

input DIRECTIVE IN ANGULARJS

- AngularJS alter the default behavior of the **<input>** element, when the ng-model attribute is present.
- The **ng-model** attribute is combined with the HTML input element it provides the **data binding, validation** and **input state control**.
- HTML input element is control with AngularJS **data-binding** which means they are part of the ng-model and it can be referred and updated both in the AngularJS functions and in the DOM.
- An <input> element provides the validation, when the required attribute value is empty the \$valid state is false condition otherwise true.

Syntax for input directive in AngularJs:

```
<input ng-model="name"  
      [name = "string"]  
      [required="string"]  
      [ng-required="Boolean"]  
      [ng-minlength="number"]  
      [ng-maxlength="number"]  
      [ng-pattern="string"]  
      [ng-change="string"]  
      [ng-trim="Boolean"]
```



>

Parameter Values:

Parameters	Type	Description
ngModel	string	Defines angular expression to data-bind.
name (optional)	string	Name of the form under which the control is available.
required (optional)	string	Denotes the required validation error key if the value is not entered.
ngrequired (optional)	boolean	Denotes the required attribute is set to true state.
ngMinlength (optional)	number	States the minlength validation error key if the value is shorter than minlength.
ngMaxlength (optional)	number	States the maxlength validation error key if the value is longer than maxlength.
ngPattern (optional)	string	States pattern validation error key if the ng model value does not match RegExp found by evaluating the Angular expression given in the attribute value. If the expression evaluates to a RegExp object, then this is used directly.



		<p>If the expression evaluates to a string, then it will be converted to a RegExp after wrapping it in ^ and \$ characters.</p> <p>For instance, "abc" will be converted to new RegExp ('^abc\$').</p>
ngChange (optional)	string	An expression of Angular to be executed when input changes due to user interaction with the input element.
ngTrim (optional)	boolean	If set to false Angular will not automatically trim the input. Parameter is ignored for input [type=password] controls, which will never trim the input.

Properties of the input directive in AngularJS application:

Properties	Description
\$pristine	Denotes no fields has not been modified.
\$dirty	One or more fields has been changed.
\$invalid	Content in the form is not valid.
\$valid	Content in the form is valid.
\$touched	Denotes that the field has been touched
\$untouched	Denotes that the field has been not touched

CSS Class Properties of the input directive in AngularJS application:

Properties	Description
ng-pristine	Denotes no fields has not been modified.
ng-dirty	One or more fields has been changed.
ng-invalid	Content in the form is not valid.
ng-valid	Content in the form is valid.
ng-valid-key	One key for each validation. (Example: more than one thing to be validated.) ng-valid-required
ng-invalid-key	Example: ng-invalid-required
\$touched	Denotes that the field has been touched
\$untouched	Denotes that the field has been not touched

Sample coding for input 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" > </sc
ript>
  </head>
  <body>
    <h3>AngularJS input directive</h3>
    <form ng-app="myApp" ng-controller="inputCtrl" name="myform"
ng-submit="submitForm(myform.$valid)" novalidate>
      <input type="text" name="number" ng-model="number" required
ng-minlength="3" ng-change="changeFun()" ng-pattern="/^[0-9]+$/"
ng-maxlength="6">
      <span style="color:red" ng-show="myform.number.$dirty &&
myform.number.$invalid">
        <span ng-show="myform.number.$error.required">Phone is
required.</span>
        <span ng-show="myform.number.$error.pattern">Pattern does not
match</span>
        <span ng-show="myform.number.$error.minlength">Phone must be max 6
char</span>
      </span>
      <p>{{mychange}}</p>
      <input type="submit" value="submit" ng-click="Submit()" />
    </form>
    <script>
      var app = angular.module("myApp", []);
      app.controller("inputCtrl", function($scope) {
```



```
$scope.changefun = function() {  
  $scope.mychange="text changed";  
  }  
};  
</script>  
</body>  
</html>
```



Code Explanation input 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>
    <h3>AngularJS input directive</h3>
    <form ng-app="myApp" ng-controller="inputCtrl" name="myform"
      ng-submit="submitForm(myform.$valid)" novalidate>
      <input type="text" name="number" ng-model="number" required
        <span style="color:red" ng-show="myform.number.$dirty && myform.number.$invalid">
        <span ng-show="myform.number.$error.required">Phone is required.</span>
        <span ng-show="myform.number.$error.pattern">Pattern does not match
        </span>
        <span ng-show="myform.number.$error.minlength">Phone must be max 6 char
        </span>
        </span>
        </span>
        <p>{{mychange}}</p>
        <input type="submit" value="submit" ng-click="Submit()" />
      </form>
      <script>
        var app = angular.module("myApp", []);
        app.controller("inputCtrl", function($scope) {
          $scope.changefun = function() {
            $scope.mychange="text changed";
          };
        });
      </script>
    </body>
  </html>
```



