

[AngularJS 教程](#)[AngularJS 简介](#)[AngularJS 表达式](#)[AngularJS 指令](#)[AngularJS 模型](#)[AngularJS Scope\(作用域\)](#)[AngularJS 控制器](#)[AngularJS 过滤器](#)[AngularJS Service](#)[AngularJS Http](#)[AngularJS Select](#)[AngularJS 表格](#)[AngularJS SQL](#)[AngularJS HTML DOM](#)[AngularJS 事件](#)[AngularJS 模块](#)[AngularJS 表单](#)[AngularJS 输入验证](#)[AngularJS API](#)[AngularJS Bootstrap](#)[← AngularJS 指令](#)[AngularJS Scope\(作用域\) →](#)

AngularJS ng-model 指令

ng-model 指令用于绑定应用程序数据到 HTML 控制器(input, select, textarea)的值。

ng-model 指令

ng-model 指令可以将输入域的值与 AngularJS 创建的变量绑定。

AngularJS 实例

```
<div ng-app="myApp" ng-controller="myCtrl">
  名字: <input ng-model="name">
</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.name = "John Doe";
});
</script>
```

[尝试一下 »](#)

双向绑定

双向绑定，在修改输入域的值时， AngularJS 属性的值也将修改：

AngularJS 实例

```
<div ng-app="myApp" ng-controller="myCtrl">
```

[AngularJS 包含](#)[AngularJS 动画](#)[AngularJS 依赖注入](#)[AngularJS 路由](#)[AngularJS 应用](#)

AngularJS 实例

[AngularJS 实例](#)

AngularJS 参考手册

[AngularJS 参考手册](#)

```
名字: <input ng-model="name">
<h1>你输入了: {{name}}</h1>
</div>
```

[尝试一下 »](#)

验证用户输入

AngularJS 实例

```
<form ng-app="" name="myForm">
  Email:
  <input type="email" name="myAddress" ng-model="text">
  <span ng-show="myForm.myAddress.$error.email">不是一个合法的邮箱地址</span>
</form>
```

[尝试一下 »](#)

以上实例中，提示信息会在 `ng-show` 属性返回 `true` 的情况下显示。

应用状态

`ng-model` 指令可以为应用数据提供状态值(`invalid`, `dirty`, `touched`, `error`):

AngularJS 实例

```
<form ng-app="" name="myForm" ng-init="myText = 'test@runoob.com'">
  Email:
  <input type="email" name="myAddress" ng-model="myText" required></p>
  <h1>状态</h1>
  {{myForm.myAddress.$valid}}
  {{myForm.myAddress.$dirty}}
  {{myForm.myAddress.$touched}}
</form>
```

[尝试一下 »](#)

CSS 类

ng-model 指令基于它们的状态为 HTML 元素提供了 CSS 类：

AngularJS 实例

```
<style>
input.ng-invalid {
    background-color: lightblue;
}
</style>
<body>

<form ng-app="" name="myForm">
    输入你的名字:
    <input name="myAddress" ng-model="text" required>
</form>
```

尝试一下 »

ng-model 指令根据表单域的状态添加/移除以下类：

- ng-empty
- ng-not-empty
- ng-touched
- ng-untouched
- ng-valid
- ng-invalid
- ng-dirty
- ng-pending
- ng-pristine

← AngularJS 指令

AngularJS Scope(作用域) →



笔记列表



ng-invalid:未通过验证的表单

ng-valid:布尔型属性，它指示表单是否通过验证。如果表单当前通过验证，他将为true

ng-dirty:布尔值属性，表示用户是否修改了表单。如果为ture，表示有修改过；false 表示修没有修改过

ng-touched:布尔值属性，表示用户是否和控件进行过交互

ng-pristine:布尔值属性，表示用户是否修改了表单。如果为ture，表示没有修改过；false表示修改过

luobu 5个月前 (07-12)



● ng-valid: 验证通过

● ng-invalid: 验证失败

● ng-valid-[key]: 由\$setValidity添加的所有验证通过的值

● ng-invalid-[key]: 由\$setValidity添加的所有验证失败的值

● ng-pristine: 控件为初始状态


● ng-dirty: 控件输入值已变更

● ng-touched: 控件已失去焦点

● ng-untouched: 控件未失去焦点

● ng-pending: 任何为满足\$asyncValidators的情况

Vivi 2个月前 (10-12)

 点我分享笔记

笔记需要是本篇文章的内容扩展！

[注册邀请码获取方式](#)



[反馈/建议](#)