ANGULARJS

DIRETIVAS

Leandro Costa

http://leandrocosta.github.io/curso-angularjs-diretivas

> *“AngularJS Directive is what we have close to Web Components at this time and is the most crucial and difficult piece in AngularJS.” -- Amit Gharat (@codef0rmer)*

> *“Directives are the most important components of any AngularJS application.” -- Sandeep Panda (@sandeepg33k)*

WHAT ARE DIRECTIVES?

> *At a high level, directives are **markers** on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS's HTML compiler ($compile) to attach a specified **behavior** to that DOM element or even **transform** the DOM element and its children.*

Guide to AngularJS Documentation

jQuery UI Datepicker

```
<body>
  <input type="text">

  <script>
    $(function() {
      $("input").datepicker();
    });
  </script>
</body>
```

http://jsbin.com/vukuzo

Angular Datepicker

```
<body>
  <datepicker>
    <input ng-model="date" type="text"/>
  </datepicker>

  <script>
    (function() {
      angular.module('myApp', ['720kb.datepicker']);
    })();
  </script>
</body>
```

http://jsbin.com/vulibe

```
<body>
  <input type="text">

  <script>
    $(function() {
      $("input").datepicker();
    });
  </script>
</body>
```

```
<body>
  <datepicker>
    <input ng-model="date" type="text"/>
  </datepicker>

  <script>
    (function() {
      angular.module('myApp', ['720kb.datepicker']);
    })();
  </script>
</body>
```

How I've Improved My Angular Apps by Banning ng-controller

> *“I've recently come to a realization that using standalone controllers, as with ng-controller, rarely leads to an optimal design.”*

> *“I've talked with several people who have said they don't use ng-controller at all in their apps, and just use directives instead.”*

Tero Parviainen (@teropa)

Problemas com *ng-controller*

- Herança (*the "dot rule"*)
- Dados semi-globais (*controller as*)
- Organização visual fraca

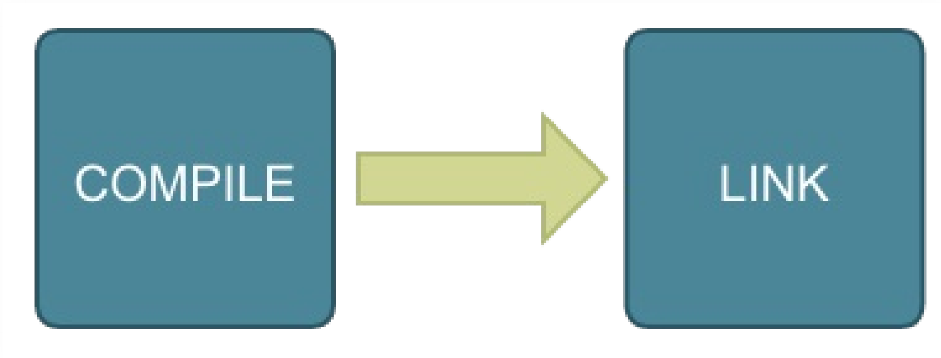
The "dot rule"

http://jsbin.com/cimoqe

http://jsbin.com/diyuru

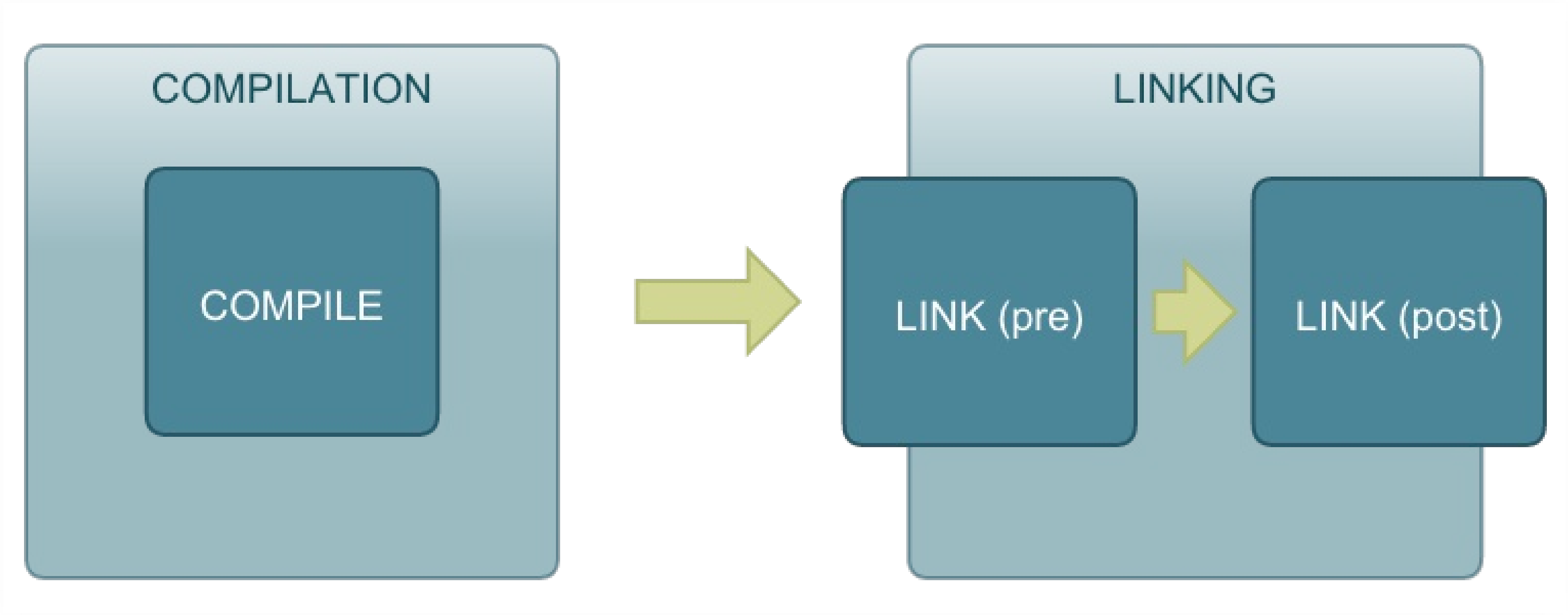
https://egghead.io/lessons/angularjs-the-dot

