

## ng-form DIRECTIVE IN ANGULARJS

- The **ng-form directive** in AngularJS used to create a nested form. Nested forms mean one form inside the another form.
- HTML does not allow the nesting of form elements.
- It specifies an inherits controls from HTML form.
  - For ex: If the validity of a sub-group of control forms needs to be determined in AngularJS.

### Syntax for ng-form Directive in AngularJS:

```
<element ng-form= [ng-form=" string"]></element>
```

### Parameter Values:

Value	Description
ng-form   name (optional)	Denotes the name of the form.



## Sample code for ng-form 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>
    <div ng-app="" >
      <form name="myForm">
        <ng-form name="Form">
          <h3>ng-form Directive example in AngularJS</h3>
          UserName:
          <input type="text" name="username" ng-model="uname"
required /> <br/> <br/>
          Password:
          <input type="text" name="password" ng-model="passname"
required /> <br/>
        </ng-form>
      </form>
      <p>form valid= <b>{{myForm.$valid}}</b> </p>
      <p>form.Username.$valid= <b>{{Form.username.$valid}}</b> </p>
      <p>form.password.$valid= <b>{{Form.password.$valid}}</b> </p>
    </div>
  </body>
</html>
```



## Using ng-form directive in AngularJS

```
<form name="myForm">
  <ng-form name="Form">
    <h3>ng-form Directive example in AngularJS</h3>
    UserName:
    <input type="text" name="username" ng-model="uname"
required /> <br/> <br/>
    Password:
    <input type="text" name="password" ng-model="passname"
required /> <br/>
  </ng-form>
</form>
```

- Here using the one form inside another form using ng-form directive in an AngularJS application.



## Code Explanation for ng-form 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>
    <div ng-app="" >
      <form name="myForm">
        <ng-form name="Form">
          <h3>ng-form Directive example in AngularJS</h3>
          UserName:
          <input type="text" name="username" ng-model="uname"
            required /><br/><br/>
          Password:
          <input type="text" name="password" ng-model="passname"
            required /><br/>
          </ng-form>
        </form>
        <p>form valid= <b>{{myForm.$valid}}</b></p>
        <p>form.Username.$valid= <b>{{Form.username.$valid}}</b></p>
        <p>form.password.$valid= <b>{{Form.password.$valid}}</b></p>
      </div>
    </body>
  </html>
```

1. The **ng-app** specifies the root element (<div>) to define AngularJS application.
2. The **ng-form** is declared by inside the <form> tag. The ng-formname is given as "Form"
3. The textbox is defined for getting name from user.
4. The textbox is defined for getting password from user.

5. `myForm.$valid` is used to check whether the form is valid or not. If the form is valid it returns true otherwise false.
6. `Form.username.$valid` function is used to check whether the username is valid or not. If the username valid is returns true. Here the form is given in the `ng-form` type so it returns the value but we are using the `<form>` tag inside the another `<form>` tag it does not return the value.
7. `Form.password.$valid` function is used to check whether the password is valid or not. If the password valid is returns true.

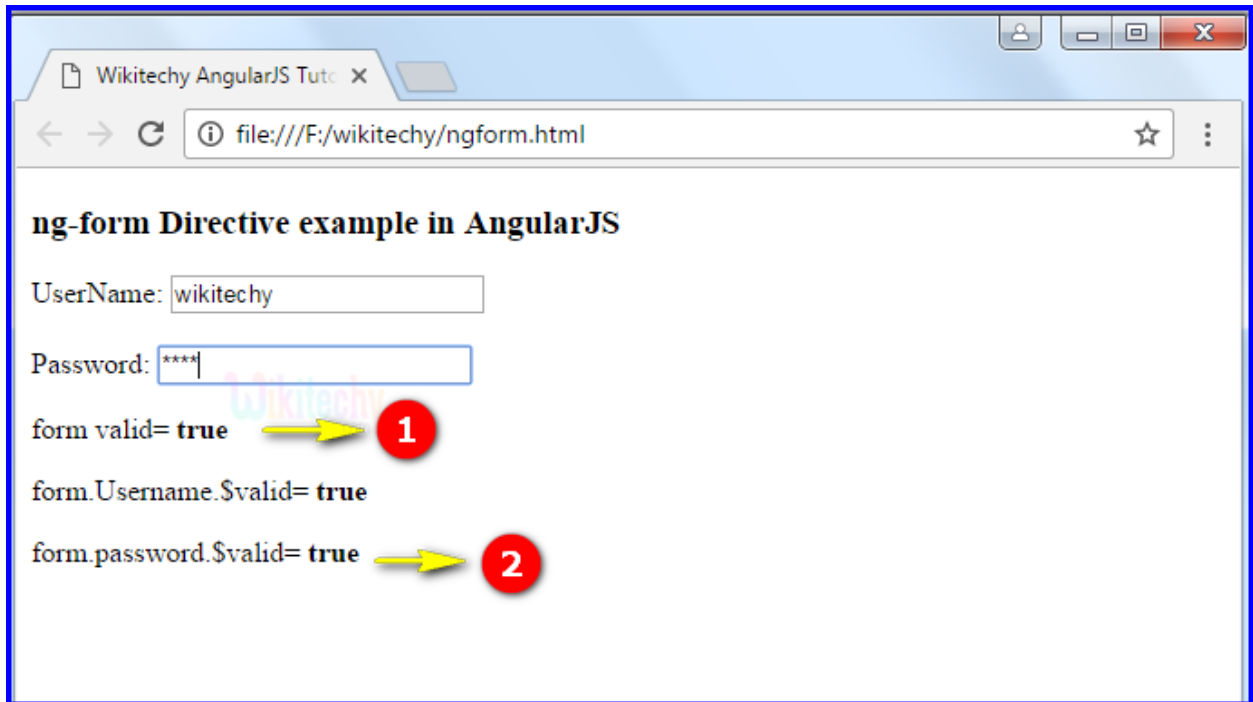
### Sample Output for ng-form Directive in AngularJS:



