input[email] DIRECTIVE IN ANGULARJS

- input [email] is one of the input directive in module ng.
- To validate an email address means to check the email address is valid email or not.
- It is used to text input with email validation. If it is not a valid email address means sets the email validation error key.

Syntax for input [email] directive in AngularJs:

```
<input type="email"

         ng-model="string"

        [name="string"]

        [required ="string"]

        [ng-required="string"]

        [ng-minlength="number"]

        [ng-maxlength="number"]

        [pattern ="string"]

        [ng-pattern ="string"]

        [ng-change ="string"]>
```

Parameter Values:

Parameters	Description
ngModel	Defines angular expression to data-bind to.
name (optional)	Name of the form under which the control is available.
required(optional)	If the value is not entered then sets the **required** validation error key.
ngRequired (optional)	Sets the required attribute and required validation to the element when the ngRequired expression evaluates true. Instead of required use ngRequired when we want to data-bind to the required attribute.
ngMinlength (optional)	States the minlength validation error key if the value is shorter than minlength.
ngMaxlength (optional)	States the maxlength validation error key if the value is longer than maxlength.
pattern(optional)	It is similar to ngPattern except that the attribute value is the actual string. It contains the regular expression body that will be converted to a regular expression as in the ngPattern directive.
ngPattern (optional)	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.

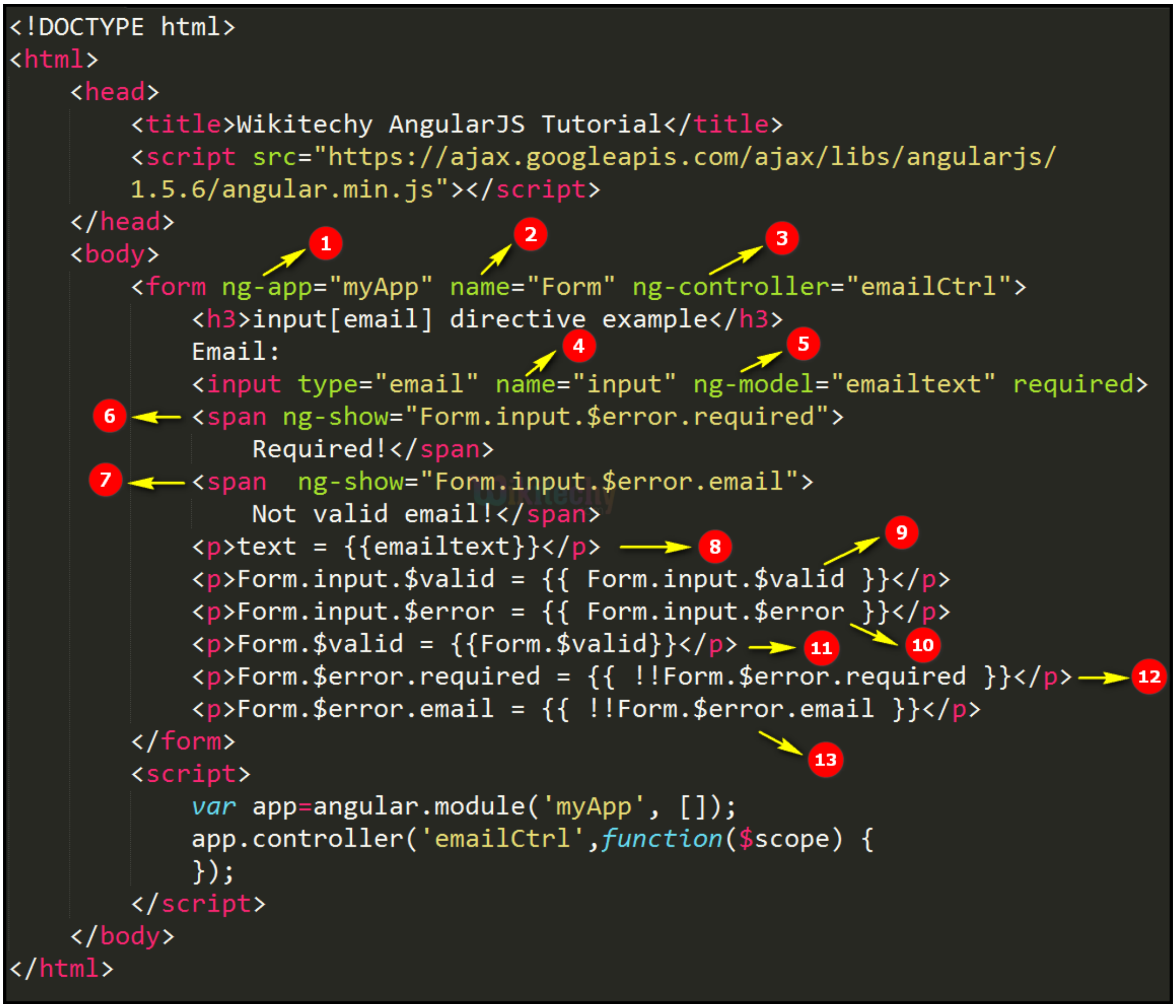
	If the expression evaluates to a string, then it will be converted to a RegExp after wrapping it in ^ and $ characters. For instance, "abc" will be converted to new RegExp ('^abc$').
ngChange (optional)	An expression of Angular to be executed when input changes due to user interaction with the input element.

