



# JavaScript

# Funções

# Funções

```
function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
}
```

Está na memória assim que  
o programa carrega

# Funções

*construindo funções em tempo de execução*

```
function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
}
```

*Está na memória assim que  
o programa carrega*

```
var soma =
```

# Funções

construindo funções em tempo de execução

```
function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
}
```

```
var soma = function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
};
```

Está na memória assim que  
o programa carrega


A função foi atribuída  
à variável soma

# Funções

construindo funções em tempo de execução

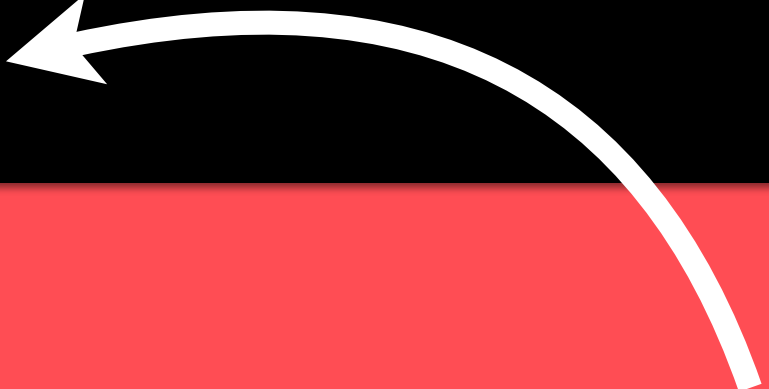
```
function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
}
```

Está na memória assim que  
o programa carrega



```
var soma = function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
};
```

É preciso de um ponto e  
vírgula no final da sentença



# Funções

construindo funções em tempo de execução

```
function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
}
```

Está na memória assim que  
o programa carrega

```
var soma = function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
};
```

Agora a função carrega  
apenas quando este código  
for alcançado



# Funções

construindo funções em tempo de execução

```
function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
}
```

Está na memória assim que  
o programa carrega

```
var soma = function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
};
```

Agora a função carrega  
apenas quando este código  
for alcançado

```
soma(3,4);
```



25

# Funções

construindo funções em tempo de execução

```
function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
}
```

```
var soma = function somaDosQuadrados(x, y){  
  return Math.pow(x,2) + Math.pow(y,2);  
};
```

O nome da função é opcional,  
já que estamos usando uma  
variável que já possui um  
nome

```
soma(3,4);
```



25

← A variável precisou de parênteses e ponto e vírgula

# Funções Anônimas

*não precisam ser nomeadas já que possuem uma variável*

```
var soma = function somaDosQuadrados(x, y){  
    return Math.pow(x,2) + Math.pow(y,2);  
};
```

# Funções Anônimas

*não precisam ser nomeadas já que possuem uma variável*

```
var soma = function (x, y){  
    return Math.pow(x,2) + Math.pow(y,2);  
};
```

# Funções Anônimas

*não precisam ser nomeadas já que possuem uma variável*

```
var soma = function (x, y){  
    return Math.pow(x,2) + Math.pow(y,2);  
};
```

*Sem nome*

```
soma(3,4);
```

*Ainda precisa de parênteses e ponto e vírgula*

*25*



# Curiosidade

```
var soma = function (x, y){  
  return x + y;  
}
```

```
console.log(soma);
```



# Curiosidade

```
var soma = function (x, y){  
    return x + y;  
}
```

```
console.log(soma);
```



# 1.1 Funções como parâmetros



# Funções como parâmetros

*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

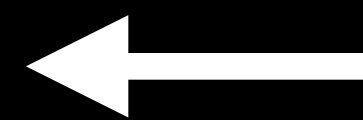
# Funções como parâmetros

*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```



*A variável mensagemFinal  
é passada como parâmetro  
para uma função existente*

# Funções como parâmetros

*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```

```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```

← A variável *mensagemFinal* é passada como parâmetro para uma função existente

# Funções como parâmetros

*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```



```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```

```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```

# Funções como parâmetros


*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```

```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```



```
function encerrarJogo(      ) {  
    ...  
    mensagem();  
    ...  
}
```

# Funções como parâmetros

*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```



```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```

```
function encerrarJogo(mensagemFinal) {  
    ...  
    mensagem();  
    ...  
}
```

# Funções como parâmetros

*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```



```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```

```
function encerrarJogo(mensagemFinal) {  
    ...  
    mensagemFinal();  
    ...  
}
```



