

Essential Knockout and JavaScript Tips

Round 1: Starting off on the Right Foot

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

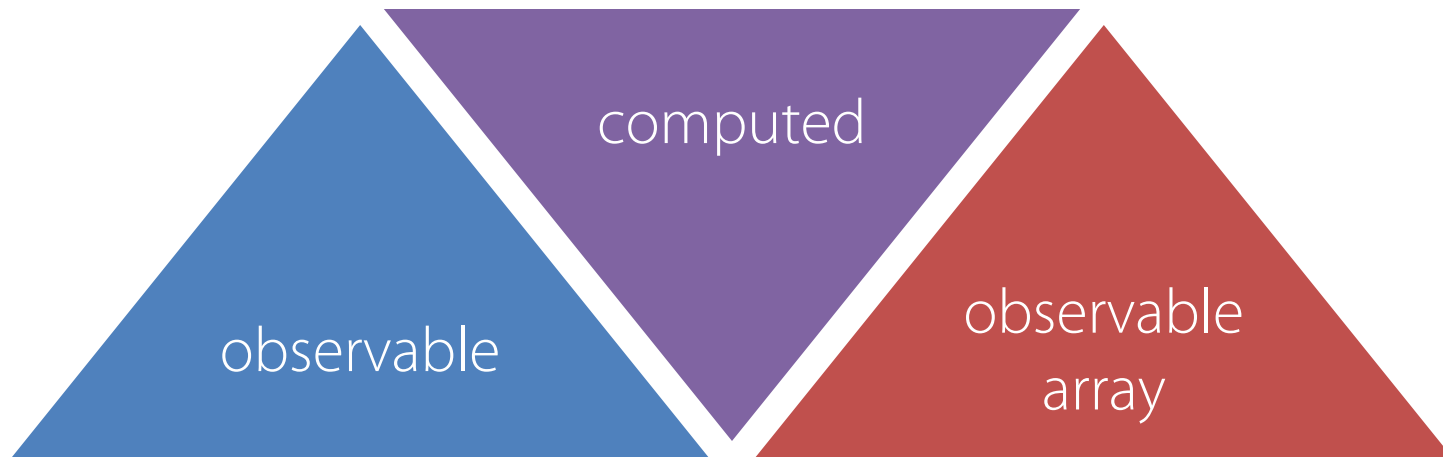


pluralsight 
hardcore developer training

Mastering Observables



Knockout's Observable Types



Knockout Observable

```
var name = ko.observable('Ha1ey');
```

```
var age = ko.observable(25);
```

How Knockout's Observable Works

Changes to
Source Notify
Targets

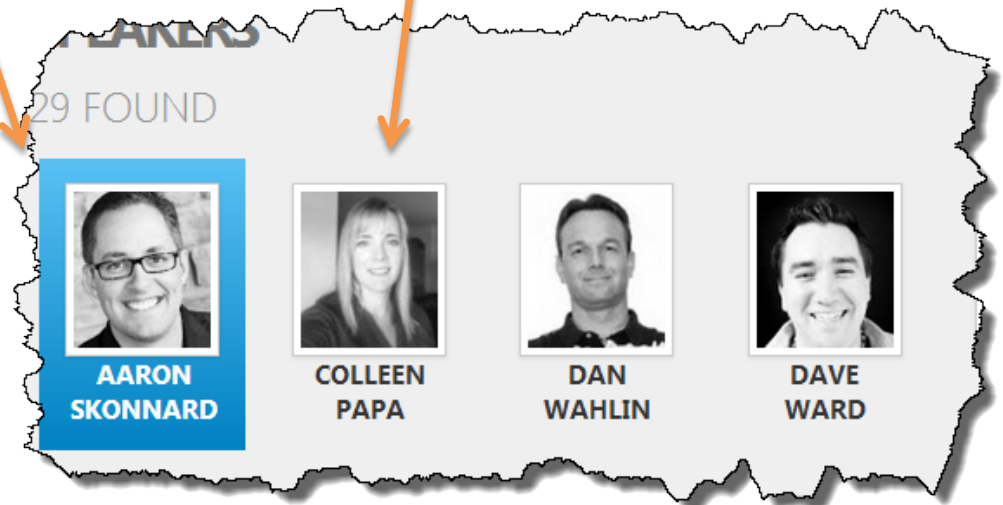
Changes to
Targets Notify
Source

Two-Way Data
Binding

Event Driven

Knockout in HTML/View (Read Only Targets)

```
<section data-bind="foreach: speakers">  
  <article>  
    <div>  
      <img data-bind="attr: { src: imageName }" />  
      <address data-bind="text: firstName"></address>  
      <address data-bind="text: lastName"></address>  
    </div>  
  </article>  
</section>
```