Directive Compilation in AngularJS – step-by-step



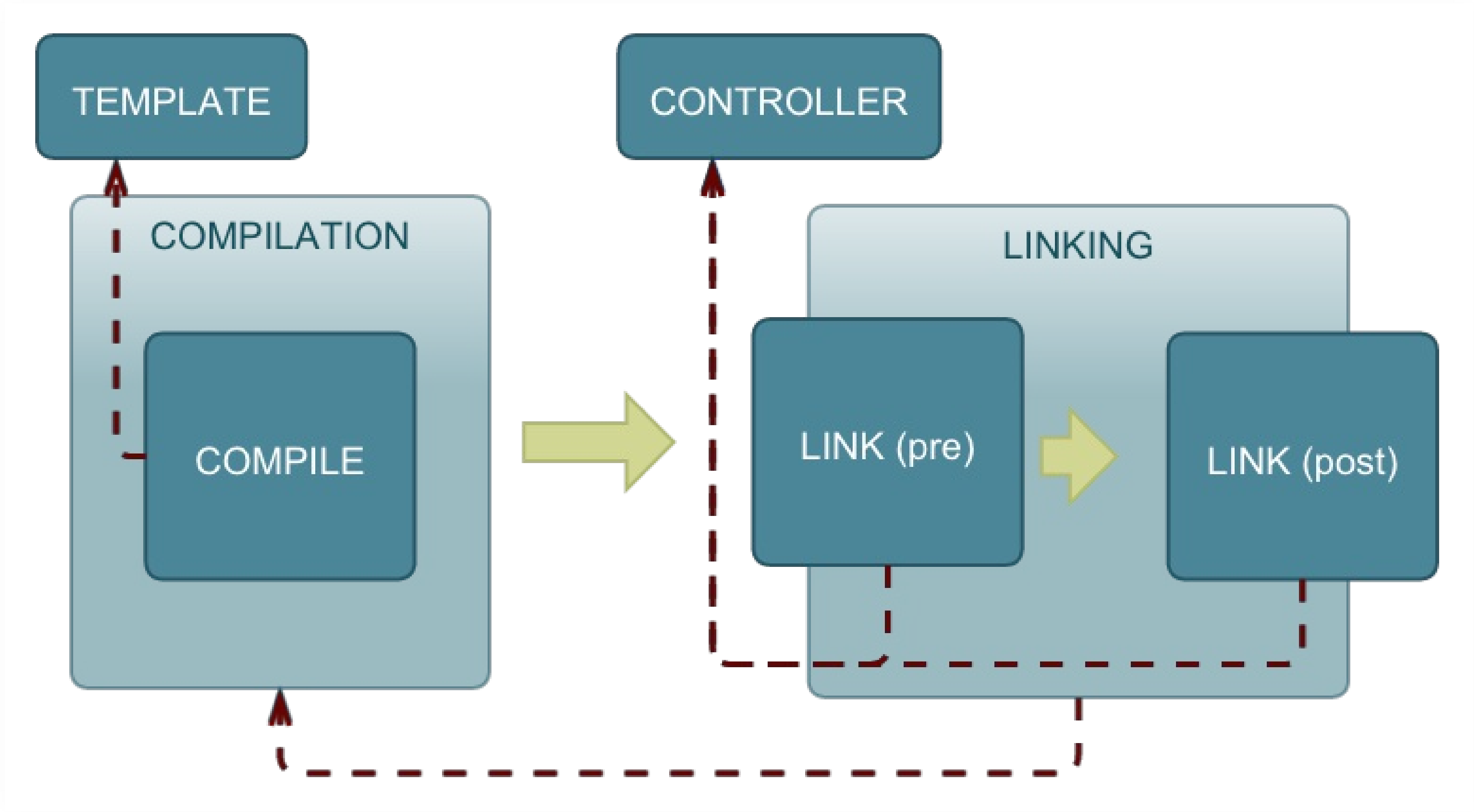
by Jakub Strojewski at triangular

Directive Compilation in AngularJS – step-by-step



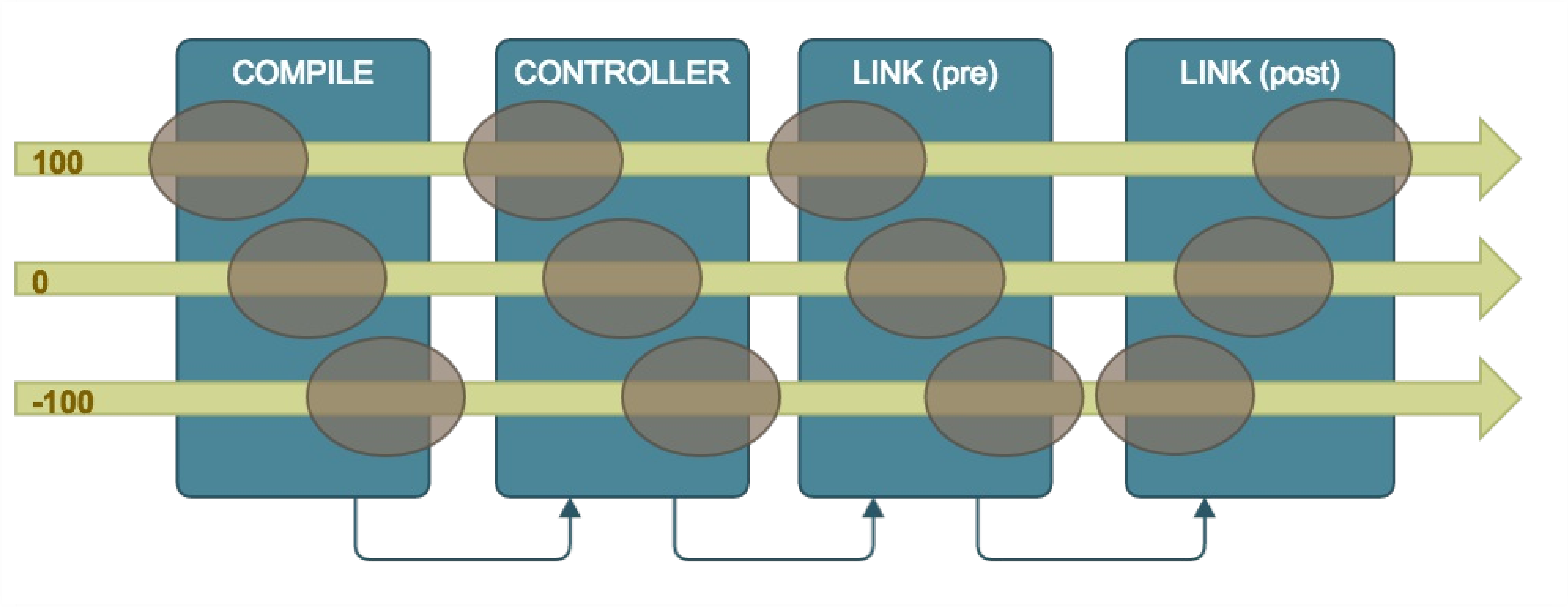
by Jakub Strojewski at triangular

Directive Compilation in AngularJS – step-by-step



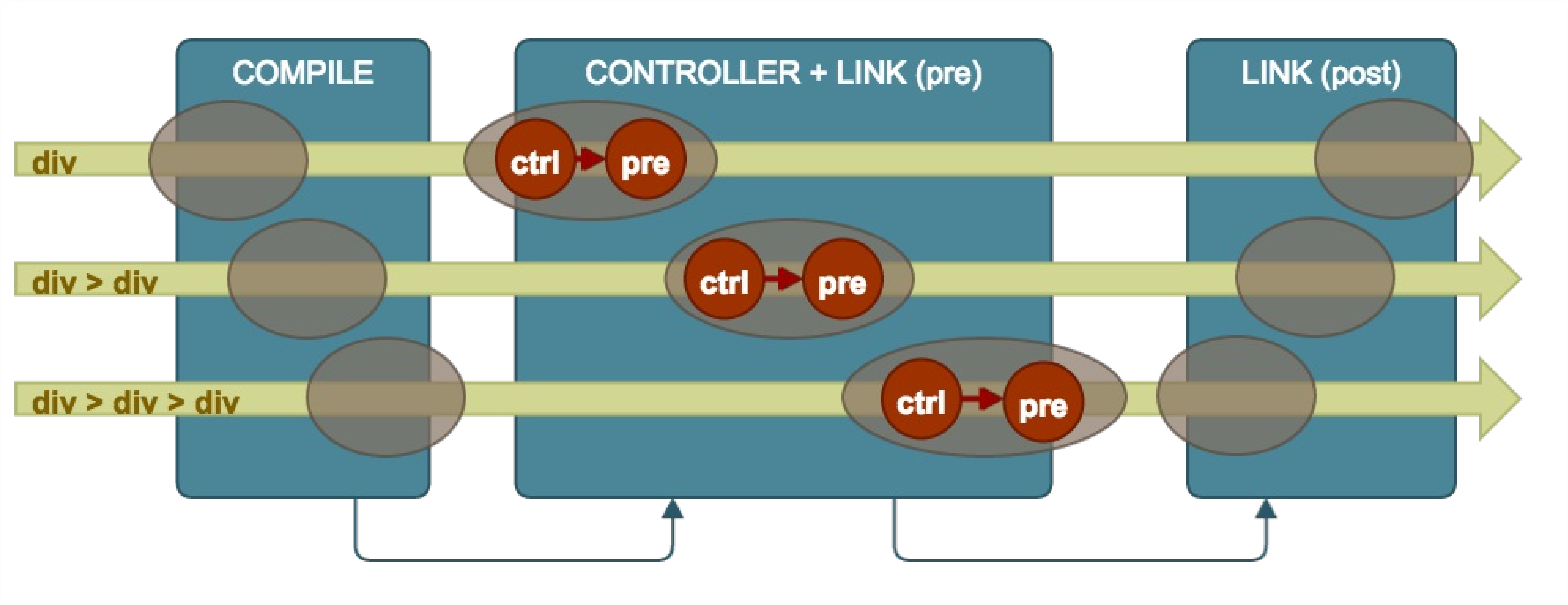
by Jakub Strojewski at triangular

Directive Compilation in AngularJS – step-by-step



by Jakub Strojewski at triangular

Directive Compilation in AngularJS – step-by-step



by Jakub Strojewski at triangular

Declaração (*return object*)

```
<body>
  <my-directive></my-directive>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function() {
          return {
            template:'<p>Olá mundo!</p>'
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/witehe

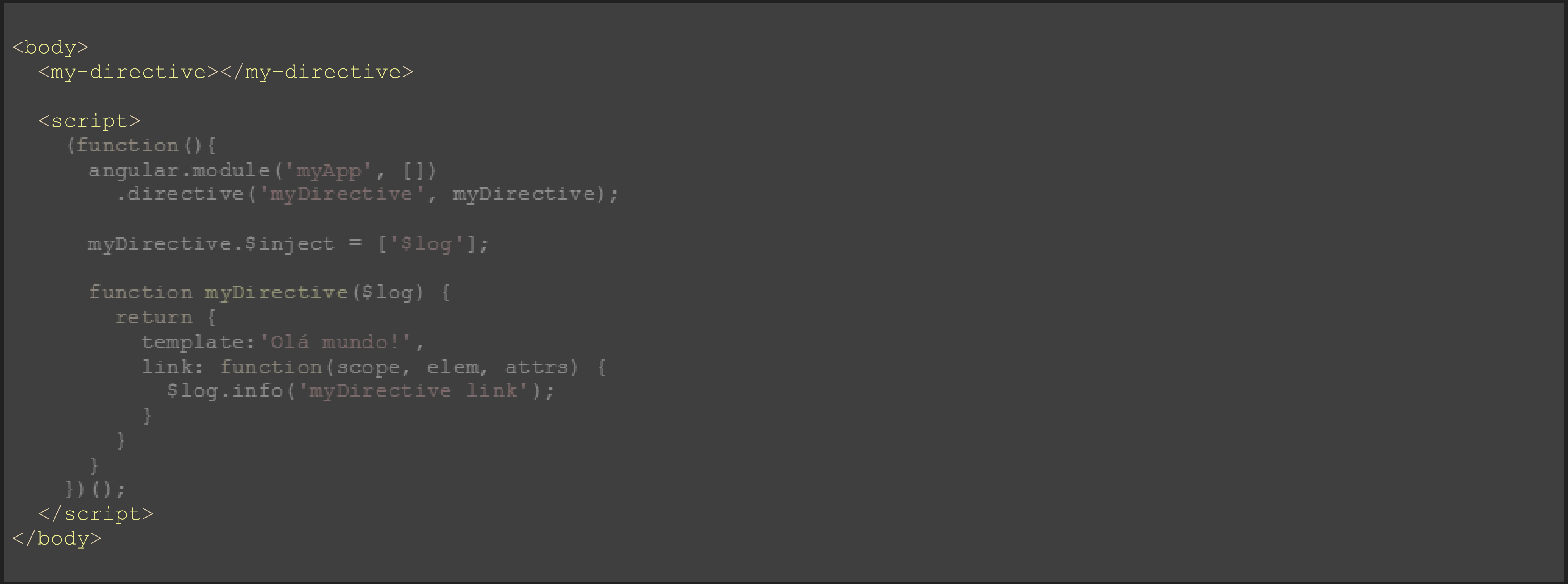
Declaração (*return function*)