# Funções como parâmetros

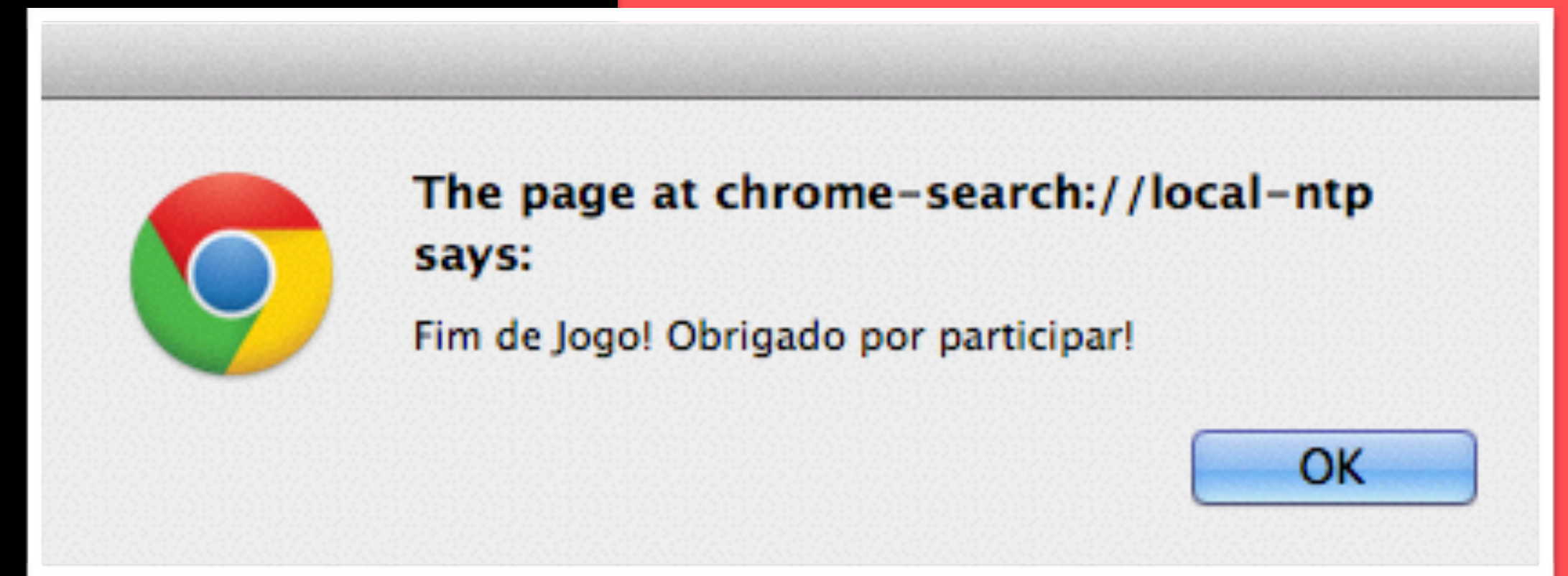
*uma variável que armazena uma função pode ser passada para outras funções*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```

```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```



```
function encerrarJogo(mensagemFinal) {  
    ...  
    mensagemFinal();  
    ...  
}
```



# E se houvesse várias mensagens?

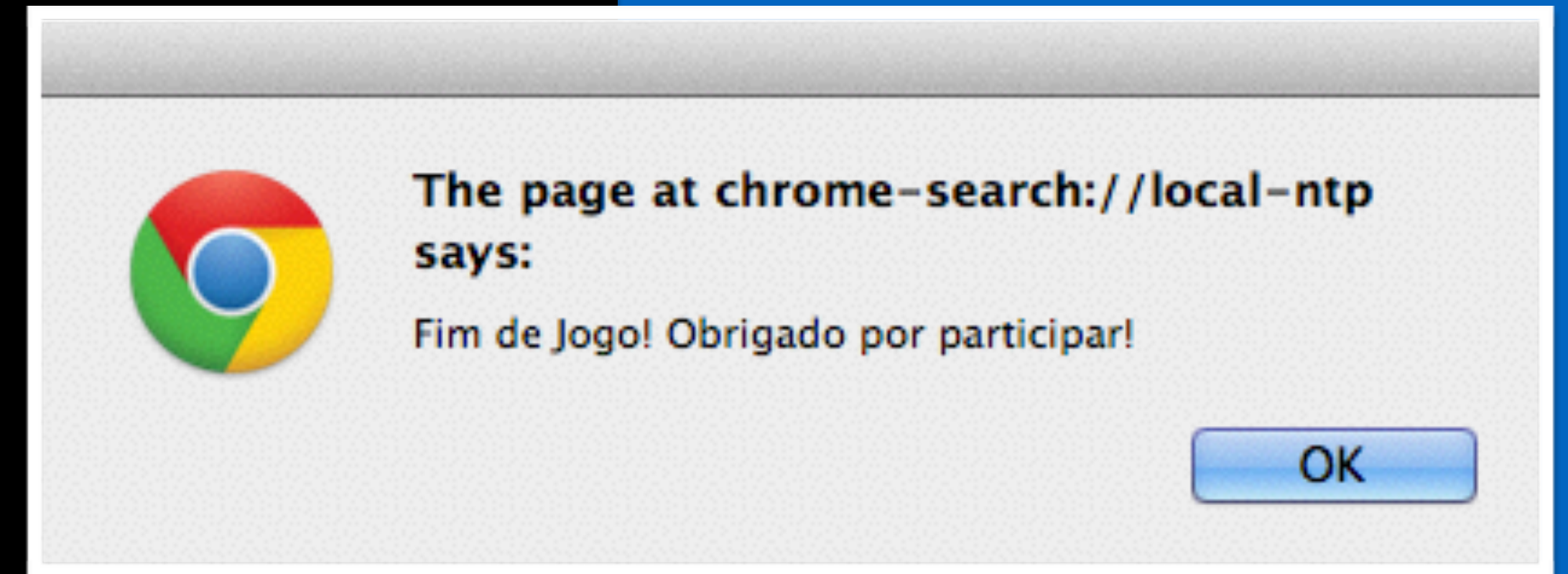
*É possível ter flexibilidade com essa estrutura e escolher uma mensagem específica*

```
var mensagemFinal = function() {  
    alert("Fim de Jogo! Obrigado por participar!");  
}
```

...

