangesStop

Custom extenders

```
1 Get length: () .length. Get item: () [index]
```



By **mwvlee**
cheatography.com/mwvlee/

Published 2nd April, 2015.
Last updated 13th May, 2016.
Page 1 of 2.

```
ko.extenders.log =  
function(target, option) {  
    target.subscribe(function(newValue) {  
        console.log(option + ": " + newValue);  
    });  
    return target;  
};
```

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>