```
<body>
  <my-directive></my-directive>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function($log) {
          return function(scope, elem, attrs) {
            $log.info('myDirective');
          }
        });
    })();
  </script>
</body>
```

http://jsbin.com/dudovi

Declaração (*injection function annotation*)



```
<body>
  <my-directive></my-directive>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', myDirective);

      myDirective.$inject = ['$log'];

      function myDirective($log) {
        return {
          template:'Olá mundo!',
          link: function(scope, elem, attrs) {
            $log.info('myDirective link');
          }
        }
      }
    })();
  </script>
</body>
```

http://jsbin.com/jopaga

Directive Definition Object

```
myModule.directive('directiveName', function factory(injectables) {
  return {
    priority: 0,
    template: '<div></div>' /* or function(tElement, tAttrs) { ... } */, // or
    templateUrl: 'directive.html' /* or function(tElement, tAttrs) { ... } */,
    transclude: false,
    restrict: 'A',
    templateNamespace: 'html',
    scope: false,
    controller: function($scope, $element, $attrs, $transclude, otherInjectables) { ... },
    controllerAs: 'stringIdentifier',
    bindToController: false,
    require: 'siblingDirectiveName' /* or ['^parentDirectiveName', '?optionalDirectiveName', '?^optionalParent'] */,
    compile: function compile(tElement, tAttrs, transclude) {
      return {
        pre: function preLink(scope, iElement, iAttrs, controller) { ... },
        post: function postLink(scope, iElement, iAttrs, controller) { ... }
      } /* or return function postLink( ... ) { ... } */;
    }, // or
    link: {
      pre: function preLink(scope, iElement, iAttrs, controller) { ... },
      post: function postLink(scope, iElement, iAttrs, controller) { ... }
    } /* or link: function postLink( ... ) { ... } */
  };
});
```

Guide to AngularJS Documentation

> *Best Practice: It's recommended to use the "directive definition object" form.*

AngularJS API Reference - Service components in ng ($compile)

camelCase / snake-case (x-, data-)

```
<body>
  <my:directive></my:directive>
  <my_directive></my_directive>
  <my-directive></my-directive>
  <x-my-directive></x-my-directive>
  <data-my-directive></data-my-directive>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function() {
          return {
            template:'<p>Olá mundo!</p>'
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/rupora

The normalization process is as follows:

- Strip x- and data- from the front of the element/attributes.
- Convert the :, -, or _-delimited name to camelCase.

Guide to AngularJS Documentation

> *A custom data attribute is an attribute in no namespace whose name starts with the string "data-", has at least one character after the hyphen, is XML-compatible, and contains no characters in the range U+0041 to U+005A (LATIN CAPITAL LETTER A to LATIN CAPITAL LETTER Z).*

W3C HTML5 - Embedding custom non-visible data with the data-* attributes

> *Attribute names beginning with the two characters "x-" are reserved for user agent use and are guaranteed to never be formally added to the HTML language. For flexibility, attributes names containing underscores (the U+005F LOW LINE character) are also reserved for experimental purposes and are guaranteed to never be formally added to the HTML language.*

W3C HTML 5.1 - Extensibility

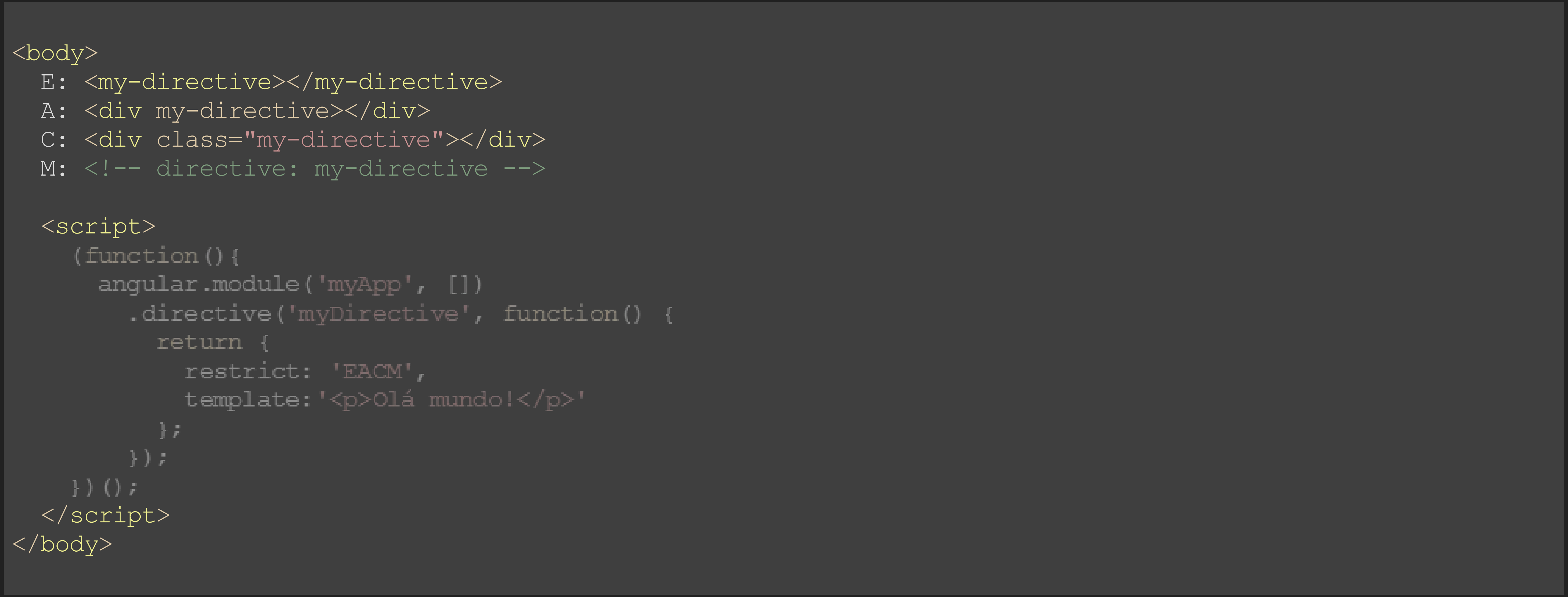
restrict

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      restrict: 'EA'
      ...
    };
  });
```

Restringe o tipo de declaração da diretiva para:

- Elementos: 'E'
- Atributos: 'A'
- Classes: 'C'
- Comentários: 'M'
- Elementos e atributos: 'EA' (*default*)

restrict



```
<body>
  E: <my-directive></my-directive>
  A: <div my-directive></div>
  C: <div class="my-directive"></div>
  M: <!-- directive: my-directive -->

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function() {
          return {
            restrict: 'EACM',
            template:'<p>Olá mundo!</p>'
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/hogubi

> *When creating a directive that makes sense as a stand-alone element, allow restrict E (custom element) and optionally restrict A (custom attribute). Why?: It makes sense.*

Style Y074 - Restrict to Elements and Attributes

templateUrl

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      templateUrl: 'directive.tpl.html'
      ...
    };
  });
```

Carrega o *template* de forma assíncrona.

templateUrl

```
<body>
  <my-directive></my-directive>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function() {
          return {
            templateUrl:'my-directive.tpl.html'
          };
        });
    })();
  </script>

  <script type="text/ng-template" id="my-directive.tpl.html">
    <p>Olá mundo!</p>
  </script>
</body>
```

http://jsbin.com/poraso

replace

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      replace: {boolean},
      ...
    };
  });
```

Substitui o elemento em que a diretiva é declarada pelo seu *template*.

replace: false

```
<body>
  <my-directive>
    <p>Olá mundo!</p>
  </my-directive>
</body>
```

replace: true

```
<body>
  <p>Olá mundo!</p>
</body>
```

replace

```
<body>
  <my-directive></my-directive>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function() {
          return {
            replace:true,
            template:'<p>Olá mundo!</p>'
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/mosaqo

> *“The `replace` flag for defining directives that replace the element that they are on will be removed in the next major angular version. This feature has difficult semantics (e.g. how attributes are merged) and leads to more problems compared to what it solves. Also, with WebComponents it is normal to have custom elements in the DOM.”*

docs($compile): deprecate `replace` directives

transclude

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      transclude: {boolean},
      ...
      template: "...<ng-transclude></ng-transclude>...",
      ...
    };
  });
```

Extrai o conteúdo do elemento em que a diretiva é declarada e o disponibiliza para a diretiva.