```
encerrarJogo(mensagemFinal);
```

```
function encerrarJogo(mensagem) {  
    ...  
    mensagem();  
    ...  
}
```



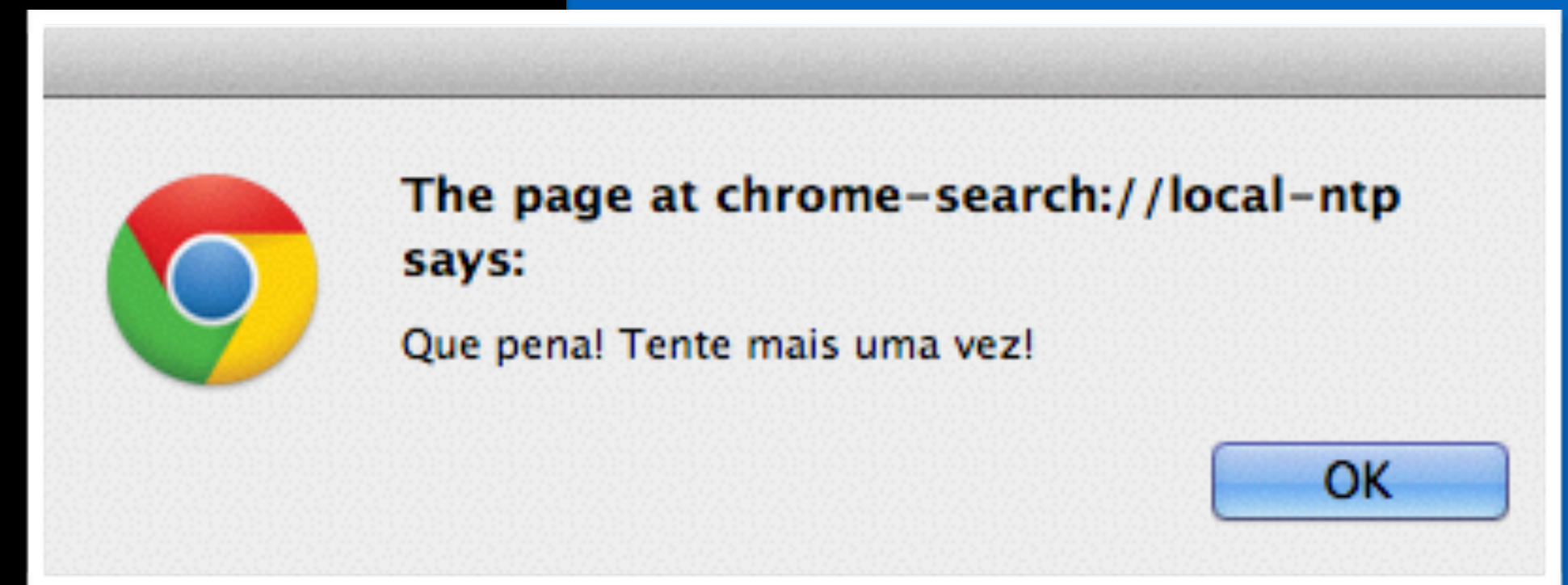
```
function encerrarJogo(mensagemFinal) {  
    ...  
    mensagemFinal();  
    ...  
}
```

# E se houvesse várias mensagens?

*É possível ter flexibilidade com essa estrutura e escolher uma mensagem específica*

```
function encerrarJogo(mensagem) {  
    mensagem();  
}  
  
var mensagemFinal;  
  
if (vencedor) {  
    mensagemFinal = function() {  
        alert("Você venceu! Batata frita!");  
    };  
} else {  
    mensagemFinal = function() {  
        alert("Que pena! Tente mais uma vez!");  
    };  
}  
  
encerrarJogo(mensagemFinal);
```

```
var vencedor = false;
```

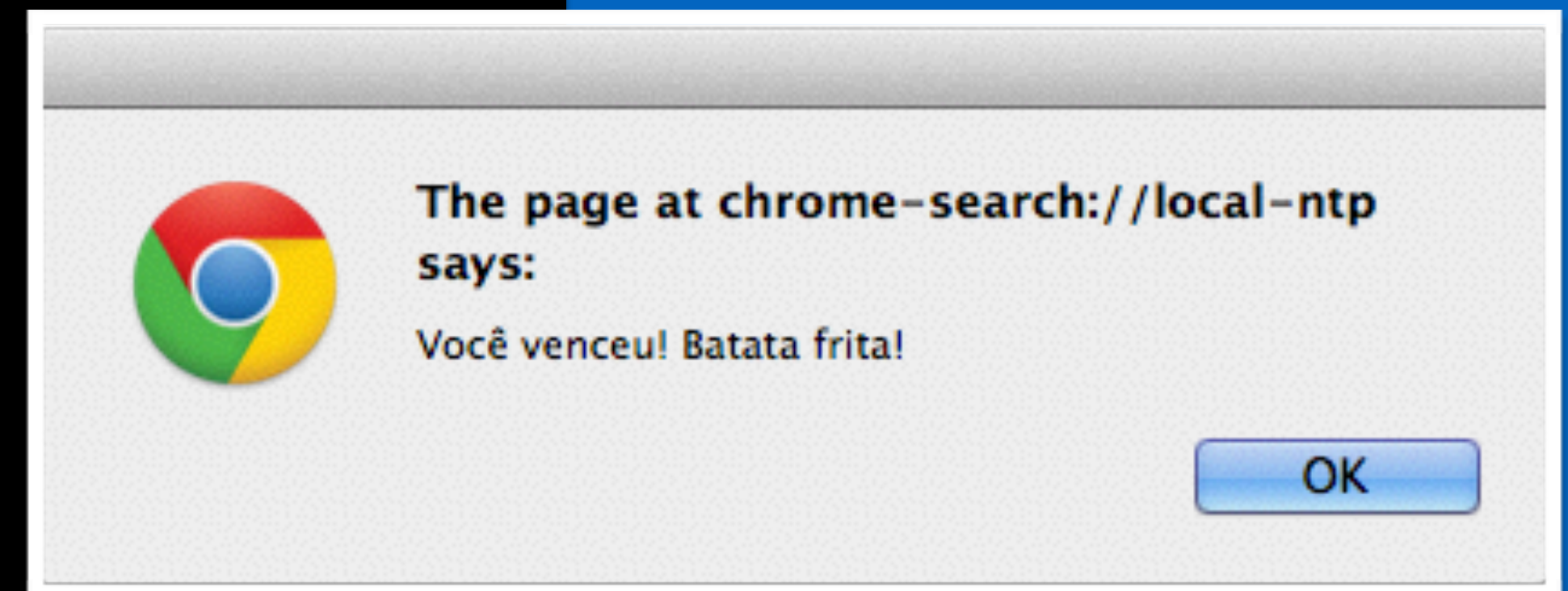


# E se houvesse várias mensagens?

*É possível ter flexibilidade com essa estrutura e escolher uma mensagem específica*

