

Introducción:

HTML es excelente para declarar documentos estáticos, pero falla cuando intentamos usarlo para declarar vistas dinámicas en aplicaciones web. AngularJS le permite ampliar el vocabulario HTML para su aplicación. El entorno resultante es extraordinariamente expresivo, legible y rápido de desarrollar.

El binding es un enlace que une unas cosas y otras. Así pues, el data binding sería "enlace de datos". En nuestro caso particular con AngularJS, el binding enlaza la información que tenemos en el "scope" con lo que mostramos en el HTML. Esto se produce en dos sentidos:

Desarrollo:

El siguiente código es empleado en la practica:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Ej de AngularJS</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.8/angular.min.js"></script>
  <script type="text/javascript">
    var app = angular.module('angulartwobindapp', []);
    app.controller('angulartwobindingCtrl', function($scope) {
      $scope.name = 'Probando One-wayBinding';
    });
  </script>
</head>

<body ng-app="angulartwobindapp">
  <div ng-controller="angulartwobindingCtrl">
    Enter Name : <input type="text" ng-model="name" style="width:250px" />
    <p>Message: {{ name }}</p>
    Nombre: <input type="text" ng-model="n1" style="width: 250px"/> <br>
  </div>
</body>
</html>
```

Para obtener el siguiente resultado:

Enter Name :

Message: Probando One-wayBinding

Posteriormente se agregan líneas para poder capturar el nombre completo de una persona de la siguiente manera:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Ej de AngularJS</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.8/angular.min.js"></script>
  <script type="text/javascript">
    var app = angular.module('angulartwobindapp', []);
    app.controller('angulartwobindingCtrl', function($scope) {
      $scope.name = 'Probando One-wayBinding';
    });
  </script>
</head>

<body ng-app="angulartwobindapp">
  <div ng-controller="angulartwobindingCtrl">
    Enter Name : <input type="text" ng-model="name" style="width:250px" />
    <p>Message: {{ name }}</p>
    Nombre: <input type="text" ng-model="nombre" style="width: 250px"/> <br>
    Apellido Materno: <input type="text" ng-model="apellidoM" style="width: 250px"/> <br>
    Apellido Paterno: <input type="text" ng-model="apellidoP" style="width: 250px"/> <br>
    <p>Bienvenido! {{nombre}} {{apellidoM}} {{apellidoP}}</p>
  </div>
</body>

</html>

```

Obteniendo lo siguiente:

Enter Name :

Message: Probando One-wayBinding

Nombre:

Apellido Materno:

Apellido Paterno:

Bienvenido!

Conclusión:

Emplear el binding permite al desarrollador recuperar información dentro de un input para poder manipularla posteriormente de una manera más sencilla, sin necesidad de utilizar procesos complicados.