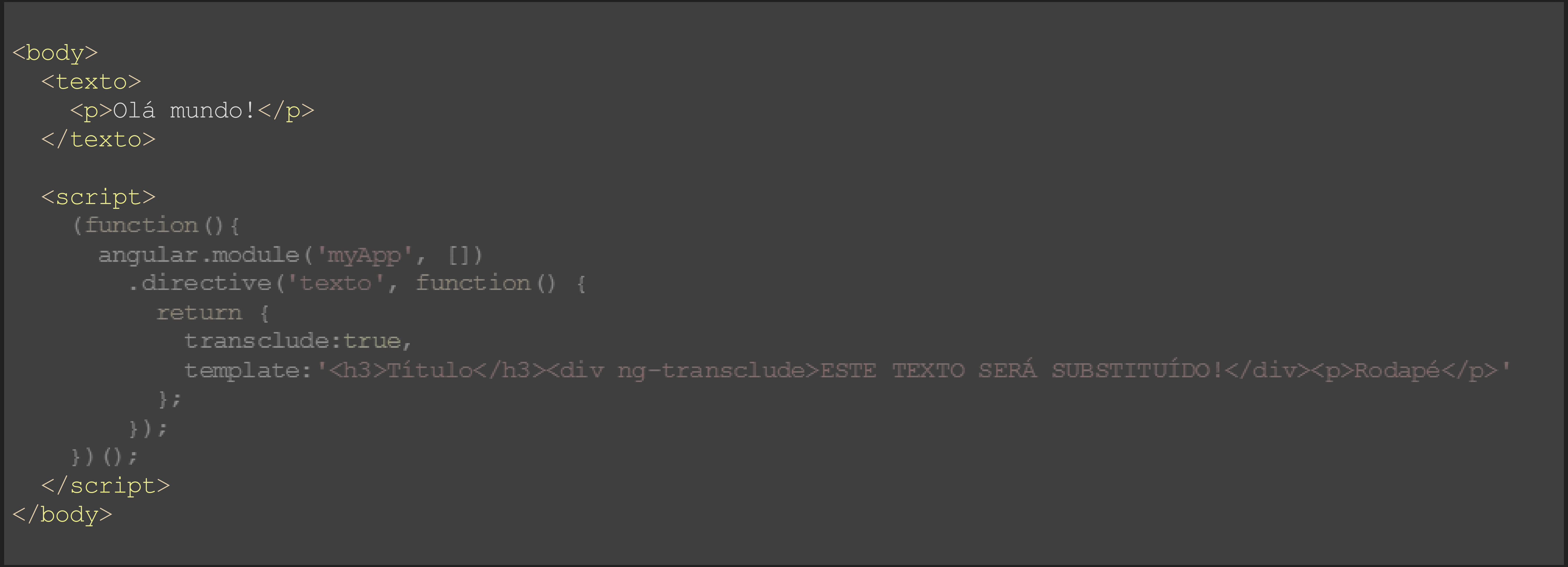
transclude

```
<body>
  <texto>
    <p>Olá mundo!</p>
  </texto>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('texto', function() {
          return {
            transclude:true,
            template:'<h3>Título</h3><ng-transclude></ng-transclude><p>Rodapé</p>'
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/papuvu

transclude (atributo)

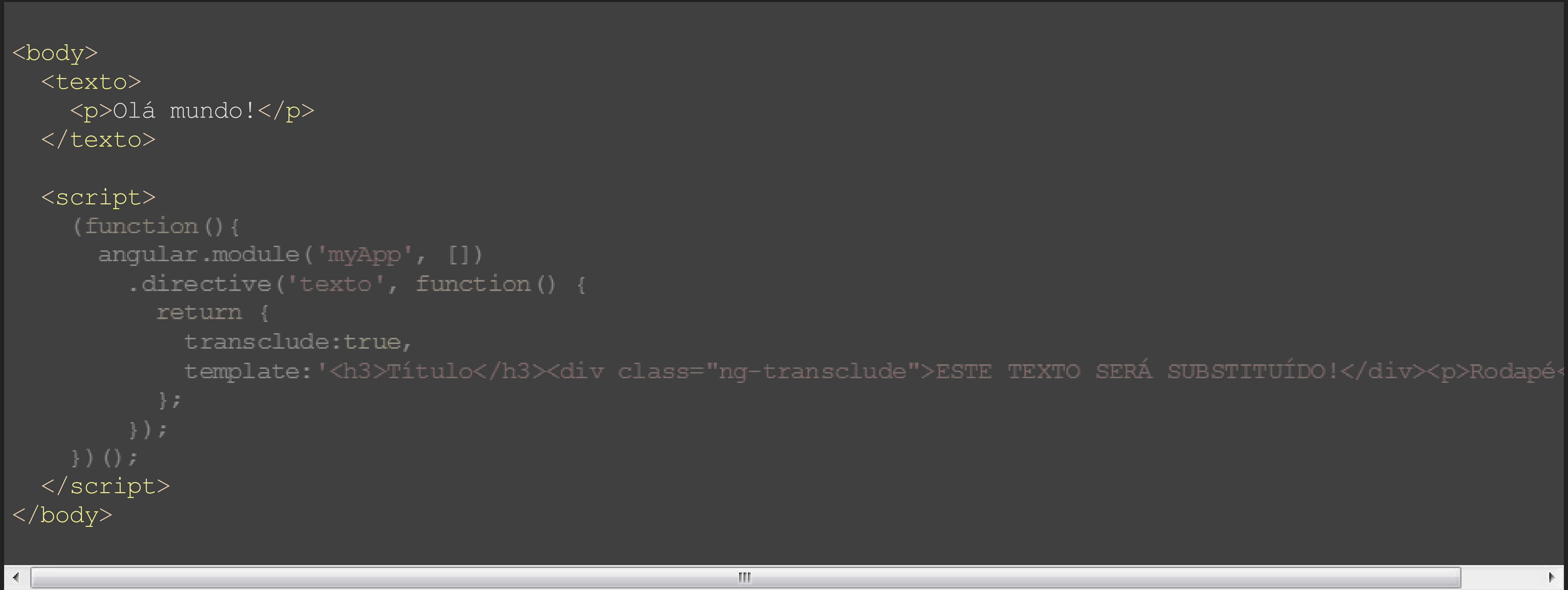


```
<body>
  <texto>
    <p>Olá mundo!</p>
  </texto>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('texto', function() {
          return {
            transclude:true,
            template:'<h3>Título</h3><div ng-transclude>ESTE TEXTO SERÁ SUBSTITUÍDO!</div><p>Rodapé</p>'
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/yepupi

transclude (classe)



```
<body>
  <texto>
    <p>Olá mundo!</p>
  </texto>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('texto', function() {
          return {
            transclude:true,
            template:'<h3>Título</h3><div class="ng-transclude">ESTE TEXTO SERÁ SUBSTITUÍDO!</div><p>Rodapé<
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/lololi

transclude: 'element' + replace: true

```
<body>
  <texto>
    <p>Olá mundo!</p>
  </texto>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('texto', function() {
          return {
            replace:true,
            transclude:'element',
            template:'<div ng-transclude>ESTE TEXTO SERÁ SUBSTITUÍDO!</div>'
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/fikogu

transclude: true

```
<texto>
  <ng-transclude>
    <p class="ng-scope">Olá mundo!</p>
  </ng-transclude>
</texto>
```

transclude: 'element' + replace: true

```
<ng-transclude>
  <texto class="ng-scope">
    <p>Olá mundo!</p>
  </texto>
</ng-transclude>
```

controller

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      controller: {function},
      ...
      controllerAs: {string},
      ...
    };
  });
```

Best Practice: use controller when you want to expose an API to other directives.

controller

```
<body>
  <div texto-expandivel>
    <h3 ng-click="toggle()">Título</h3>
    <p ng-show="mostrar">Olá mundo!</p>
  </div>

  <script>
    angular.module('myApp', [])
      .directive('textoExpandivel', function() {
        return {
          controller:function($scope, $element) {
            $scope.mostrar = false;
            $scope.toggle = function() {
              $scope.mostrar = !$scope.mostrar;
            }
          }
        };
      });
  </script>
</body>
```

http://jsbin.com/detuji

controller

```
angular.module('myApp', [])
  .controller('TextoExpandivelCtrl', function($scope, $element){
    $scope.mostrar = false;

    $scope.toggle = function() {
      $scope.mostrar = !$scope.mostrar;
    };
  })
  .directive('textoExpandivel', function() {
    return {
      controller:'TextoExpandivelCtrl'
    };
  });
```

http://jsbin.com/yukamo

controllerAs

```
angular.module('myApp', [])
  .directive('textoExpandivel', function() {
    return {
      scope: {
        titulo:'=',
        texto:'='
      },
      template: '<h3 ng-click="vm.toggle()">{{titulo}}</h3><p ng-show="vm.mostrar">{{texto}}</p>',
      controller:function($scope, $element) {
        var vm = this;
        vm.mostrar = false;
        vm.toggle = function() {
          vm.mostrar = !vm.mostrar;
        }
      },
      controllerAs: 'vm'
    };
  });
```

http://jsbin.com/yewoka

compile

```
myModule.directive('myDirective', function factory(injectables) {
    return {
      ...
      compile: function compile(tElement, tAttrs, transclude) {
        return {
          pre: function preLink(scope, iElement, iAttrs, controller) { ... },
          post: function postLink(scope, iElement, iAttrs, controller) { ... }
        }
        // or
        // return function postLink( ... ) { ... }
      },
      ...
    };
  });
```

Utilizado para realizar manipulações no DOM.

compile