```
function encerrarJogo(mensagem) {  
    mensagem();  
}  
  
var mensagemFinal;  
  
if (vencedor) {  
    mensagemFinal = function() {  
        alert("Você venceu! Batata frita!");  
    };  
} else {  
    mensagemFinal = function() {  
        alert("Que pena! Tente mais uma vez!");  
    };  
}  
  
encerrarJogo(mensagemFinal);
```

```
var vencedor = true;
```



# Utilizando funções arrays e map

*uma variável que armazena uma função pode ser passada para outras funções*

```
var numeros = [2, 4, 5, 6, 7, 8, 9];
```

*o método **map** recebe uma função e retorna um array processado pela função*

```
var resultados = numeros.map( - outra função - );
```

2	4	5	6	7	8	9
---	---	---	---	---	---	---

*- outra função -*

outro valor	outro valor	outro valor	outro valor	outro valor	outro valor	outro valor
-------------	-------------	-------------	-------------	-------------	-------------	-------------



# Utilizando funções arrays e map

*Map funciona com um laço que executa sobre cada índice do array*

```
var numeros = [2, 4, 5, 6, 7, 8, 9];
```

```
var resultados = numbers.map(    - outra função -    );
```

```
var resultados = [];  
for (var i = 0; i < numeros.length; i++) {  
    resultados[i] = algumaFuncao(numeros[i]);  
}
```

o método **map** encapsula este processamento e o resume em uma linha de código

# Utilizando funções arrays e map

*Map funciona com um laço que executa sobre cada índice do array*

```
var numeros = [2, 4, 5, 6, 7, 8, 9];
```

```
var resultados = numeros.map(function(celula) {  
    return celula * 2;  
});
```

2	4	5	6	7	8	9
---	---	---	---	---	---	---

– função –

4	8	10	12	14	16	18
---	---	----	----	----	----	----

# Introdução

# AngularJS

*Framework JS*

*para o lado do cliente*

*capaz de adicionar*

*interatividade ao HTML*

Boa escolha para sites dinâmicos

Ajuda a organizar o JS

Se comunica com jQuery

Possibilita a criação de sites interativos

Fácil de testar



# Arquitetura MVC

# Cross-Browser



**Diretivas** Anotações HTML que disparam comportamentos JS

**Módulos** Onde estão localizados os componentes

**Controles** Onde se adicionar comportamento à aplicação

**Expressões** A forma de exibir os valores nas páginas

# Diretivas

# Directives

*Uma Directive é um marcador em um tag HTML, como um atributo, que faz a ponte entre o HTML e um código JS*

```
<!DOCTYPE html>
<html>
  <body>
    ...
  </body>
</html>
```

```
function HelloAngular() {
  alert("Hello from Angular, World");
}
```

# Directives

```
<!DOCTYPE html>
<html>
  <body ng-controller="HelloAngular"    >
    ...
  </body>
</html>
```

```
function HelloAngular() {
  alert("Hello from Angular, World");
}
```

# Utilizando AngularJS

# Utilizando AngularJS

1

download AngularJS

2

carregue-o em seu documento HTML

```
<script src="angular.min.js"></script>
```

3

carregue seus componentes angularjs

```
<script src="app.js"></script>
```

4

utilizaremos também Bootstrap

```
<link href="bootstrap.min.css" ... />
```





# Iniciando com AngularJS

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>
  <body>
    <script src="js/angular.min.js"></script>
    <script src="js/app.js"></script>
  </body>
</html>
```

# Módulos

# Módulos

*Onde são escritos partes de uma aplicação AngularJS*

Tornam o código mais fácil de manter e de entender

Também serve para comunicar as dependências da aplicação

Módulo podem depender de outros módulos

# Criando o primeiro Módulo

```
var app = angular.module('livraria', [ ]);
```

AngularJS



Nome da  
Aplicação

Dependências

# Criando o primeiro Módulo

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>
  <body>
    <script src="js/angular.min.js"></script>
    <script src="js/app.js"></script>
  </body>
</html>
```

```
var app = angular.module('livraria', [ ]);  app.js
```

# Criando o primeiro Módulo

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>
  <body    ng-app="livraria" >
    <script src="js/angular.min.js"></script>
    <script src="js/app.js"></script>
  </body>
</html>
```

```
var app = angular.module('livraria', [ ]);  app.js
```

