Angular Js

# Directives

## What are Directives?

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.

e.g. inbuilt directives ngBind, ngModel, and ngClass.

What does it mean to "compile" an HTML template? For AngularJS, "compilation" means attaching event listeners to the HTML to make it interactive. The reason we use the term "compile" is that the recursive process of attaching directives mirrors the process of compiling source code in compiled programming languages.

Best Practice: Prefer using the dash-delimited format (e.g. ng-bind for ngBind). If you want to use an HTML validating tool, you can instead use the data-prefixed version (e.g. data-ng-bind for ngBind). The other forms shown above are accepted for legacy reasons but we advise you to avoid them.

## Directive types

In the following way directive can be implemented.

* attribute name
* tag name
* comments
* class name

Best Practice: Prefer using directives via tag name and attributes over comment and class names. Doing so generally makes it easier to determine what directives a given element matches.

## Creating Directives

* Much like controllers, directives are registered on modules.
* To register a directive, you use the **module.directive** API.
* module.directive takes the normalized directive name followed by a factory function.
* This factory function should return an object with the different options to tell $compile how the directive should behave when matched.

Best Practice: Prefer using the definition object over returning a function.

# $templateCache