```
<body>
  <div my-directive></div>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function() {
          return {
            compile:function(elem, attrs) {
              elem.append('Adicionado pelo compile');
            }
          };
        });
    })();
  </script>
</body>
```

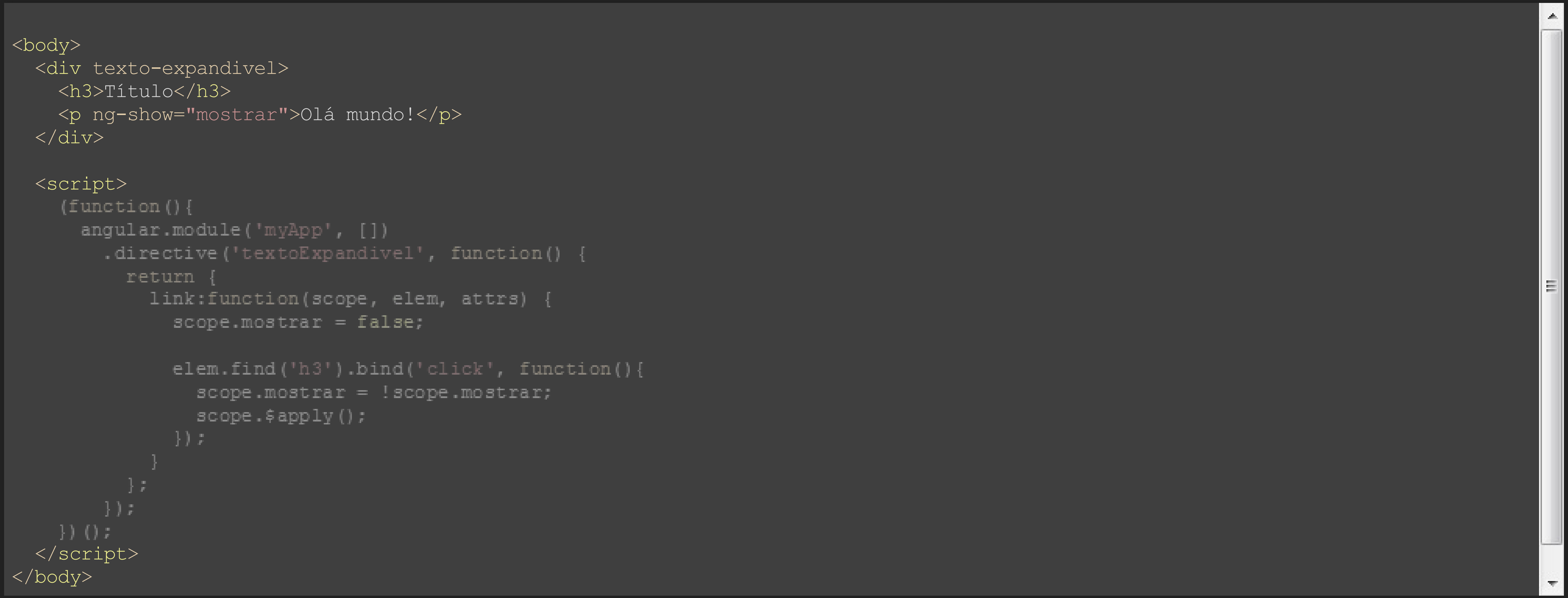
http://jsbin.com/cojeze

link

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      link: {
        pre: function preLink(scope, iElement, iAttrs, controller) { ... },
        post: function postLink(scope, iElement, iAttrs, controller) { ... }
      }
      // or
      link: function postLink( ... ) { ... }
      ...
    };
  });
```

Best Practice: use controller when you want to expose an API to other directives. ***Otherwise use link.***

link



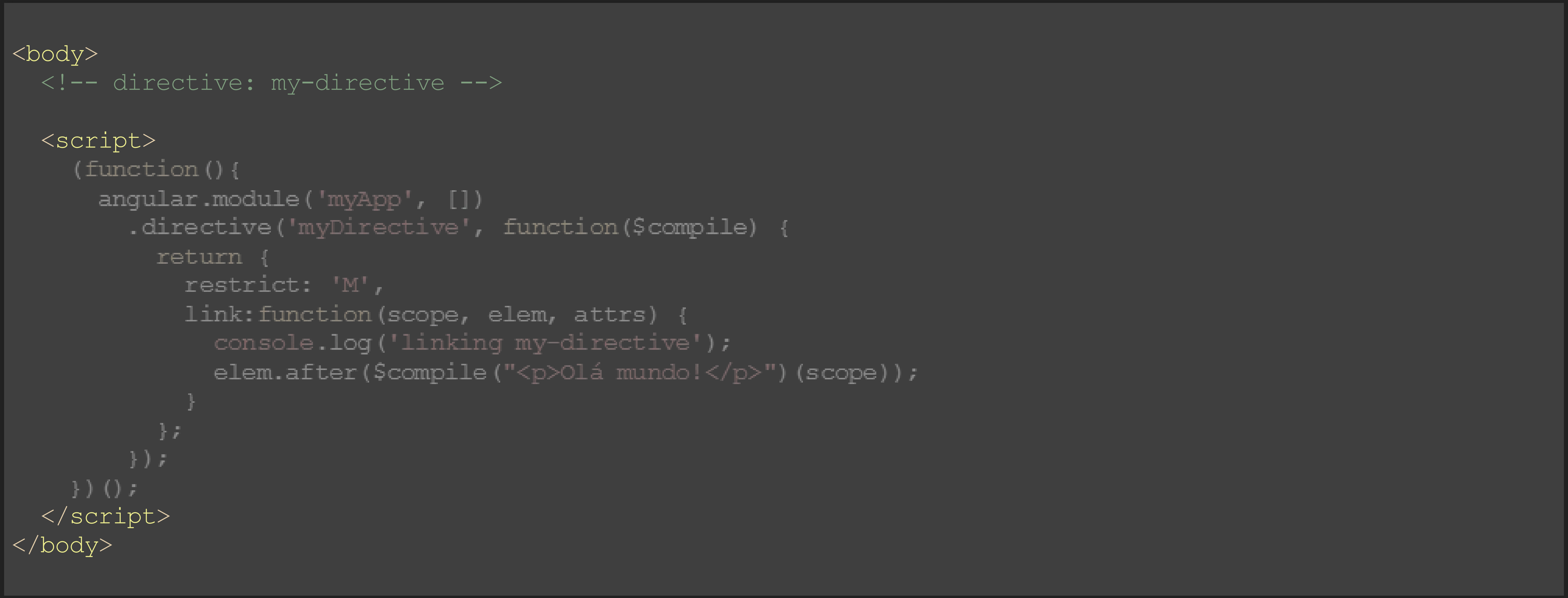
```
<body>
  <div texto-expandivel>
    <h3>Título</h3>
    <p ng-show="mostrar">Olá mundo!</p>
  </div>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('textoExpandivel', function() {
          return {
            link:function(scope, elem, attrs) {
              scope.mostrar = false;

              elem.find('h3').bind('click', function(){
                scope.mostrar = !scope.mostrar;
                scope.$apply();
              });
            }
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/xuruzo

restrict: comment



```
<body>
  <!-- directive: my-directive -->

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function($compile) {
          return {
            restrict: 'M',
            link:function(scope, elem, attrs) {
              console.log('linking my-directive');
              elem.after($compile("<p>Olá mundo!</p>")(scope));
            }
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/kixiho

scopes

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      scope: {boolean} ou {object},
      ...
    };
  });
```

- scope: false - utiliza o escopo pai;
- scope: true - cria um novo escopo que herda do escopo pai;
- scope: object - cria um novo escopo que não herda do escopo pai.

scopes

~~http://jsbin.com/kimige~~

Exemplo com AngularJS Batarang

scope: object

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      scope: {
        foo: '@',
        bar: '=',
        xulambs: '&'
      },
      ...
    };
  });
```

- '@': *bind* de variáveis por valor (*read only*);
- '=': *bind* de variáveis por referência (*read/write*);
- '&': *bind* de expressões ou métodos que serão executados no escopo de que fazem parte.

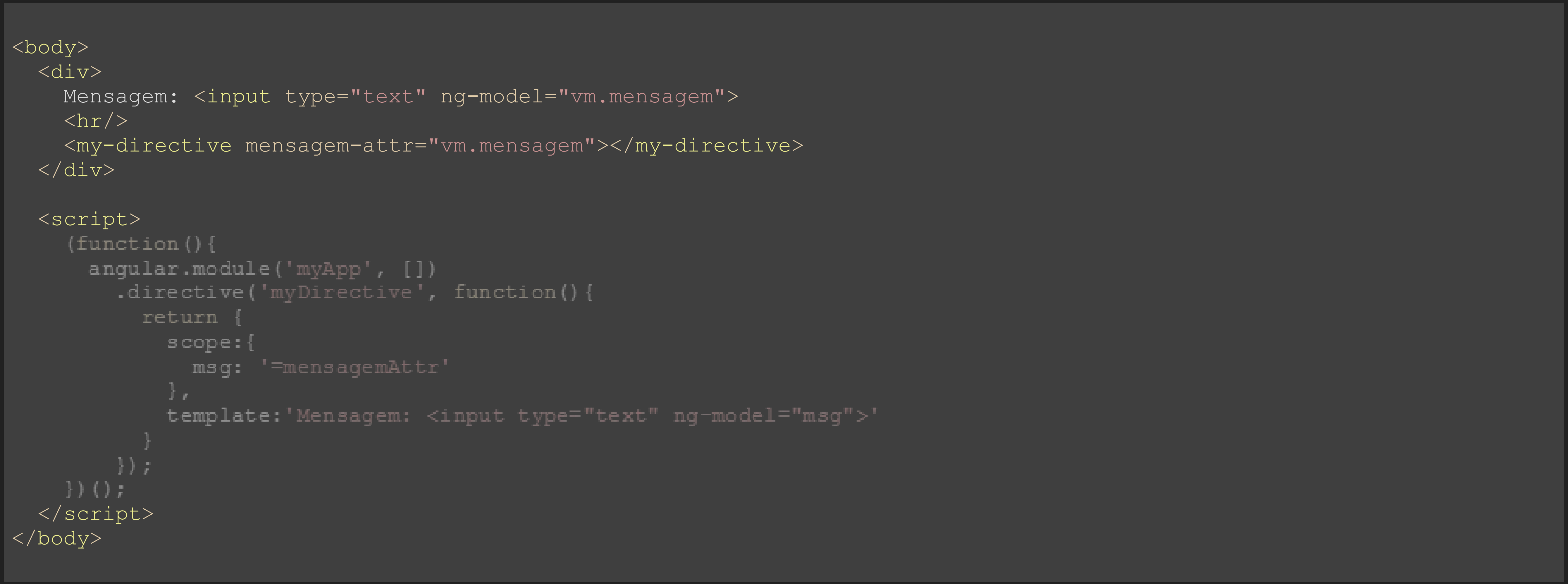
isolated scope - bind (@)

```
<body>
  <div>
    Mensagem: <input type="text" ng-model="vm.mensagem">
    <hr/>
    <my-directive mensagem-attr="{{vm.mensagem}}"></my-directive>
  </div>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function(){
          return {
            scope:{
              msg: '@mensagemAttr'
            },
            template:'Mensagem: {{msg}}'
          }
        });
    })();
  </script>
</body>
```

http://jsbin.com/fopede

isolated scope - bind (=)