.

Sample coding for input[email] directive in AngularJS

```
<!DOCTYPE html>
<html>
    <head>
        <title>Wikitechy AngularJS Tutorial</title>
        <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js">
       </script>
    </head>
    <body>
        <form ng-app="myApp" name="Form" ng-controller="emailCtrl">
            <h3>input[email] directive example</h3>
            Email:
            <input type="email" name="input" ng-model="emailtext"  required>
            <span ng-show="Form.input.$error.required">
                Required!</span>
            <span  ng-show="Form.input.$error.email">
                Not valid email!</span>
            <p>text = {{emailtext}}</p>
            <p>Form.input.$valid = {{ Form.input.$valid }}</p>
            <p>Form.input.$error = {{ Form.input.$error }}</p>
            <p>Form.$valid = {{Form.$valid}}</p>
            <p>Form.$error.required = {{ !!Form.$error.required }}</p>
            <p>Form.$error.email = {{ !!Form.$error.email }}</p>
        </form>
        <script>
            var app=angular.module('myApp', []);
            app.controller('emailCtrl',function($scope) {
            });
        </script>
    </body>
</html>
```

Code Explanation input[email] directive in AngularJS:

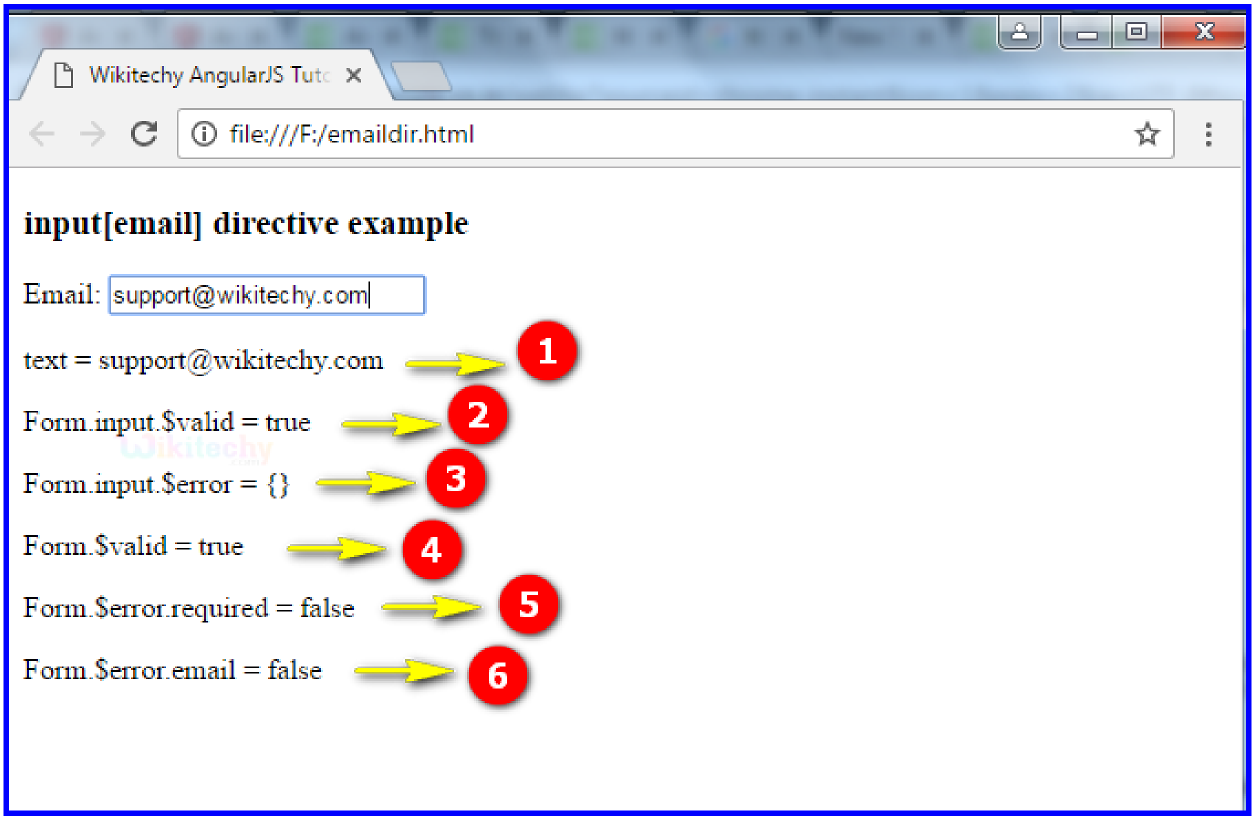


```
<!DOCTYPE html>
<html>
    <head>
        <title>Wikitechy AngularJS Tutorial</title>
        <script src="https://ajax.googleapis.com/ajax/libs/angularjs/
        1.5.6/angular.min.js"></script>
    </head>
    <body>
        <form ng-app="myApp" name="Form" ng-controller="emailCtrl">
            <h3>input[email] directive example</h3>
            Email:
            <input type="email" name="input" ng-model="emailtext" required>
            <span ng-show="Form.input.$error.required">
                Required!</span>
            <span  ng-show="Form.input.$error.email">
                Not valid email!</span>
            <p>text = {{emailtext}}</p>
            <p>Form.input.$valid = {{ Form.input.$valid }}</p>
