

Round 5: The Truth About Computed Observables

John Papa

Twitter: [@john_papa](https://twitter.com/john_papa)



Essential Knockout and JavaScript Tips

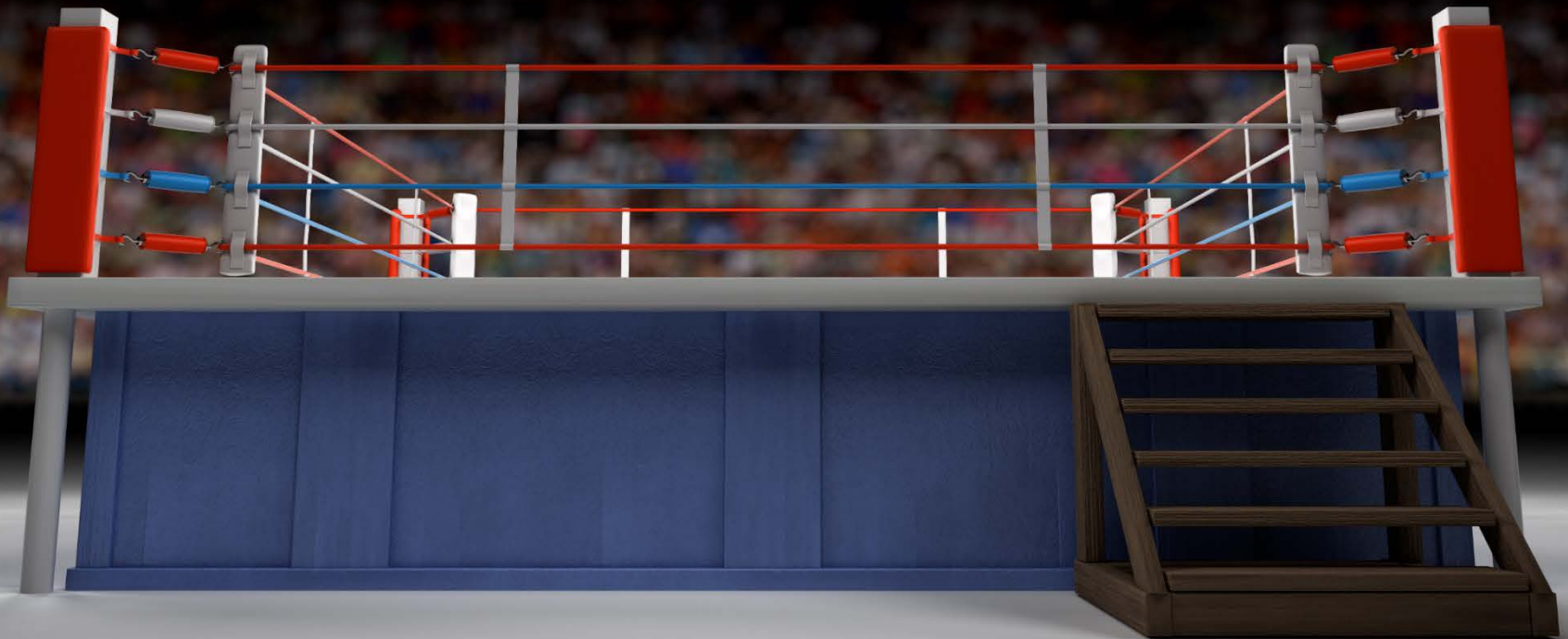
Round 5: The Truth About Computed Observables

John Papa
<http://johnpapa.net>
Twitter: @john_papa



pluralsight 
hardcore developer training

Take Advantage of the Power of Computeds



Computed Observables

$$\int_a^b f(x) dx = F(b) - F(a)$$

A Function that Evaluates an Expression.
Wrapped in an Observable

Calculating Expressions

```
vm.extendedPrice = ko.computed(function () {  
    return this.product() ?  
        this.product().price()  
        * parseInt("0" + this.qty(), 10) : 0;  
    },  
    vm  
);
```

Defining Computed Arrays

```
productsToShow = ko.computed(function () {  
  
    if (showAll()) {  
        return products();  
    }  
    return ko.utils.arrayFilter(products(), function (p) {  
        return (p.isAcousticGuitar);  
        // Acoustic Guitars only  
    });  
})
```

Keys to Computed Observables

Derived and
Calculated
Properties

Re-evaluate
When
Observables
Change

Understand
Their Nuances

Timing and
Features

Understanding Timings for Computed



Timings When Computed Evaluate

```
var canSave = ko.computed(function() {  
    return hasChanges() && !isSaving();  
});
```

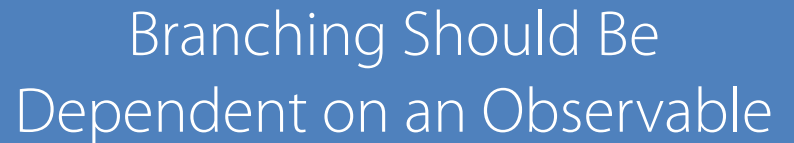
When Either Observable Changes,
canSave Re-evaluates

Re-Evaluating Computed

```
var adminRoutes = ko.computed(function() {  
  return router.allRoutes().filter(function(r) {  
    return r.settings.admin;  
  });  
});
```

Branch on Observables, or Not at All

Branching Should Be
Dependent on an Observable



```
productsToShow = ko.computed(function () {  
  if (showAll()) {  
    return products();  
  } else {  
    return filteredProducts();  
  };  
}),
```

When Observables That Were Just Evaluated Change,
the Computed Re-evaluates

Disposing of Resources using `disposeWhenNodeIsRemoved`



When to Dispose of the Computed ?

```
var pre = document.createElement('pre');
    element.appendChild(pre);

var dumpJSON = ko.computed({
    read: function () {
        var en = allBindings.enable === undefined
            || allBindings.enable;
        return en ? toJSON(context, null, 2) : '';
    },
    disposeWhenNodeIsRemoved: element
});
```

When the Element is Removed,
Computed is Disposed

Deferring Evaluation



Deferring Initial Evaluation

```
function getShowSpinnerComputed() {  
  return ko.computed(  
    {  
      read: function () {  
        return !shell.activeItem()  
          || shell.activeItem.isActivating();  
      }  
    },  
    deferEvaluation: true  
  ));  
}
```

Causes Evaluation Wait Until the
Computed is Actually Accessed

Computed Can Be Writeable

```
function sessionInitializer(session) {  
  session.tagsFormatted = ko.computed({  
    read: function () {  
      var text = session.tags();  
      return text ? text.replace(/\|/g, ', ') : text;  
    },  
    write: function (value) {  
      session.tags(value.replace(/\\, /g, '|'));  
    }  
  });  
}
```

Ideal When Bound to an
<INPUT> Element

Writeable Computed Observables



Powerful Computed Observable Features

Knowing
When and
Why
Evaluation
Occurs

Defer Initial
Evaluation

Bind to
Editable
Elements with
Writeable
Computed

Can Auto-
Dispose of
Computed