Week 3

coursera.org/learn/single-page-web-apps-with-angularjs/discussions/weeks/3/threads/1bha-A55Eee1sA6jIVM2EA

SHORT REVIEW For Week 3 - Part 3: The Reference Binding ('&')

The **&** sign signifies a **Reference Binding** that allows us to execute an expression in a child scope but in the context of a parent scope. For example, a value in a directive's template that is bound to an argument in a controller method's.

Going through steps:

a) In the Controller's function declaration:

```
function myController() {

  ctrl = this;

  ctrl.myCtrlMethod= function (myArg) {

    ...

  };

}
```

> Declare a method with argument to which data will be passed back from an isolate scope.

b) In the parent's scope HTML template:

```
<div ng-controller="myController as ctrl">

  <my-directive myParentAttribute="ctrl.myCtrlMethod(myArg)">

  </my-directive>

</div>
```

1. Declare an attribute providing a method reference to call on the parent (**myParentAttribute**);
2. Declare argument keys for directive to bind values to (**myArg**) - from isolate scope back to parent scope.

c) In the isolate scope's Directive Definition Object (DDO):

```
function myDirective() {
  var ddo = {
    templateUrl: 'myDirectiveTemplate.html',
    scope: {
      myDirectiveIsolateMethod: '&myParentAttribute'
    }
  };
  return ddo;
}
```

> The directive's **myDirectiveIsolateMethod** (an alias method) is bound to the controller's **ctrl.myCtrlMethod()** through the reference binding '**&**myParentAttribute'.

d) In the Directive's Template:

```
<button ng-click="myDirectiveIsolateMethod( {myArg: val1} )">
  Do Something
</button>
```

So, how does it work: the argument object **{myArg: val1}** is passed through the ddo's isolate scope ***&myParentAttribute*** as ***reference***, mapping its object's key **myArg** straight to the parent's HTML method ***myCtrlMethod***'s argument "***(myArg)***" and so passing the object's value of **val1** into it (this parent's HTML method is, in turn, the controller function's method).