            <p>Form.input.$error = {{ Form.input.$error }}</p>
            <p>Form.$valid = {{Form.$valid}}</p>
            <p>Form.$error.required = {{ !!Form.$error.required }}</p>
            <p>Form.$error.email = {{ !!Form.$error.email }}</p>
        </form>
        <script>
            var app=angular.module('myApp', []);
            app.controller('emailCtrl',function($scope) {
            });
        </script>
    </body>
</html>
```

1. The **ng-app** specifies the root element (**"myapp"**) to define AngularJS application.
2. Declare the form name as **"form".**

3. Ng-controller specifies the application controller in AngularJS the controller value is given as **"emailCtrl"**.
4. Here we declare the input type name as **"input".**
5. The **ng-model** bind an input field value to AngularJS application variable (**"emailtext"**).
6. Ng-show directive is used to hides HTML elements. If the required directive shows an error the content is displayed like **"Required"**
7. Ng-show directive is used to hides HTML elements. If the required directive shows an error the content is displayed like **"Not valid email"**
8. The expression (emailtext) to check whether the valid text or not. If the text is not valid and the output will not updated any text in the <p> tag. otherwise the text is updated.
9. **Form.input.$valid** to checks the valid email or not and the output will be displayed in <p> tag.
10. **Form.input.$error** to check whether the email is error or not .If the email is in error it through the exception (Like "required"="true) otherwise it is an empty curly braces ({ }).
11. **Form.$valid** is used to check whether the form is valid or not and output will be displayed in <p> tag.
12. **Form.error.$required** is used to email is required or not. If the email is required and output will be displayed as false otherwise true.
13. **Form.error.$email** is used to check whether the email is error or not and output will be displayed in the <p> tag.