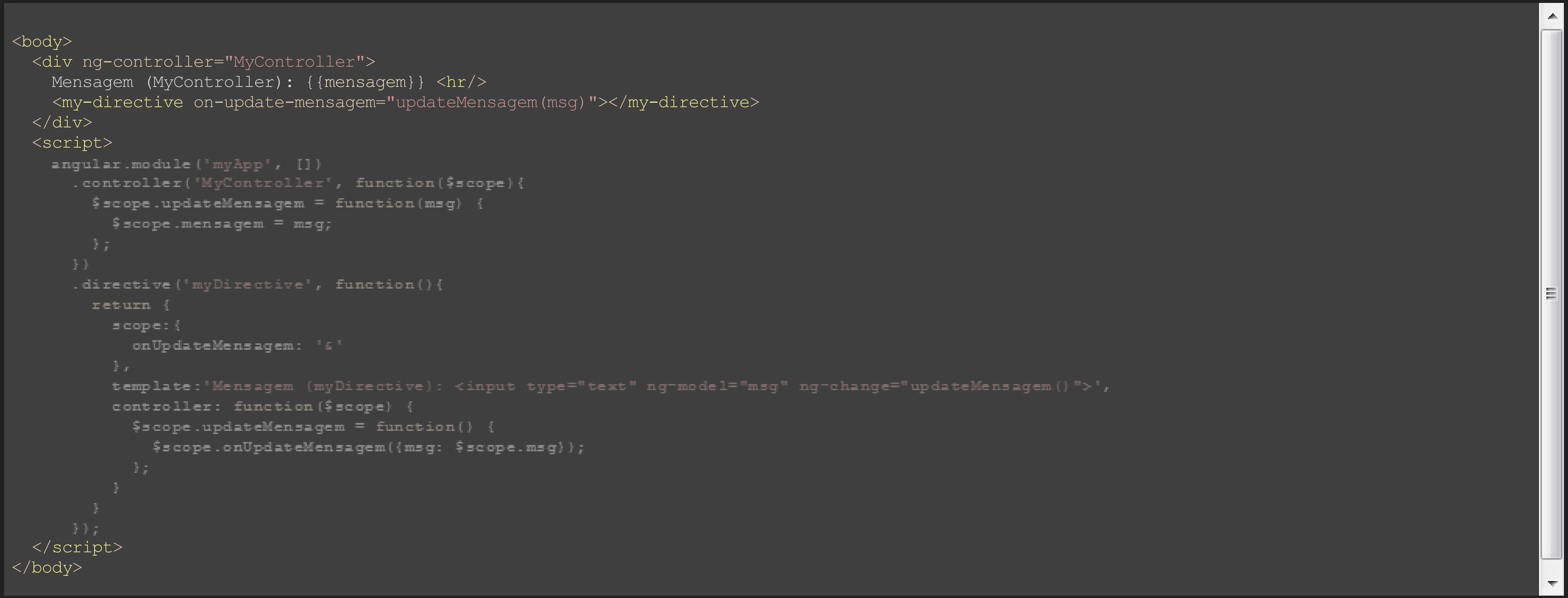


```
<body>
  <div>
    Mensagem: <input type="text" ng-model="vm.mensagem">
    <hr/>
    <my-directive mensagem-attr="vm.mensagem"></my-directive>
  </div>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('myDirective', function(){
          return {
            scope:{
              msg: '=mensagemAttr'
            },
            template:'Mensagem: <input type="text" ng-model="msg">'
          }
        });
    })();
  </script>
</body>
```

http://jsbin.com/hojoci

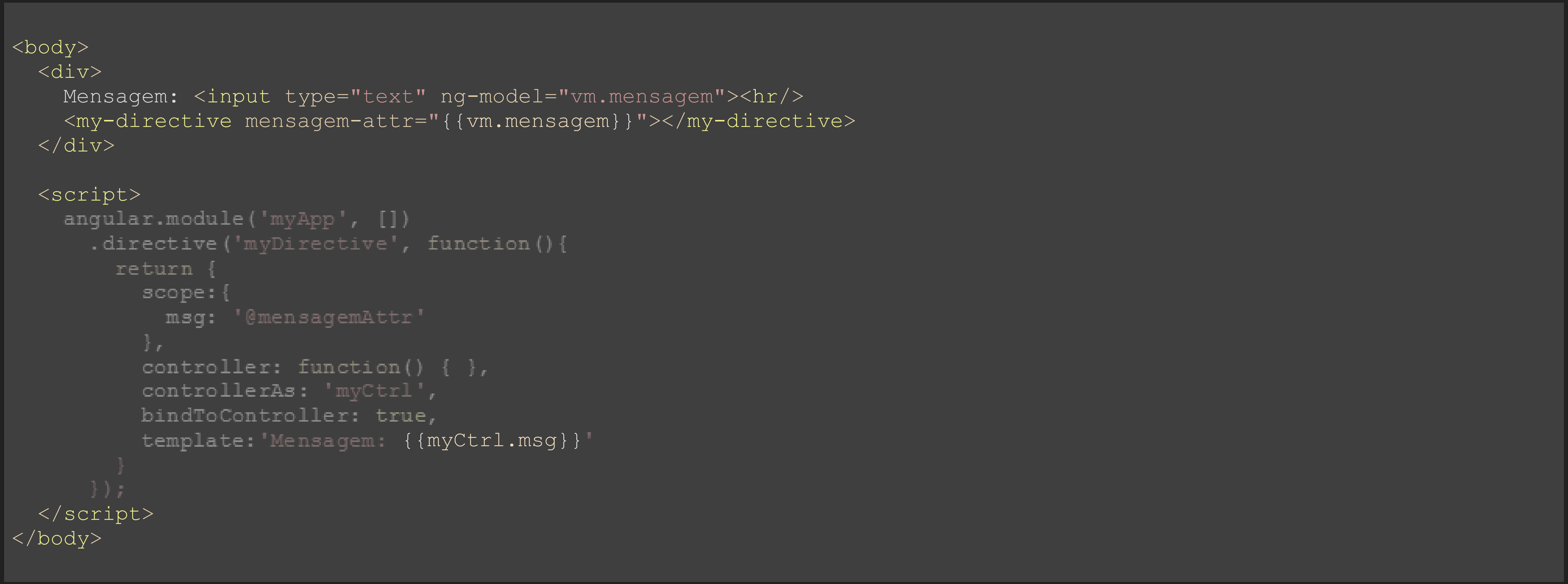
isolated scope - bind (&)



```
<body>
  <div ng-controller="MyController">
    Mensagem (MyController): {{mensagem}} <hr/>
    <my-directive on-update-mensagem="updateMensagem(msg)"></my-directive>
  </div>
  <script>
    angular.module('myApp', [])
      .controller('MyController', function($scope){
        $scope.updateMensagem = function(msg) {
          $scope.mensagem = msg;
        };
      })
      .directive('myDirective', function(){
        return {
          scope:{
            onUpdateMensagem: '&'
          },
          template:'Mensagem (myDirective): <input type="text" ng-model="msg" ng-change="updateMensagem()">',
          controller: function($scope) {
            $scope.updateMensagem = function() {
              $scope.onUpdateMensagem({msg: $scope.msg});
            };
          }
        }
      });
  </script>
</body>
```

http://jsbin.com/pelivu

isolated scope - bindToController



```
<body>
  <div>
    Mensagem: <input type="text" ng-model="vm.mensagem"><hr/>
    <my-directive mensagem-attr="{{vm.mensagem}}"></my-directive>
  </div>

  <script>
    angular.module('myApp', [])
      .directive('myDirective', function(){
        return {
          scope:{
            msg: '@mensagemAttr'
          },
          controller: function() { },
          controllerAs: 'myCtrl',
          bindToController: true,
          template:'Mensagem: {{myCtrl.msg}}'
        }
      });
  </script>
</body>
```