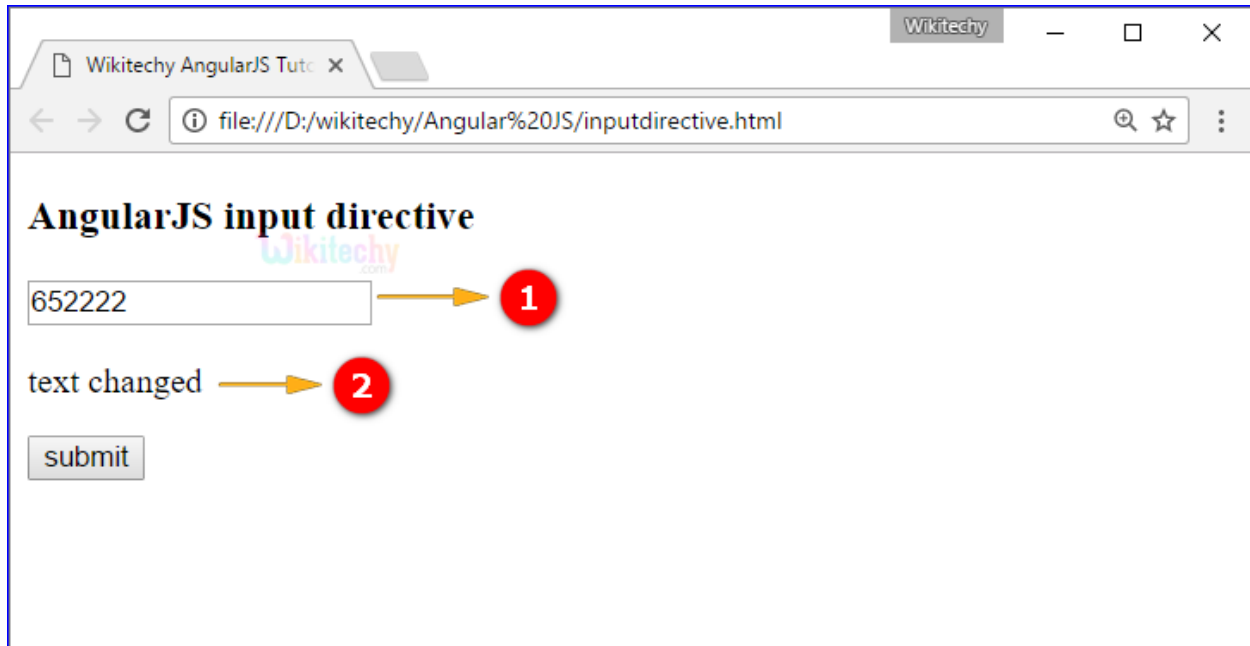
1. The **ng-app** specifies the root element (e.g. <body> or <html> or <div> tags) to define AngularJS application.
2. Ng-controller specifies the application controller in AngularJS the controller value is given as "inputctrl"
3. ng-model binds the value from the HTML control to application data.
4. The **ng-change** directive is used to call the changefun().
5. Ng-pattern directive define the pattern from validator to ng model used for text based input.
6. Ng-minlength calculate the minlength from the validator to ngmodel used for text based input.
7. Ng-maxlength calculate the maxlength from validator to ng model used for text based input.
8. Span tag is used to display the content in the input field.
9. If the required directive shows an error the content is displayed like "phone is required"
- 10.If the pattern directive shows an error the content is displayed like "Pattern does not match"
- 11.If the minlength directive shows an error the content is displayed like "Phone must be max 6 char"
12. The changed directive shows "text changed" when the valid character is given in the input.
- 13.Controller is called in the function as **app.controller**.
- 14.changefun () is the ng-change directive value of the input function.

