

ANGULAR JS

Angular é uma biblioteca criada em Java Script com o objetivo de simplificar o desenvolvimento de aplicações com formulários e validações. Como o Angular hoje é suportado pelo google, a maneira mais simples de utiliza-lo é acrescentando a seguinte linha a seus códigos:

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
```

Angular acrescenta as seguintes TAGS HTML para serem usadas:

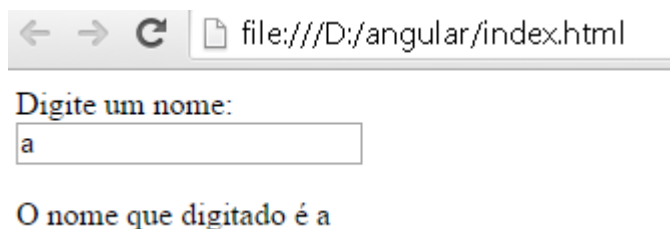
ng – app – aplicação angular

ng – model – controla campos de formulário

ng – bind – controla a exibição de dados em HTML

```
index.html
1 <html>
2   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
3   <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
4   <body>
5
6   <div ng-app="">
7     <p>Digite um nome:<br>
8     <input type="text" ng-model="nome"></p>
9     <p> O nome que digitado é <label ng-bind="nome"></p>
10  </div>
11
12 </body>
13 </html>
```

Tem como resultado visual:



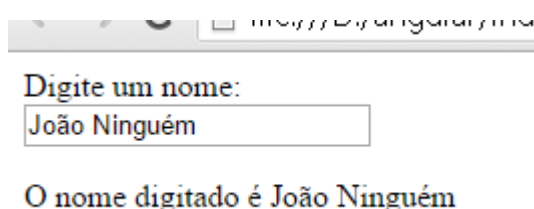
← → ↻ file:///D:/angular/index.html

Digite um nome:

O nome que digitado é a

Podemos sempre associar um campo bind a um model, assim é possível realizar tarefas como validação e exibição e dados em qualquer parte de nosso código HTML. Vamos imaginar que a caixa de nome tenha um nome padrão por exemplo, neste caso usaremos “**ng-init**” para preencher uma valor inicial para o campo. Vamos modificar o código para que fique como abaixo:

```
<div ng-app="" ng-init="nome='João Ninguém'">
  <p>Digite um nome:<br>
  <input type="text" ng-model="nome"></p>
  <p> O nome digitado é <label ng-bind="nome"></p>
</div>
```



← → ↻ file:///D:/angular/index.html

Digite um nome:

O nome digitado é João Ninguém

É possível realizar cálculos com angular, um exemplo:

```
<html>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
<body>

<div ng-app="">
  <p>Digite a:<br>
  <input type="number" ng-model="a"></p>
  <p>Digite b:<br>
  <input type="number" ng-model="b"></p>
  <p> {{a}} + {{b}} = {{a+b}}</p>
</div>

</body>
</html>
```

Resultado:

Digite a:

Digite b:

78 + 4 = 82

Angular permite também separar o controle do código de sua aparência.

```
<div ng-app="minhaAplicacao" ng-controller="somador" ng-init="a=0;b=0">
  <p>Digite a:<br>
  <input type="number" ng-model="a"></p>
  <p>Digite b:<br>
  <input type="number" ng-model="b"></p>
  <p> {{a}} + {{b}} = {{soma()}}</p>
</div>

<script>
var app = angular.module('minhaAplicacao', []);
app.controller('somador', function($scope) {
  $scope.soma= function(){
    return $scope.a + $scope.b;
  };
});
</script>
```

Resultado visual:

Digite a:

Digite b:

$$9 + 67 = 76$$

Vamos exibir um exemplo mais complexo, onde vamos escrever a tabuada de um número digitado:

```
<div ng-app="minhaAplicacao" ng-controller="tabuada" ng-init="a=0">
  <p>Digite valor:<br>
  <input type="number" ng-model="a"></p>
  <ul>
    <li ng-repeat="x in valores">
      {{ a + ' vezes ' + x + " = " + (a*x) }}
    </li>
  </ul>
</div>

<script>
var app = angular.module('minhaAplicacao', []);
app.controller('tabuada', function($scope) {
  $scope.valores = [1,2,3,4,5,6,7,8,9,10];
});
</script>
```

Resultado visual:

Digite valor:

- 9 vezes 1 = 9
- 9 vezes 2 = 18
- 9 vezes 3 = 27
- 9 vezes 4 = 36
- 9 vezes 5 = 45
- 9 vezes 6 = 54
- 9 vezes 7 = 63
- 9 vezes 8 = 72
- 9 vezes 9 = 81
- 9 vezes 10 = 90