http://jsbin.com/mogisa

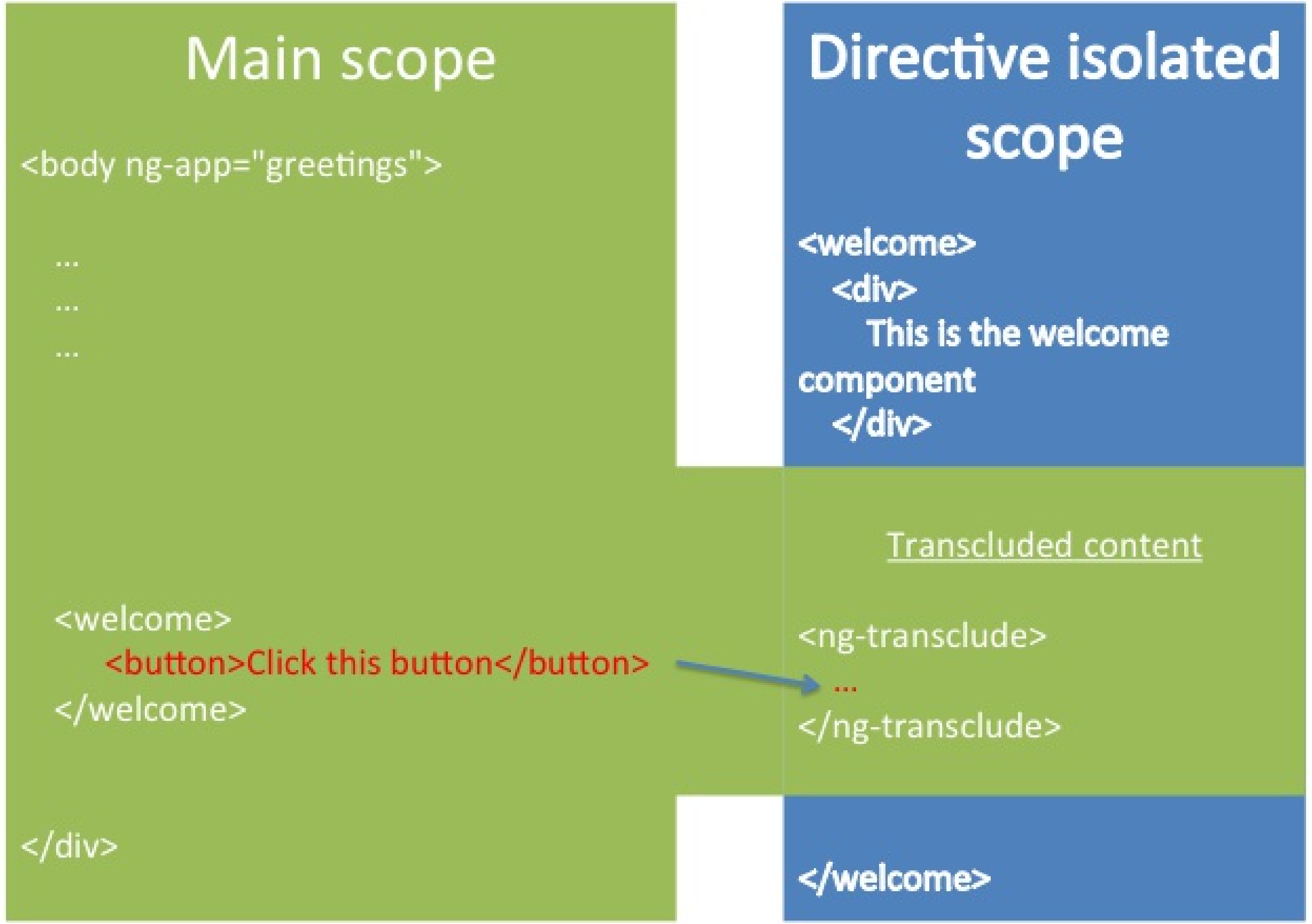
Angular Tips - Transclusion and Scopes (by Jesus Rodriguez)

```
<body ng-init="person={name:'John Doe'};header='Person'">
  <person header="header">
    <h2>{{header}}</h2>
    <p>Hello, I am {{person.name}}</p>
  </person>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('person', function() {
          return {
            scope: { header: '=' },
            transclude:true,
            template: '<div ng-transclude></div>',
            link: function(scope, element, attrs) {
              scope.person = {
                name: 'Directive Joe'
              };

              scope.header = 'Directive\'s header';
            }
          };
      });
    })();
  </script>
</body>
```

http://jsbin.com/turoje



Main scope

```
<body ng-app="greetings">

  ...
  ...
  ...

  <welcome>
     <button>Click this button</button>
  </welcome>

</div>
```

Directive isolated scope

```
<welcome>
  <div>
     This is the welcome
component
  </div>
```

Transcluded content

```
<ng-transclude>
  ...
</ng-transclude>
```

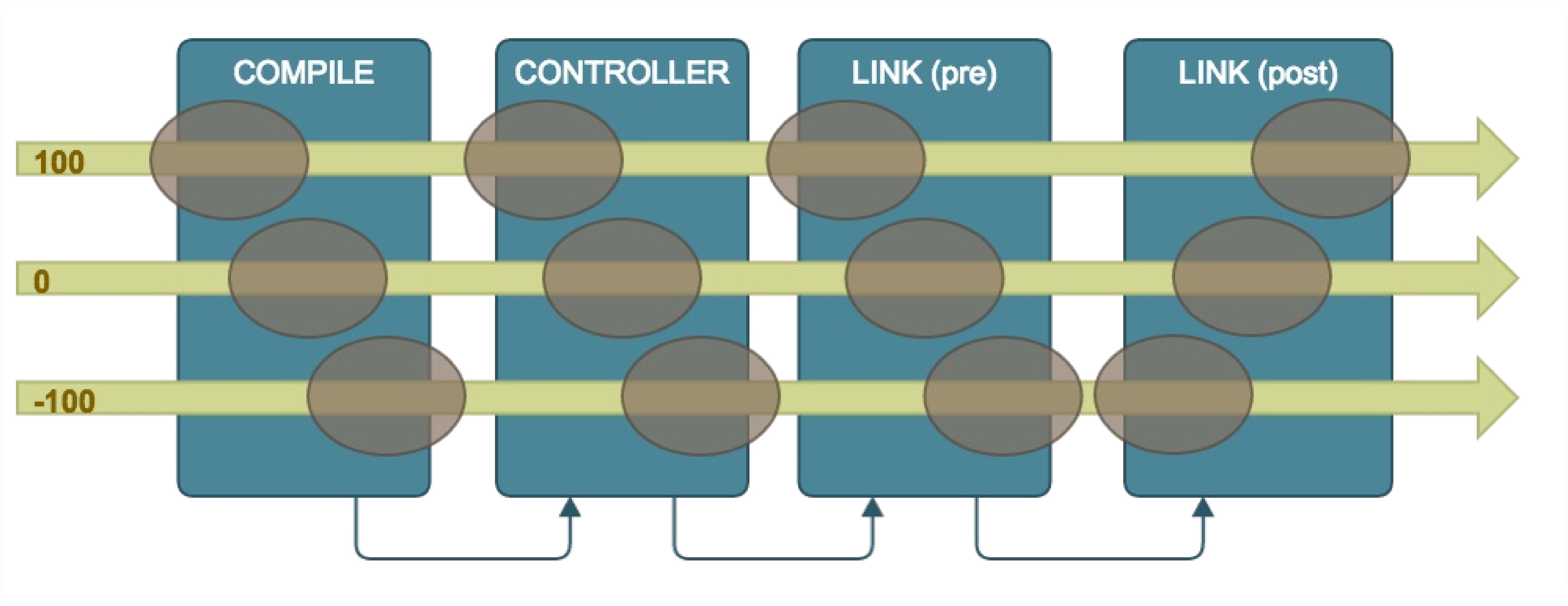
```
</welcome>
```

priority

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      priority: {number},
      ...
    };
  });
```

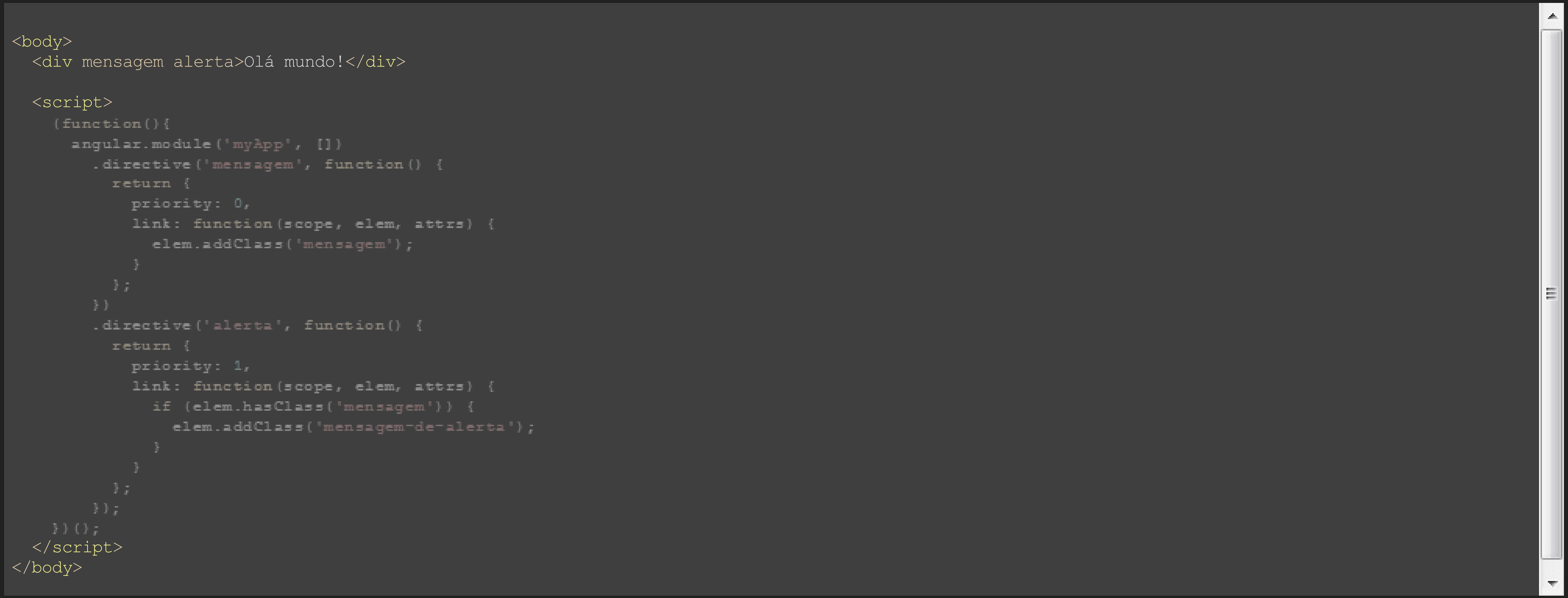
Quando há muitas diretivas num mesmo elemento a prioridade especifica a ordem em que serão compiladas. Diretivas com maior prioridade são compiladas primeiro. *default: 0*

Ordem de execução das funções, conforme prioridade



triangular.io - Directive Compilation in AngularJS – step-by-step

priority



