Directive's Isolate Scope '=' and '@'

Isolate Scope

```
function MyDirective() {
  var ddo = {
    scope: { }
    ...
  };
  return ddo;
}
```

Signals Isolate Scope: Parent scope is NOT inherited

Bidirectional Property Binding

```
function MyDirective() {
  var ddo = {
    scope: {
      myProp: '=attributeName'
    }
    ...
  };
  return ddo;
}
```

HTML template attribute name

Local scope property

Bidirectional binding

Bidirectional Property Binding

```
function MyDirective() {
  var ddo = {
    scope: {
      myProp: '='
    }
    ...
  };
  return ddo;
}
```

Assumes the attribute is named the same as property name: my-prop

Bidirectional Property Binding

```
function MyDirective() {
  var ddo = {
    scope: {
      myProp: '=?'
    }
    ...
  };
  return ddo;
}
```

Signifies that the attribute is optional.

Bidirectional Property Binding (HTML)

```
<my-directive my-prop="outerProp">
</my-directive>
```

Attributes follow the same camelCase normalization
my-prop => myProp

DOM Attribute Property Binding

```
function MyDirective() {
  var ddo = {
    scope: {
      myProp: '@myAttribute'
    }
    ...
  };
  return ddo;
}
```

Binds myProp to the value of DOM attribute my-attribute

DOM Attribute Property Binding (HTML)

```
<my-directive my-attribute="{{outerProp}}">
</my-directive>
```

As the value of outerProp changes, so does the value of my-attribute and so does the value of myProp inside the directive

DOM Attribute Property Binding (HTML)

```
<my-directive
     my-attribute="Hi {{outerProp + '!'}}">
</my-directive>
```

Summary

- ✧ Having isolate scope on the directive
  - Breaks the prototypal inheritance of the scope from the parent
  - Makes the directive more independent, less coupled w/ controller
- ✧ We pass values into the directive using scope bindings
- ✧ Bidirectional binding (‘=‘) is such that directive scope property change affects the bound property and visa versa
- ✧ DOM attribute value binding (‘@’) always results in directive property being a string
  - Changes to DOM attribute value are propogated to the directive property, but not the other way around