

ng-change DIRECTIVE IN ANGULARJS

- **ng-change** directive is used to specify the custom behavior that execute when you change the value of an HTML element.
- The **ng-change** directive requires the **ng-model** directive to be existing.
- In AngularJS, the **ng-change** directive will not override the HTML element's original **onchange** event, both will be executed.
- In AngularJS, the **ng-change** directive expression is evaluated immediately, unlike the JavaScript **onchange** event, which only activated at the end of a change.
- The **ng-change** directive is compiling at the priority level "0" (zero is a default priority level).
- The **ng-change** expression is only executed when a change in the input value causes a new value to be committed to the model.
- It will not be evaluated:
 1. If the value returned from the \$parser transformation pipeline has not changed
 2. If the input has continued to be invalid since the model will stay null.
 3. If the model is changed programmatically and not by a change to the input value



Syntax for ng-change directive in AngularJS:

```
<input ng-change= "expression" >  
.....  
</input>  
  
Or  
  
<ng-change ng-change= "expression" >  
.....  
</ng-change>
```

Parameter value for ng-change directive in AngularJS:

Value	Description
expression	It is used to define an expression that execute when the text of an element will be cut.

Sample coding for ng-change directive in AngularJS:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy AngularJS Tutorials</title>
    <script src="https://ajax.googleapis.com/ajax/libs/
      angularjs/1.5.6/angular.min.js">
    </script>
  </head>
  <body>
    <h2> Wikitechy ng-change Directive in AngularJS Tutorials </h2>
    <div ng-app="myApp" ng-controller="myCtrl">
      <label>Enter the text</label>
      <input type="text" ng-change="changeFunc()"
        ng-model = "myValue">
      <h3> The input field has been changed {{ count }} times </h3>
    </div>
    <script>
      angular.module('myApp', [])
        .controller('myCtrl', ['$scope', function($scope) {
          $scope.count = 0;
          $scope.changeFunc = function() {
            $scope.count++;
          };
        }]);
    </script>
  </body>
</html>
```



Code Explanation for ng-change directive in AngularJS:

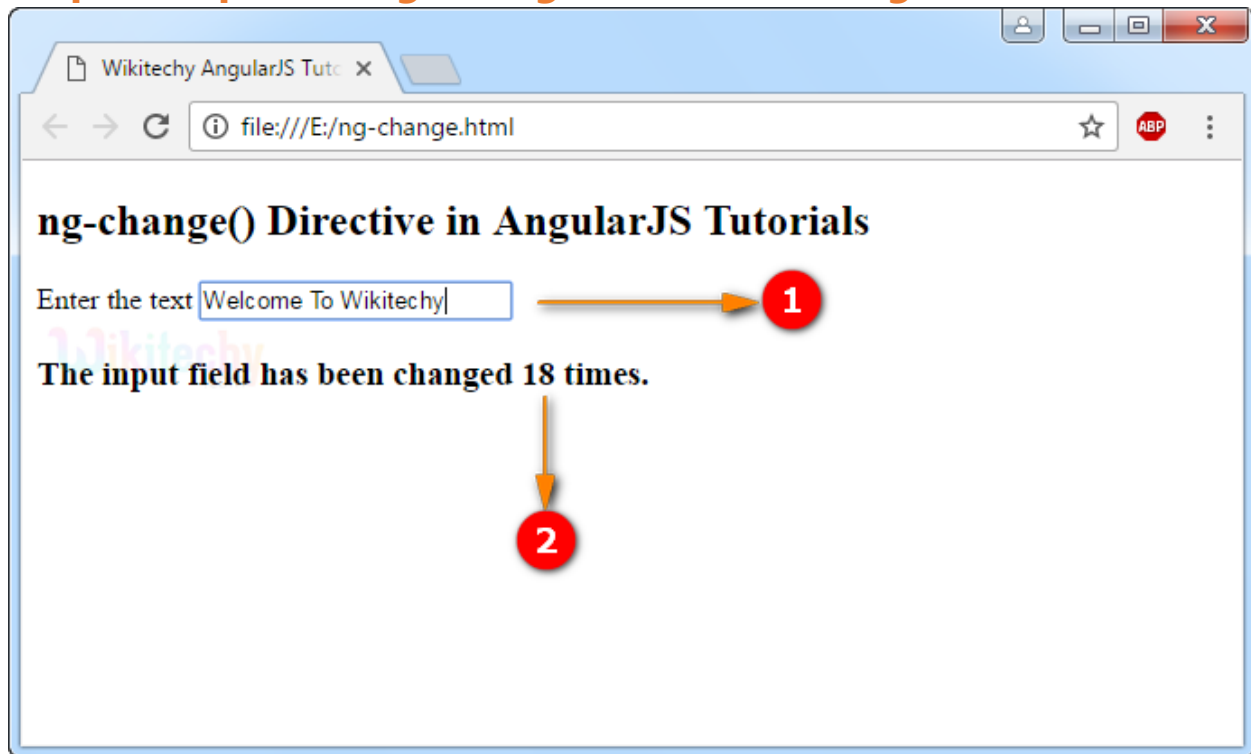
```
<!DOCTYPE html>
<html>
  <head>
    <title> Wikitechy AngularJS Tutorials</title>
    <script src="https://ajax.googleapis.com/ajax/libs/\
1      angularjs/1.5.6/angular.min.js">
    </script>
  </head>
  <body>
    <h2> ng-change() Directive in AngularJS Tutorials</h2>
    <div ng-app="myApp" ng-controller="myCtrl">
    <table>Enter the text</table>
    <input type="text" ng-change="changeFunc()" ng-model="myValue">
    <h3>The input field has been changed {{count}} times.</h3>
    </div>
    <script>
    angular.module('myApp', [])
    <div ng-app="myApp" ng-controller="myCtrl">
    <table>Enter the text</table>
    <input type="text" ng-change="changeFunc()" ng-model="myValue">
    <h3>The input field has been changed {{count}} times.</h3>
    </div>
    <script>
    angular.module('myApp', [])
    .controller('myCtrl', ['$scope', function($scope) {
    $scope.count = 0;
    $scope.changeFunc = function() {
    $scope.count++;
    };
    }]);
    </script>
  </body>
</html>
```

1. AngularJS is distributed as a JavaScript file, and can be added to a HTML page with a `<script>` tag.
2. The AngularJS application is defined by **ng-app="myApp"**. The application runs inside the **<div>** tag. It's also used to define a **<body>** tag as a root element. and **ng-controller=" myCtrl"** is used to define a controller.

3. The **ng-change= "changeFunc()"** is called a changeFunc() function when a change in the input value causes a new value to be committed to the model.
4. Here the **ng-model** is bind an input field value to AngularJS application variable by using ng-model name as **"myValue"**.
5. **angular.module()** function is used to create a module for application. We have passed an empty array to it.
6. The **.controller()** function is used to create a controller to an application.
7. Here we have set the value of **\$scope.count** as 0.
8. Here we have assign the function() as value to **\$scope.changeFunc**
9. Here the count value is increased by one by using **\$scope.count++**



Sample Output for ng-change directive in AngularJS:



1. The output shows that when the user start to typing the input field, the count value will be increased by one by using the ng-change directive which one is called the changeFunc() function.
2. The output displays a count value in the content **“The input field has been changed 18 times.”**