# Expressions

# Expressions

*Permite inserir valores dinâmicos no código HTML*

```
<p>  
    0 valor de 1 + 1 = {{1 + 1}}  
</p>
```

```
<p>  
    0 valor de 1 + 1 = 2  
</p>
```

```
<p>  
    {{"Alô," + " mamãe"}}  
</p>
```

```
<p>  
    Alô, mamãe  
</p>
```



# Controles e Dados

# Utilizando dados

```
var livro = {  
  titulo: 'O Senhor dos Aneis',  
  preco: 40.0,  
  descricao: 'Descricao 1'  
}
```



Objeto JS

# Controllers

*Controlam o comportamento da aplicação em Angular, através da definição de funções e valores*

*Encapsule tudo em uma closure*

*Críe um controller*

```
(function() {  
  var app = angular.module('livraria', []);  
  app.controller('LojaControle', function() {  
  
  });  
  
})();
```

```
var livro = {  
  titulo: 'O Senhor dos Aneis',  
  preco: 40.0,  
  descricao: 'Descricao 1'  
}
```

# Controllers

Controlam o comportamento da aplicação em Angular, através da definição de funções e valores

*Encapsule tudo em uma closure*

*Críe um controller*

*Cría uma propriedade no controle para receber o valor do livro*

```
(function() {  
  var app = angular.module('livraria', []);  
  app.controller('LojaControle', function() {  
    this.produto = livro;  
  });  
  var livro = {  
    titulo: 'O Senhor dos Aneis',  
    preco: 40.0,  
    descricao: 'Descricao 1'  
  }  
})();
```

# O HTML

```
<!DOCTYPE html>
<html ng-app="livraria">
  <head>
    <meta charset="utf-8"/>
    <link rel="stylesheet" href="css/bootstrap.min.css"/>
  </head>
  <body>
    <div>
      <h1>Nome do Produto</h1>
      <h2>R$ preco</h2>
      <p>Descrição do Produto</p>
    </div>

    <script src="js/angular.min.js"></script>
    <script src="js/app.js"></script>
  </body>
</html>
```

# O HTML

```
<body ng-controller="LojaControle as loja">
  <div>
    <h1>Nome do Produto</h1>
    <h2>R$ preco</h2>
    <p>Descrição do Produto</p>
  </div>

  <script src="js/angular.min.js"></script>
  <script src="js/app.js"></script>
</body>
</html>
```

# O HTML

```
<body ng-controller="LojaControle as loja">
  <div>
    <h1>{{loja.produto.titulo}}</h1>
    <h2>R$ {{loja.produto.preco}}</h2>
    <p>{{loja.produto.descricao}}</p>
  </div>

  <script src="js/angular.min.js"></script>
  <script src="js/app.js"></script>
</body>
</html>
```

## O Senhor dos Aneis

Descricao 1

**R\$ 40**

# Exibição Condicional



# Livraria

## O Senhor dos Aneis

Descricao 1

Adicionar ao Carrinho

**R\$ 40**

Exibição de um botão  
condicionado a existir  
quantidade em estoque

# Diretiva ng-show

# Adequando os dados à necessidade

```
var livro = {  
  titulo: 'O Senhor dos Aneis',  
  preco: 40.0,  
  descricao: 'Descricao 1'  
}
```

# Adequando os dados à necessidade

```
var livro = {  
  titulo: 'O Senhor dos Aneis',  
  preco: 40.0,  
  descricao: 'Descricao 1',  
  quantidade: 10  
}
```

# Adequando os dados à necessidade

```
var livro = {  
  titulo: 'O Senhor dos Aneis',  
  preco: 40.0,  
  descricao: 'Descricao 1',  
  quantidade: 10  
}
```

```
<body ng-controller="LojaControle as loja">  
<div>  
  <div>  
    <h1>{{loja.produto.titulo}}</h1>  
    <h2>R$ {{loja.produto.preco}}</h2>  
    <p>{{loja.produto.descricao}}</p>  
    <button ng-show="{{loja.produto.quantidade > 0}}">  
      Adicionar ao Carrinho  
    </button>  
  </div>  
</div>
```



# Um acervo de livros

*Normalmente se tem mais que um livro*

```
app.controller('LojaControle', function() {  
    this.produto = livro;  
});  
  
var livro = {  
    titulo: 'O Senhor dos Aneis',  
    preco: 40.0,  
    descricao: 'Descricao 1',  
    quantidade: 10  
}
```

# Um acervo de livros

*Normalmente se tem mais que um livro*

```
app.controller('LojaControle', function () {  
    this.acervo = livros;  
});  
  
var livros = [ Array  
    {  
        titulo: 'O Senhor dos Aneis',  
        preco: 40.0,  
        descricao: 'Descricao 1',  
        quantidade: 10  
    }, {  
        titulo: 'A Saga Star Wars',  
        preco: 80.0,  
        descricao: 'Descricao 2',  
        quantidade: 0  
    }  
]
```

