Different properties of directive :

a1)restricta2)scope a3)template a4)templateUrl a5)controller a6)link

Use it with attribute | ‘E’, ‘A’, class | restrict : ‘C’, comment | ‘M’ | replace : ‘true’

restrict  |template  | replace

Ref : https://www.sitepoint.com/practical-guide-angularjs-directives/

* Note :replace – This specifies if the generated template will replace the HTML element on which the directive is attached. In our case we have used the directive as <hello-world></hello-world>, and replace is set to true. So, after the directive is compiled, the produced output template replaces <hello-world></hello-world>. The final output is <h3>Hello World!!</h3>. If you set replace to false, the default, the output template will be inserted into the element on which the directive is invoked.

9a)Create a new custom directive with company details .

employee\_id, employeefirstname ,employeelastname, salary

9a1)Create a custom directive pass length f3 side of the triangle.Display the area of the triangle.

9a2)

9a3)Create a simple directive with link events as mouseover,mouseout.Change the color of paragraph tag.

9a4)Change image src on mouseover and out.

Inside the mouseover do scope.$apply to change the scope value.

## Note : $apply enables to integrate changes with the digest cycle. <http://stackoverflow.com/questions/15112584/how-do-i-use-scope-watch-and-scope-apply-in-angularjs>

Note **:** When that button is clicked the $scope.updateTime() function is called, and after that AngularJS calls $scope.$digest() so that data bindings are updated.

**Note** : You can think of the $apply **function as of an integration mechanism**. You see, each time you change some **watched variable attached to the** $scope object directly, AngularJS will know that the change has happened. This is because AngularJS already knew to monitor those changes. So if it happens in code managed by the framework, the digest cycle will carry on.

However, sometimes you want to **change some value outside of the AngularJS world** and see the changes propagate normally. Consider this - you have a $scope.myVar value which will be modified within a jQuery's $.ajax() handler. This will happen at some point in future. AngularJS can't wait for this to happen, since it hasn't been instructed to wait on jQuery.

To tackle this, $apply has been introduced. It lets you to start the digestion cycle explicitly. However, you should only use this to migrate some data to AngularJS (integration with other frameworks), but never use this method combined with regular AngularJS code, as AngularJS will throw an error then.

9a5)Also use scope.$watch for checking any change in the scope variable last\_name. If yes then display some message with console.log

elem.bind('click', function() {

elem.css('background-color', 'white');

scope.$apply(function() {

scope.color = "white";

});

9a2)Add the scope.$apply for background-color,color on mouseover and out.

Also give pointer (hand) on mouseover.

element.css(‘cursor’,’pointer’)

Note :The link function is mainly used for attaching event listeners to DOM elements, watching model properties for changes, and updating the DOM.

9a3)Write a directive such that initialize the $scope.carsPrint every cars in li tag with strong tag through directive. Take restrict as ‘A’,’C’

Css : font-size , color

<li ng-repeat="x in movies">

<my-directive name='{{x}}'></my-directive>

</li>

</ul>

First : use the ‘@’ scope tag

9a4)Create a directive superman so that it will alert ‘I am superman’. There should be no parameter to the link function. Again make two directive with superman and stronger .<span superman stronger></span>.Both will alert something..

Ref :<https://egghead.io/lessons/angularjs-isolate-scope-two-way-binding>

9a5)Write a script by making use of $watch . Create a form for the employee containing emp\_first\_name, emp\_last\_name, emp\_sal . Add button so that it can be pushed to the array.When the size os array increases greater then 5 the make the table background as red.

Splitting the scope :

All are isolate scope :

* @ - pass by value (attribute binding)
* & - pass by method (expression binding)
* = - pass by reference (two-way data binding)

*Ref :<https://thinkster.io/egghead/isolate-scope-am>*

*Ref :https://www.youtube.com/watch?v=STKRRuSkL3c*

Isolate Scope @

Q1)Create a directive for first assigning the flavor through the tag itself like <myDirective flavor=”Vanila”>, later on do it with the help of Scope @.

Again the flavor from the controller.like : <myDirective flavor=”{{mtTest}}”>. myTest is initializing from the controller.

Isolate scope =

Application :

Q1)Makea application to add any item to the list and remove it using splice(x,1);

**if ($scope.products.indexOf($scope.addMe) == -1) {**  
            $scope.products.push($scope.addMe);  
**} else {**  
**$scope.errortext = "The item is already in your shopping list.";**  
**}**

9a3)angular directive compile phase : <https://www.bennadel.com/blog/2794-when-do-you-need-to-compile-a-directive-in-angularjs.htm>

Note :link and compile do not work together.

In the directive definition object, if you **only** define link, that's like shorthand for having an empty compile function with an empty preLink function with your code in the postLink function. As soon as you define compile, link is ignored by angular, because compile should return the linking functions.

If you only return one function from compile, then it'll be executed **post** link.

Or, put differently, link is just a shortcut to the postLink function that gets called after the scope has been linked by compile.

10)Write an angular script for the email validation.Make a form and then an email textfield.validate email with

Application status : directive can provide the status of the application data.(invalid,dirty,touched,error)

Updating last line..