Sample Output input[email] directive in AngularJS:



input[email] directive example

Email: support@wikitechy.com

text = support@wikitechy.com

Form.input.$valid = true

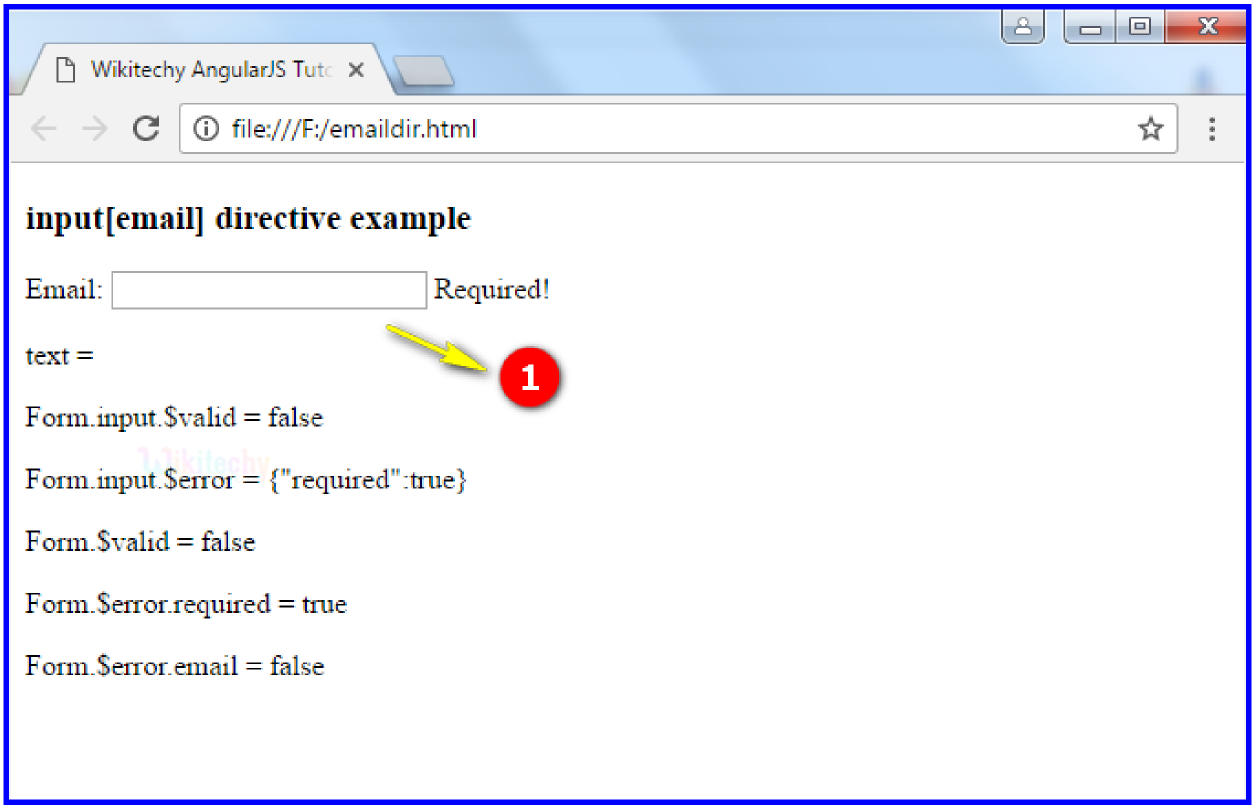
Form.input.$error = {}

Form.$valid = true

Form.$error.required = false

Form.$error.email = false

1. When user type the text in input field and the output will be displays as **text=support@wikitechy.com.**
2. The output displays true because it is consider as a valid email.
3. The output displays empty curly braces there is no error in the email.
4. The output displays true it is valid mail.
5. If the text box is not an empty so the output displays as false.
6. If the email is correct **(support@wikitechy.com)** so the output will displays false.