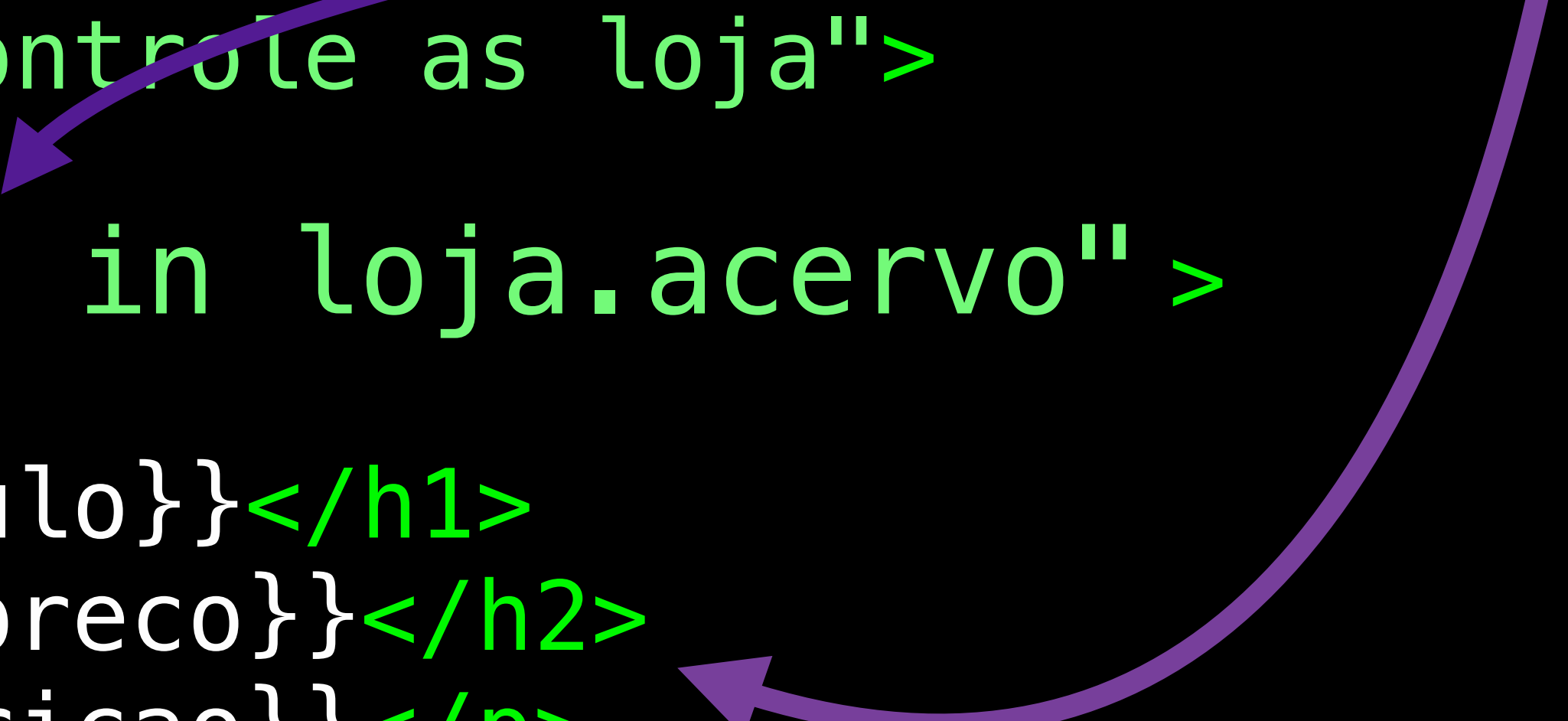
# Diretiva ng-repeat



# Aplicando o ng-repeat

Deve ser referenciado  
pela variável livro

```
<body ng-controller="LojaControle as loja">  
  <div ng-repeat="livro in loja.acervo">  
    <h1>{{loja.produto.titulo}}</h1>  
    <h2>R$ {{loja.produto.preco}}</h2>  
    <p>{{loja.produto.descricao}}</p>  
    <button ng-show="{{loja.produto.quantidade > 0}}">  
      Adicionar ao Carrinho  
    </button>  
  </div>  
</div>
```



# Aplicando o ng-repeat

```
<body ng-controller="LojaControle as loja">
```

```
  <div ng-repeat="livro in loja.acervo" >
```

```
    <h1>{{livro.titulo}}</h1>
```

```
    <h2>R$ {{livro.preco}}</h2>
```

```
    <p>{{livro.descricao}}</p>
```

```
    <button ng-show="{{livro.quantidade > 0}}">
```

Adicionar ao Carrinho

```
  </button>
```

```
</div>
```

```
</div>
```

*Repetirá a div  
para cada livro no array*

# Diretivas vistas

ng-app : anexa o Módulo da aplicação à pagina

```
<html ng-app="livraria">
```

ng-controller : anexa o função de Controle à pagina

```
<body ng-controller="LojaControle as loja">
```

ng-show/ng-hide : exhibe algo mediante a avaliação de uma Expressão

```
<h1 ng-show="algoBooleano">Condicionado</h1>
```

ng-repeat : executa um laço para cada item de um Arrau

```
<li ng-repeat="livro in loja.acervo">
```



# Utilizando Abas



## Controlador

propriedade

- aba

comportamentos

- setAba(aba)
- isSet(aba)

# AbasControle

```
app.controller('AbasControle', function() {  
    this.aba = 1;  
  
    this.setAba = function(aba) {  
        this.aba = aba;  
    };  
  
    this.isSet = function(aba) {  
        return this.aba == aba;  
    };  
});
```

# Diretiva ng-click

permite especificar um comportamento personalizado quando um elemento é clicado

# O HTML das Abas

```
<section class="tab" ng-controller="AbasControle as aba">
  <ul class="nav nav-pills">
    <li><a href >Descrição</a></li>
    <li><a href >Comentários</a></li>
  </ul>

  <div>
    aba 1  <h4>Descrição</h4>
           <blockquote>{{livro.descricao}}</blockquote>
    </div>
    <div>
    aba 2  <h4>Comentários</h4>
           <blockquote></blockquote>
    </div>
</section>
```

ng-click  
setAba(?)

ng-show  
isSet(?)