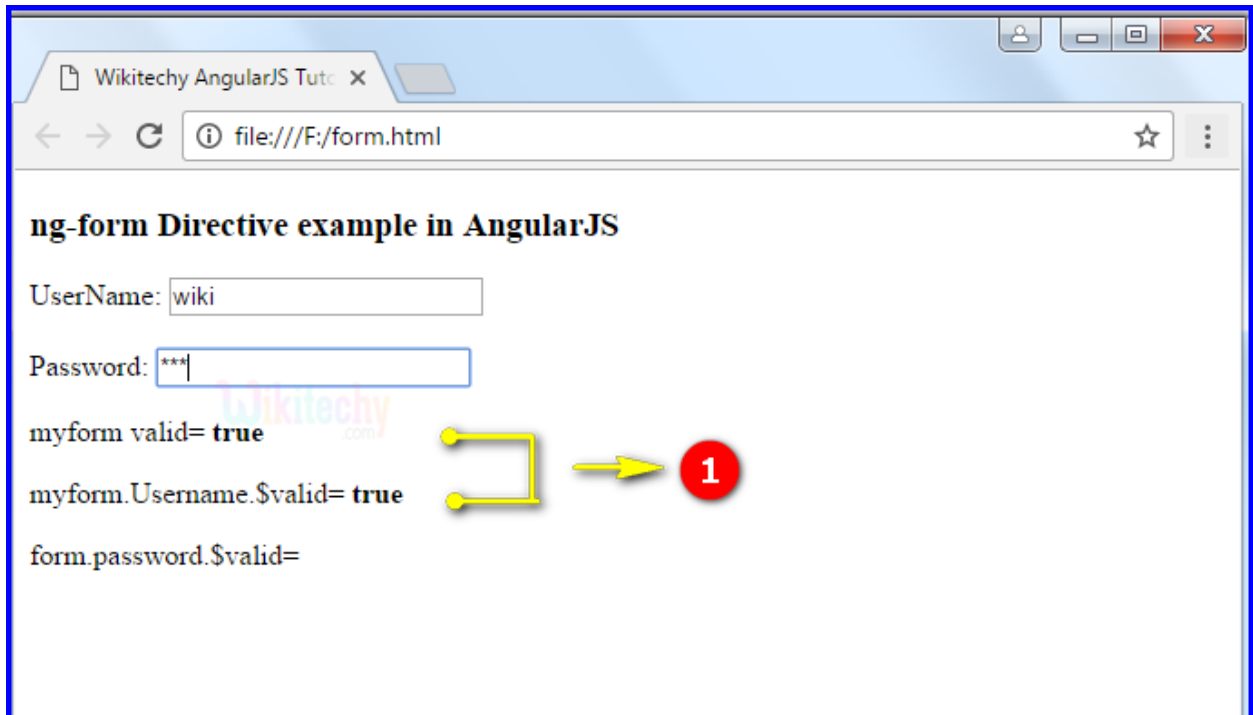
1. The form is displayed on page load.



1. When user enter the text in the input field the form (username) is valid then the output displays as true.



1. If the user enter the text in the password field then the output displays as the form is valid (like form valid=true).
2. The output displays true because the user enter the password in the input field.



1. The output displays the true value. Here using the form (form="myform") inside the form(form="form") so it returns the value only for the first form("myform") because html does not allow the nested forms.