15. The value is given as mychange in the scope object.

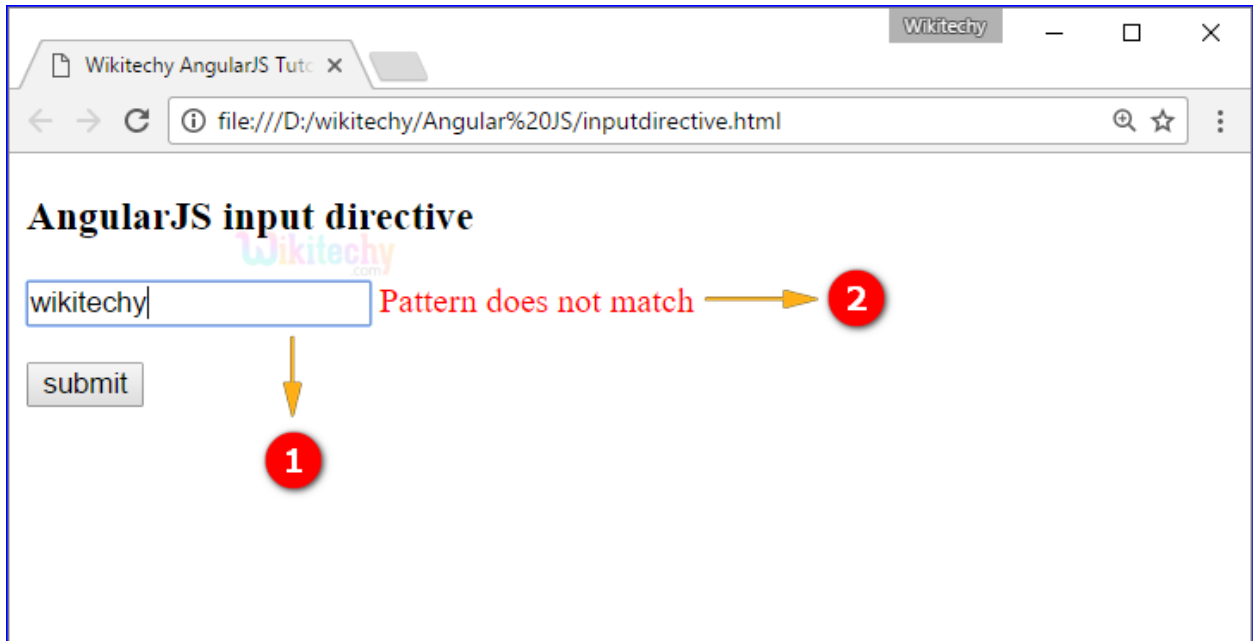
Sample Output input directive in AngularJS:



1. The output is displayed on the page load.



1. If we type the 6 char of number in the input field is consider as valid data.
2. If the data is valid then the "text changed" is displayed in the output.



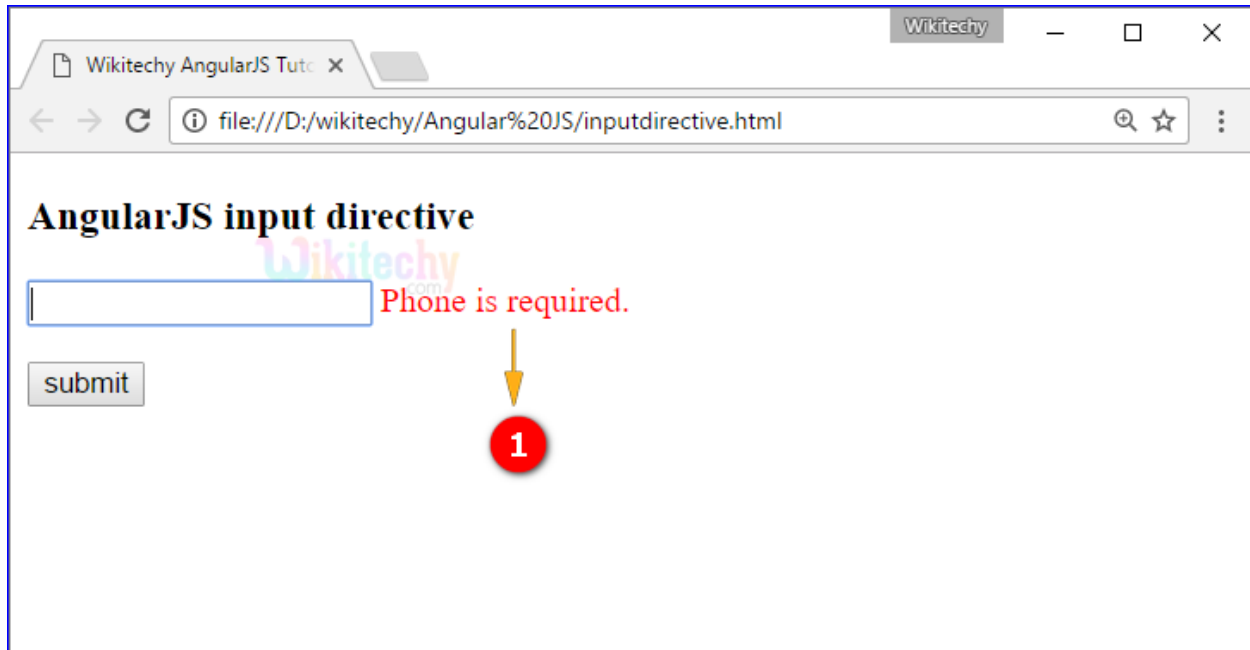
1. If we type the letter in the input field is, consider as invalid data.
2. If the data is invalid the text is displayed in the input "Pattern does not match".



1. If we type the number with one char its shows that the input field is valid but the char of the input field 6 by the "text" Phone must be max 6 char".



1. If we type the letter in the input field is consider as invalid data.
2. If the data is invalid the text is displayed in the input "Pattern does not match phone must be max 6 char".



1. If we click the submit without input field it shows the text "Phone is required"