# O HTML das Abas

```
<section class="tab" ng-controller="AbasControle as aba">
  <ul class="nav nav-pills">
    <li><a href >Descrição</a></li>
    <li><a href >Comentários</a></li>
  </ul>

  <div ng-show="aba.isSet(1)">
    <h4>Descrição</h4>
    <blockquote>{{livro.descricao}}</blockquote>
  </div>
  <div ng-show="aba.isSet(2)">
    <h4>Comentários</h4>
    <blockquote></blockquote>
  </div>

</section>
```



# O HTML das Abas

```
<section class="tab" ng-controller="AbasControle as aba">
  <ul class="nav nav-pills">
    <li><a href ng-click="aba.setAba(1)">Descrição</a></li>
    <li><a href ng-click="aba.setAba(2)">Comentários</a></li>
  </ul>

  <div ng-show="aba.isSet(1)">
    <h4>Descrição</h4>
    <blockquote>{{livro.descricao}}</blockquote>
  </div>
  <div ng-show="aba.isSet(2)">
    <h4>Comentários</h4>
    <blockquote></blockquote>
  </div>

</section>
```

# Formulários

# Forms

Livraria

O Senhor dos Aneis

Descricao 1

Descricao 1

Adicionar ao Carrinho

Descricao

Comentários

R\$ 40

A Saga Star Wars

Descricao 2

Descricao 2

Descricao

Comentários

R\$ 80

# Adicionando comentarios

```
var livros = [  
  {  
    titulo: 'O Senhor dos Aneis',  
    preco: 40.0,  
    descricao: 'Descricao 1',  
    quantidade: 10,  
    comentarios : [{  
      autor: "Autor1",  
      texto: "Texto 1"  
    }, {  
      autor: "Autor2",  
      texto: "Texto 2"  
    }]  
  }  
]
```

Array

# O HTML das Abas

```
<section class="tab" ng-controller="AbasControle as aba">
  <ul class="nav nav-pills">
    <li><a href ng-click="aba.setAba(1)">Descrição</a></li>
    <li><a href ng-click="aba.setAba(2)">Comentários</a></li>
  </ul>

  <div ng-show="aba.isSet(1)">
    <h4>Descrição</h4>
    <blockquote>{{livro.descricao}}</blockquote>
  </div>
  <div ng-show="aba.isSet(2)">
    <h4>Comentários</h4>
    <blockquote></blockquote>
  </div>

</section>
```

# O HTML dos Comentários (ng-repeat)

```
<div ng-show="aba.isSet(2)">
  <h4>Comentários</h4>
  <blockquote ng-repeat="comentario in livro.comentarios">
    {{comentario.autor}} disse: {{comentario.texto}}
  </blockquote>
</div>
```

# O HTML do Formulário

```
<div ng-show="aba.isSet(2)">
  <h4>Comentários</h4>
  <blockquote ng-repeat="comentario in livro.comentarios">
    {{comentario.autor}} disse: {{comentario.texto}}
  </blockquote>
  <div>
    <form name="comentarioForm">
      <p>
        <input type="text"
          placeholder="Seu nome"/>
      </p>
      <p><textarea
        placeholder="Deixe seu comentario"></textarea>
      </p>
      <input type="submit" value="Adicionar" />
    </form>
  </div>
</div>
```

# O HTML do Formulário

```
<div ng-show="aba.isSet(2)">
  <h4>Comentários</h4>
  <blockquote ng-repeat="comentario in livro.comentarios">
    {{comentario.autor}} disse: {{comentario.texto}}
  </blockquote>
  <div>
    <form name="comentarioForm">
      <blockquote>
        {{comentario.autor}} disse: {{comentario.texto}}
      </blockquote>
      <p>
        <input type="text"
          placeholder="Seu nome"/>
      </p>
      <p><textarea
        placeholder="Deixe seu comentario"></textarea>
      </p>
      <input type="submit" value="Adicionar" />
    </form>
  </div>
</div>
```



# Diretiva ng-model

vincula o valor de um input, select, textarea a uma propriedade

# O HTML do Formulário

```
<div ng-show="aba.isSet(2)">
  <h4>Comentários</h4>
  <blockquote ng-repeat="comentario in livro.comentarios">
    {{comentario.autor}} disse: {{comentario.texto}}
  </blockquote>
  <div>
    <form name="comentarioForm">
      <blockquote>
        {{comentario.autor}} disse: {{comentario.texto}}
      </blockquote>
      <p>
        <input type="text"
          placeholder="Seu nome"/>
      </p>
      <p><textarea
        placeholder="Deixe seu comentario"></textarea>
      </p>
      <input type="submit" value="Adicionar" />
    </form>
  </div>
</div>
```

# O HTML do Formulário

```
<div ng-show="aba.isSet(2)">
  <h4>Comentários</h4>
  <blockquote ng-repeat="comentario in livro.comentarios">
    {{comentario.autor}} disse: {{comentario.texto}}
  </blockquote>
  <div>
    <form name="comentarioForm">
      <blockquote>
        {{comentario.autor}} disse: {{comentario.texto}}
      </blockquote>
      <p>
        <input type="text" ng-model="comentario.autor"
          placeholder="Seu nome"/>
      </p>
      <p><textarea ng-model="comentario.texto"
        placeholder="Deixe seu comentario"></textarea>
      </p>
      <input type="submit" value="Adicionar" />
    </form>
  </div>
</div>
```

# Submetendo valores

# Forms

**Livraria**

## O Senhor dos Aneis

Descricao 1

R\$ 40

[Descrição](#) [Comentários](#)