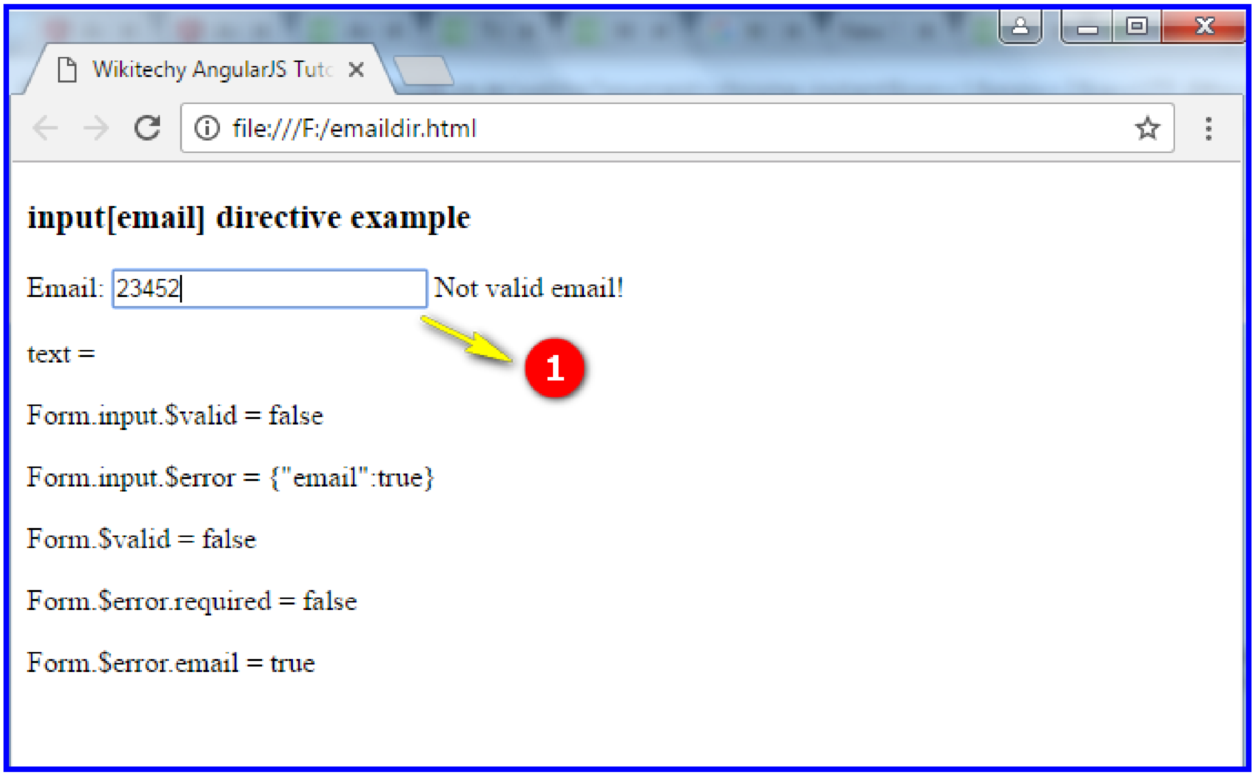


input[email] directive example

Email: Required!

```
text =

Form.input.$valid = false

Form.input.$error = {"required":true}

Form.$valid = false

Form.$error.required = true

Form.$error.email = false
```

1. If user does not type any text in the input field and output displays as "Required".



input[email] directive example

Email: 23452 Not valid email!

text =

Form.input.$valid = false

Form.input.$error = {"email":true}

Form.$valid = false

Form.$error.required = false

Form.$error.email = true

1. If the user does not type a valid mail in the text field and output displays like "Not valid email!"

Tips and Notes

- The input [email] uses a regular expression to validate email addresses. If we need exact validation, you can use **ng-pattern** or Modify the in-built validators.