**Directive** - At a high level, directives are markers on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS's HTML compiler ($compile) to attach a specified behavior to that DOM element or even transform the DOM element and its children.

**Usage of Directive**

Create a reusable HTML elements, attributes, and classes

Modify the behavior of an HTML element.

**ng-model directive -** It creates property in scope object if property does not exist and set value that is value of HTML controls. ng-model = "PropertyName"

The **ng-bind** directive binds application data to the HTML view or HTML tag.

**Four ways to specify Directive**

1) Tag

2) Attribute

3) class - <div class="ng-form">

4) HTML - comments

**Event Directives**

ng-Click, ng-MouseUp, ng-Change etc.

ng-BindTemplate - supports multiple bindings

ng-bind - supports single binding

**Usage of Custom Directive**

Create custom elements, Widgets

Create custom events

Observe and react to changes e.g. ngShow, ngHide

**Angular will allow same name for multiple directive but cause additional problem at runtime. So provide unique name to directives.**

Provide prefix to directive.

**compile function**

Use the compile function to change the original DOM (template element) before AngularJS creates an instance of it and before a scope is created.

That makes the compile function the perfect place to make changes to the DOM that should be applied to all instances later on, because it will only be run once and thus greatly enhances performance if you are stamping out a lot of instances.

The template element and attributes are passed to the compile function as arguments, but no scope is available.

**Pre-link function**

Use the pre-link function to implement logic that runs when AngularJS has already compiled the child elements, but before any of the child element's post-link functions have been called.

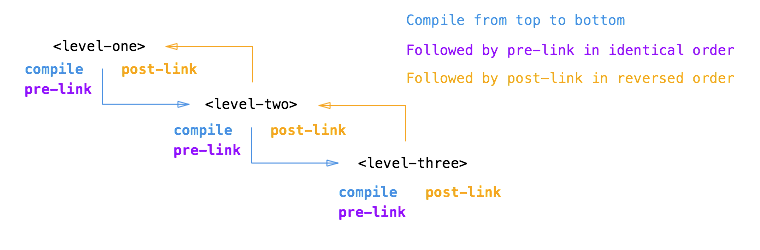
The scope, instance element and instance attributes are passed to the pre-link function as arguments

**Post-link function**

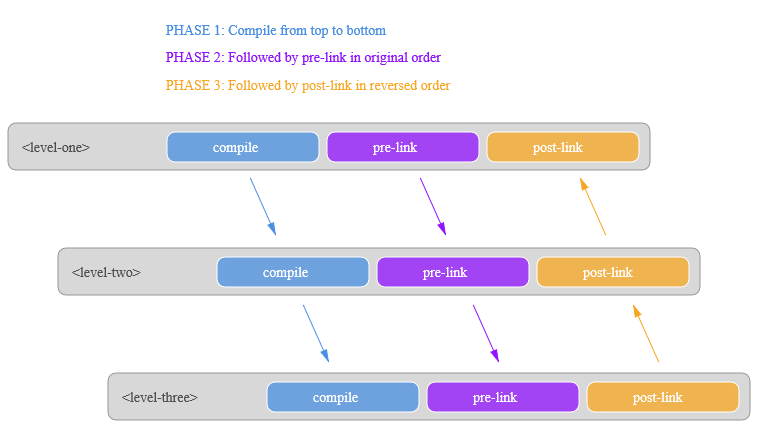
Use the post-link function to execute logic, knowing that all child elements have been compiled and all pre-link and post-link functions of child elements have been executed.

This is the reason the post-link function is considered the safest and default place for your code.

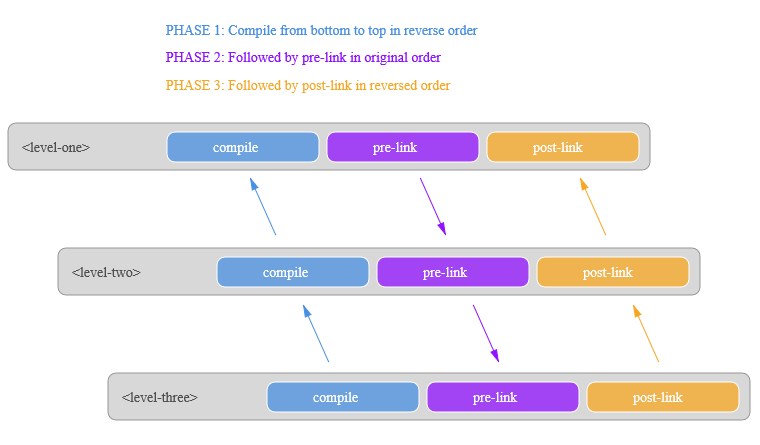
The scope, instance element and instance attributes are passed to the post-link function as arguments



**Without transclusion**



**With transclusion**



**Scope in Directive**

**Shared scope:** When there is no scope defined at directive then it uses the scope of element on which directive is applied. So controller and directive uses the same scope, they can access all variables and data set at scope.

**Inherited scope:** When scope=true is set in directive that is inherited scope. It creates new scope for directive but it still access data from parent scope (Scope of Controller, Directive)

Setting scope.title = "test" would create a new variable in scope even though it exist in parent scope. So be careful while accessing parent variables in directive.

**Isolated scope:**

Scope by default inherits from its parent scope, but this may not be desirable behavior, especially if you are building a re-usable widget. It is important that directives cannot accidentally read or write properties in the parent scope. This is where isolated scope comes in. Isolated scope does not prototypically inherit from the parent scope.

When scope = {} creates an isolated scope with no bindings.

**@** This sets up a one-way databinding from the parent scope to the isolated scope. If the parent scope changes, the isolated scope will reflect that change, but not the other way around.

**=** This sets up a two-way databinding from the parent scope to the isolated scope.

**&** Call an expression (function) on the parent scope from the isolated scope use the & symbol.

When two directives are on the same element, they share the same scope.

ng-repeat creates a scope for each item.