**Descrição**

Descricao 1

[Adicionar ao Carrinho](#)

## A Saga Star Wars

Descricao 2

R\$ 80

[Descrição](#) [Comentários](#)

**Descrição**

Descricao 2

## Controlador

propriedade

- comentario

comportamentos

- addComentario(livro)

# ComentarioControle

```
app.controller('ComentarioControle', function() {  
    this.comentario = {};  
});
```

# O HTML do Formulário

```
<div ng-show="aba.isSet(2)">
  <h4>Comentários</h4>
  <blockquote ng-repeat="comentario in livro.comentarios">
    {{comentario.autor}} disse: {{comentario.texto}}
  </blockquote>
  <div>
    <form name="comentarioForm">
      <blockquote>
        {{comentario.autor}} disse: {{comentario.texto}}
      </blockquote>
      <p>
        <input type="text" ng-model="comentario.autor"
          placeholder="Seu nome"/>
      </p>
      <p><textarea ng-model="comentario.texto"
        placeholder="Deixe seu comentario"></textarea>
      </p>
      <input type="submit" value="Adicionar" />
    </form>
  </div>
</div>
```

# O HTML do Formulário (ng-model)

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle">
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="comentario.autor"
           placeholder="Seu nome"/>
  </p>
  <p><textarea ng-model="comentario.texto"
             placeholder="Deixe seu comentario"></textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```



# O HTML do Formulário (ng-model)

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle">
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome"/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
             placeholder="Deixe seu comentario"></textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```

# Diretiva **ng-submit**

permite vincular comportamentos  
a um evento onsubmit

# O HTML do Formulário (ng-model)

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle">
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome"/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
              placeholder="Deixe seu comentario"></textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```

# O HTML do Formulário (ng-model)

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle"
      ng-submit="cmtControle.addComentario(livro)" >
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome"/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
             placeholder="Deixe seu comentario"></textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```

# ComentarioControle

```
app.controller('ComentarioControle', function() {  
    this.comentario = {};  
  
    this.addComentario = function(livro) {  
        livro.comentarios.push(this.comentario);  
    };  
});
```

# ComentarioControle

```
app.controller('ComentarioControle', function() {  
    this.comentario = {};  
  
    this.addComentario = function(livro) {  
        livro.comentarios.push(this.comentario);  
        this.comentario = {};  
    };  
});
```

## Livraria

### O Senhor dos Aneis

Descricao 1

R\$ 40

[Descrição](#)

[Comentários](#)

#### Descrição

Descricao 1

Adicionar ao Carrinho

### A Saga Star Wars

Descricao 2

R\$ 80

[Descrição](#)

[Comentários](#)

#### Descrição

Descricao 2

# Validações



# Desabilitando a validação padrão

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle"
      ng-submit="cmtControle.addComentario(livro)" >
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome"/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
              placeholder="Deixe seu comentario"></textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```

# Desabilitando a validação padrão

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle"
      ng-submit="cmtControle.addComentario(livro)" novalidate>
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome"/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
             placeholder="Deixe seu comentario"></textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```

# Desabilitando a validação padrão

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle"
      ng-submit="cmtControle.addComentario(livro)" novalidate>
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome" required/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
             placeholder="Deixe seu comentario" required> </textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```

# \$valid

propriedade form no Angular  
que indica se o formulário é válido

# Desabilitando a validação padrão

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle"
      ng-submit="cmtControle.addComentario(livro)" novalidate>
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome" required/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
             placeholder="Deixe seu comentario" required> </textarea>
  </p>
  <input type="submit" value="Adicionar" />
</form>
```

# Desabilitando a validação padrão

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle"
      ng-submit="cmtControle.addComentario(livro)" novalidate>
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome" required/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
             placeholder="Deixe seu comentario" required> </textarea>
  </p>
  <div>0 formulario é ${{comentarioForm.$valid}}</div>
  <input type="submit" value="Adicionar" />
</form>
```

# Desabilitando a validação padrão

```
<form name="comentarioForm"
      ng-controller="ComentarioControle as cmtControle"
      ng-submit="cmtControle.addComentario(livro)" novalidate>
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
           placeholder="Seu nome" required/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
             placeholder="Deixe seu comentario" required> </textarea>
  </p>
  <div>0 formulario é {{{comentarioForm.$valid}}}</div>
  <input type="submit" value="Adicionar" />
</form>
```



# Desabilitando a validação padrão

```
<form name="comentarioForm"
  ng-controller="ComentarioControle as cmtControle"
  ng-submit="comentarioForm.$valid && cmtControle.addComentario(livro)"
  novalidate>
  <blockquote>...</blockquote>
  <p>
    <input type="text" ng-model="cmtControle.comentario.autor"
      placeholder="Seu nome" required/>
  </p>
  <p><textarea ng-model="cmtControle.comentario.texto"
    placeholder="Deixe seu comentario" required> </textarea>
  </p>
  <div>0 formulario é {{comentarioForm.$valid}}</div>
  <input type="submit" value="Adicionar" />
</form>
```



# Classe no Input pelo Angular

```
<input name="autor" type="email" ng-model="cmtControle.comentario.autor" required/>
```

antes de digitar

.ng-pristine  
.ng-invalid

digitado email inválido

.ng-dirty  
.ng-invalid

digitado email válido

.ng-dirty  
.ng-valid

# CSS

```
.ng-invalid.ng-dirty {  
  border-color: darkred;  
}
```

```
.ng-invalid.ng-dirty {  
  border-color: greenyellow;  
}
```

# Projeto

# Projeto