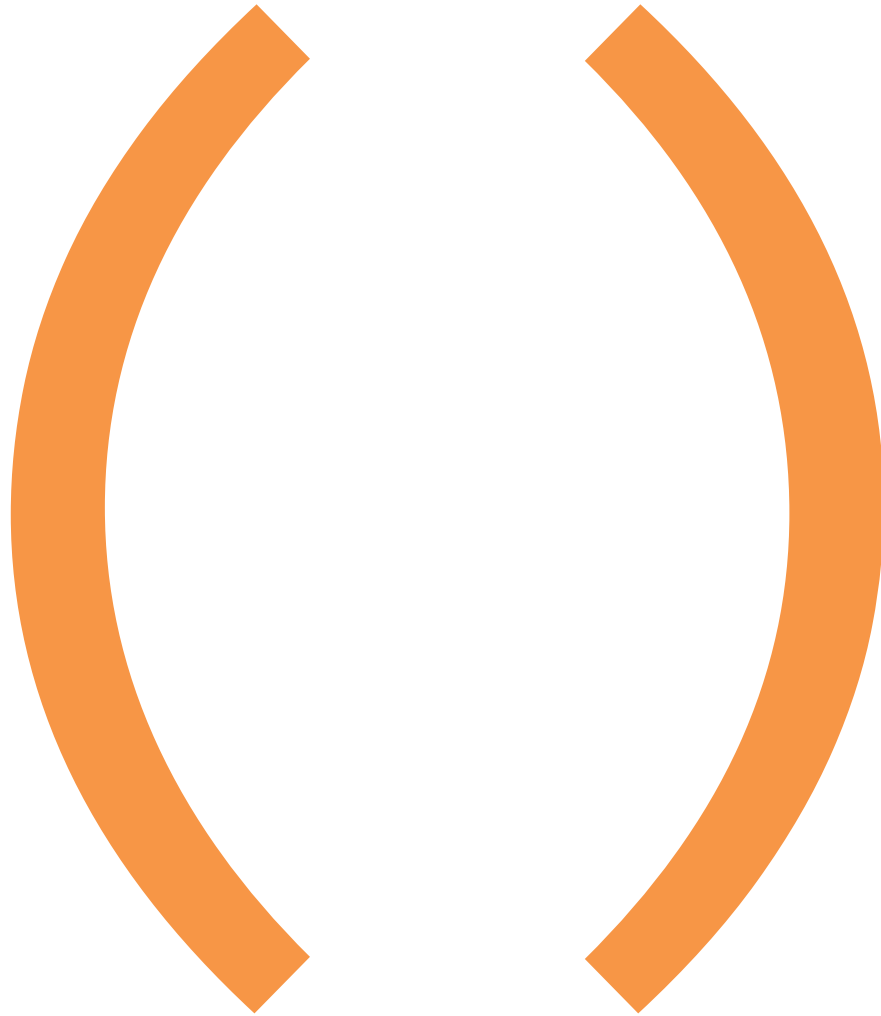


Knockout in HTML/View (Editable Targets)

```
<div>
  <label for="title">Title</label>
  <input id="title" data-bind="value: session().title"/>
</div>
<div>
  <label>Track</label>
  <select data-bind="options: tracks,
    optionsText: 'name', value: session().track">
  </select>
</div>
```

Title	<input type="text" value="SPA JumpStart"/>
Track	<div>Windows 8 ▼</div>
Time slot	<div>Sat 11:20 am ▼</div>

Observables are Functions



Unwrap Observables

```
name( 'Kat' );
```



Getting and Setting Observables

Read values
using
parentheses

```
var name = ko.observable('Hailey');
```

```
console.log(name);
```

Function definition

```
console.log(name());
```

Good!

Set values
using
parentheses

```
name('Ella');
```

Good!

```
name = 'Ella';
```

Redefined name to be a
string, not an observable

Observable Arrays are Functions Too

```
var Person = function(name){  
    this.name = ko.observable(name);  
}  
var p1 = new Person('Haley');  
var p2 = new Person('Ella');  
  
var people = ko.observableArray([p1, p2]);
```

```
people[0].name = 'Madelyn';
```

Missing parens on people
and name

```
people[0].name('Madelyn');
```

Missing parens on people

```
people()[0].name('Madelyn');
```

Good!

Use Parentheses



Unwrapping observable function in code



Dive into objects with declarative binding



`sessions().length`



`speaker().imagePath`

Right-Most Properties Don't Require ()

```
<section>
  <div data-bind="text: people().length"></div>

  <article data-bind="foreach: people">

    <address data-bind="text:name"></address>

    <address data-bind="text:age"></address>

    <address data-bind="text:address().city"></address>

  </article>

</section>
```


When Not to Use Observables ?



You May Not Need Observables When ...



Read Only
Data

One Time
Binding

Loading a List
of Objects

Mixed Objects



Not All Properties Need to be Observables

No need for raising
"property changed"

Mix Observables
with
Non-Observables

```
var Person = function(id, name, age){  
    this.id = id;  
    this.name = ko.observable(name);  
    this.age = ko.observable(age);  
}
```

```
var p1 = new Person(1, 'Hailey', 25);
```

```
var people = ko.observableArray();  
people.push(p1);
```

Observable Array of Plain Old JavaScript Objects (POJO)

```
var myViewModel = {  
  salesPerson: ko.observable("John"),  
  products: ko.observableArray([  
    { model: "Taylor 314CE", price: 1749, id=321 },  
    { model: "Martin D40", price: 1899, id=479 }  
  ])  
};
```

POJO's



Guidelines

Consider what
should be
observable

Don't be
overly
concerned,
though

Observable
overhead is
generally
negligible