

### 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>
MSIEAlways quote if in { 'if': ... }
bindings.
```

### Observables

```
.isObservable .isWritableObservable()
e() e()
.isComputed( .valueHasMutated() )
.subscribe(function(v) {}, trg1?,
event2) :s
.notifySubscribers(currentvalue)
s.dispose()
1 Default to this.
2 change (default) or beforeChange.
```

### 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,
[] ...?) :[]
.sort(function(a, b) ? : -/0/+2)
.destroy(item)R .destroyAll()R
```

<sup>1</sup> Get length: ().length. Get item: () [index]

### ko.computed(function() {})'

```
() :value
1 Use ko.pureComputed if only
observables.
```

### Appearance bindings

```
text: string html: string
css: { style: {
class:bool, ...} 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: beforeRemove:
function function
```

### Control flow bindings

```
foreach: []/{ }→ with: observable
if: bool ifnot: bool
component: observable
```

### "foreach" control flow binding

```
data: [] as: string
afterRender: function
afterAdd: beforeRemove:
function function
beforeMove: afterMove: function
function
includeDestroyedR: bool
```

### Form bindings

```
click: event: { event: function
function }
submit: function
value: string textinput:1 string
enable: bool disable: bool
hasFocus: checked: bool
bool
options: []/{ } uniqueName: bool
→
1 Updates value on keypress, clipboard, etc.
```

### "options" form bindings

```
optionsCaption: string
optionsText: optionsValue:
string string
optionsAfterRender: function
selectedOptions valueAllowUnset:
1: [] bool
optionsIncludeDestroyedR: bool
1 Only for multiple selects.
```



By **mwvdee**  
[cheatography.com/mwvdee/](http://cheatography.com/mwvdee/)

Published 2nd April, 2015.  
 Last updated 22nd September, 2015.  
 Page 1 of 2.

Sponsored by **Readability-Score.com**  
 Measure your website readability!  
<https://readability-score.com>

### Context variables

<code>\$root</code> : <i>vm</i>	<code>\$component</code> : <i>vm</i>
<code>\$parent</code> : <i>vm</i>	<code>\$parents [index]</code> : <i>vm</i>
<code>\$data</code> : <i>vm</i>	<code>\$rawData</code> : <i>vm</i>
<code>\$element</code> : <i>DOM</i>	<code>\$index<sup>1</sup></code> : <i>integer</i>
<code>\$componentTemplateNodes</code> : <i>DOM array</i>	
<code>\$context</code> : <i>context</i>	<code>\$parentContext</code> : <i>context</i>

<sup>1</sup> Only available in `foreach` loops.

### Writable `ko.computed(...)`

```
this.value = ko.observable();
this.double = ko.pureComputed({
  read: function () { return this.value * 2;
},
  write: function(v) { this.value(v / 2); },
  owner: this2
});
```

### Builtin extenders

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

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

<sup>1</sup> `notifyAtFixedRate`/`notifyWhenChangesStop`

### Custom extenders

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



By [mwwdlee](https://mwwdlee.com)

[cheatography.com/mwwdlee/](https://cheatography.com/mwwdlee/)

Published 2nd April, 2015.

Last updated 22nd September, 2015.

Page 2 of 2.

Sponsored by [Readability-Score.com](https://readability-score.com)

Measure your website readability!

<https://readability-score.com>