## Form Validation

AngularJS offers client-side form validation.

AngularJS monitors the state of the form and input fields (input, textarea, select), and lets you notify the user about the current state.

AngularJS also holds information about whether they have been touched, or modified, or not.

You can use standard HTML5 attributes to validate input, or you can make your own validation functions.

## Form State and Input State

AngularJS is constantly updating the state of both the form and the input fields.

Input fields have the following states:

* $untouched The field has not been touched yet
* $touched The field has been touched
* $pristine The field has not been modified yet
* $dirty The field has been modified
* $invalid The field content is not valid

|  |  |
| --- | --- |
| **Property** | **Description** |
| $pristine | True if the form has not been changed (no form fields has changed), false if some fields have been changed. |
| $dirty | The reverse of $pristine - false if the form has not been changed - true if it has. |
| $valid | True if the form field (or the whole form = all form fields) is valid. False if not. |
| $invalid | The reverse of the $valid - false if the field (or all fields in the form) is valid, true if the field (or a single field in the for) is invalid. |

CSS Classes

AngularJS adds CSS classes to forms and input fields depending on their states.

### Example Explained – Custom Validator

In HTML, the new directive will be referred to by using the attribute my-directive.

In the JavaScript we start by adding a new directive named myDirective.

Remember, when naming a directive, you must use a camel case name, myDirective, but when invoking it, you must use - separated name, my-directive.

Then, return an object where you specify that we require  ngModel, which is the ngModelController.

Make a linking function which takes some arguments, where the fourth argument, mCtrl, is the ngModelController,

Then specify a function, in this case named myValidation, which takes one argument, this argument is the value of the input element.

Test if the value contains the letter "e", and set the validity of the model controller to either true or false.

At last, mCtrl.$parsers.push(myValidation); will add the myValidation function to an array of other functions, which will be executed every time the input value changes.

[Styling Classes](https://scotch.io/tutorials/angularjs-form-validation#toc-styling-classes)

ng-valid { }

.ng-invalid { }

.ng-pristine { }

.ng-dirty { }

.ng-touched { }

/\* really specific css rules applied by angular \*/

.ng-invalid-required { }

.ng-invalid-minlength { }

.ng-valid-max-length { }