```
<body>
  <div mensagem alerta>Olá mundo!</div>

  <script>
    (function(){
      angular.module('myApp', [])
        .directive('mensagem', function() {
          return {
            priority: 0,
            link: function(scope, elem, attrs) {
              elem.addClass('mensagem');
            }
          };
        })
        .directive('alerta', function() {
          return {
            priority: 1,
            link: function(scope, elem, attrs) {
              if (elem.hasClass('mensagem')) {
                elem.addClass('mensagem-de-alerta');
              }
            }
          };
        });
    })();
  </script>
</body>
```

http://jsbin.com/zefojo

AngularJS standard directive priority list

```
Directive               Priority
---------------         --------
ng-switch               1200
ng-repeat               1000
ng-if                   600
ng-controller           500
ng-include              400
ng-checked              100
ng-disabled             100
ng-href                 99
ng-src                  99
ng-model                1
ng-app                  0
...
```

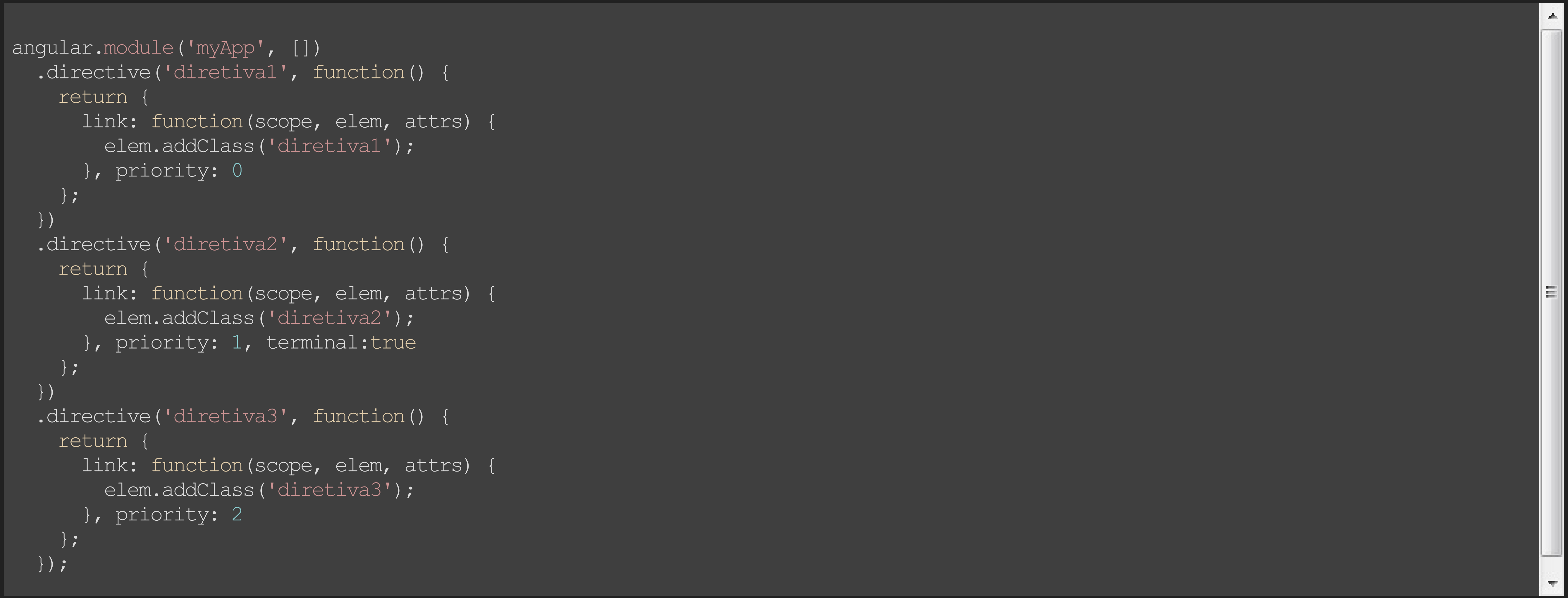
Shripal Soni at Stack Overflow

terminal

```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      terminal: {boolean},
      ...
    };
  });
```

Faz com que as diretivas de prioridade maior que a da diretiva corrente aplicadas ao mesmo elemento não sejam executadas.

terminal



```
angular.module('myApp', [])
  .directive('diretiva1', function() {
    return {
      link: function(scope, elem, attrs) {
        elem.addClass('diretiva1');
      }, priority: 0
    };
  })
  .directive('diretiva2', function() {
    return {
      link: function(scope, elem, attrs) {
        elem.addClass('diretiva2');
      }, priority: 1, terminal:true
    };
  })
  .directive('diretiva3', function() {
    return {
      link: function(scope, elem, attrs) {
        elem.addClass('diretiva3');
      }, priority: 2
    };
  });
```

http://jsbin.com/garizi

require

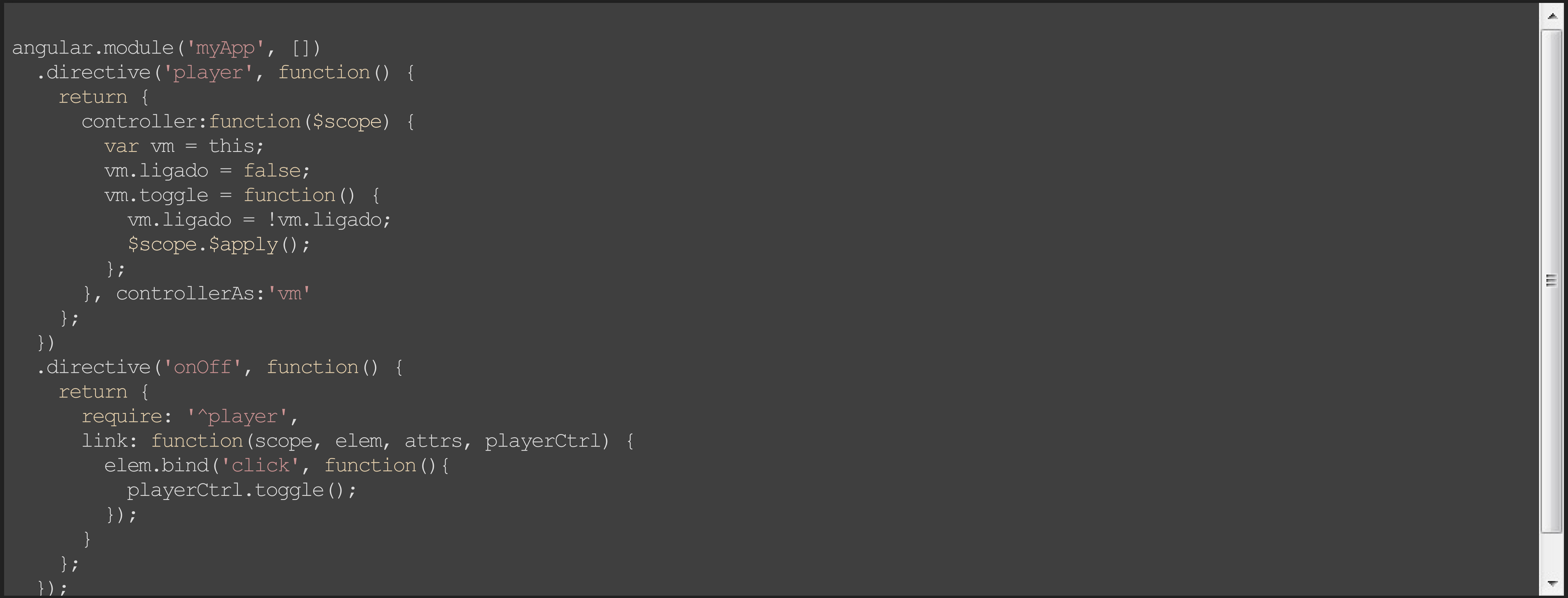
```
angular.module('myApp', [])
  .directive('myDirective', function() {
    return {
      ...
      require: 'siblingDirectiveName',
      // or
      require: ['^parentDirectiveName', '?optionalDirectiveName', '?^optionalParent'],
      ...
    };
  });
```

Injeta o *controller* de outra diretiva na função *link* da diretiva corrente.

- (nenhum prefixo) - pesquisa no elemento corrente;
- ^ - pesquisa no elemento corrente e nos pais;
- ^^ - pesquisa somente nos pais.

? - passa *null* para a função *link* caso o controller não seja encontrado.

require



```
angular.module('myApp', [])
  .directive('player', function() {
    return {
      controller:function($scope) {
        var vm = this;
        vm.ligado = false;
        vm.toggle = function() {
          vm.ligado = !vm.ligado;
          $scope.$apply();
        };
      }, controllerAs:'vm'
    };
  })
  .directive('onOff', function() {
    return {
      require: '^player',
      link: function(scope, elem, attrs, playerCtrl) {
        elem.bind('click', function(){
          playerCtrl.toggle();
        });
      }
    };
  });
```

http://jsbin.com/yagufe

Referências

YouTube: 18 Tips For The Angular Architect

YouTube: Advanced Directives with Angular JS

YouTube: Component-based Directives in Angular

Sharing data between child and parent directives and scopes (in AngularJS) (by Krystal Xu August 14, 2014)

The Hitchhiker’s Guide to the Directive (June 8, 2013 by codef0rmer)

A Practical Guide to AngularJS (by Sandeep Panda - Jan, 2014)

How I've Improved My Angular Apps by Banning ng-controller

AngularJS Directive Attribute Binding Explanation

An AngularJS directive with itself as the attribute

AngularJS Tutorial: Understanding How Directives Work

Directive Compilation in AngularJS – step-by-step

Directive Architecture, Template URLs, And Linking Order In AngularJS

Exercício

Criar diretivas para a aplicação Deputados:

- Conteúdo de cada item da lista de deputados;
- Conteúdo da tela de detalhes do deputado.

Criar tela de favoritos, fazer reuso das diretivas.