**Você sabe mesmo o que são diretivas?**

Temos as seguintes declarações sobre diretivas:

1) Diretivas são criadas através da função .directive.

2) Diretivas retornam opcionalmente um DDO (Directive Definition Object).

3) Diretivas são componentes do Angular que ensinam novos truques para o navegador.

4) Diretivas ensinam novos truques para o navegador, e podem existir sem pertencerem a um módulo

Podemos afirmar que:

* apenas 2 é falsa
* 3, 4 são falsas
* 2, 3, 4 são falsas
* 1, 3 e 4 são verdadeiras
* **1 e 3 são verdadeiras**

**Explicação**

**Diretivas são componentes reutilizáveis que podem encapsular marcação e comportamento. São criadas sempre dentro de um módulo através da função .directive:**

**angular.module('caplositaSA').directive('papagaioDePirata', function() {  
  
});**

**Toda diretiva deve sempre retornar um DDO (Directive Definition Object) configurado para que funcione:**

**angular.module('caplositaSA').directive('papagaioDePirata', function() {  
 var meuDDO = {};  
 // configura o DDO  
 return meuDDO;  
});**

**Nomenclatura**

**Veja a definição da diretiva:**

**angular.module('minhasDiretivas', [])  
 .directive('abasDinamicasEspeciais', function() {  
 var ddo = {};  
 ddo.restrict = 'AE';  
  
 // implementação da diretiva omitida  
 });**

**Quais são todos os possíveis uso da diretiva no HTML?**

* <abas-dinamicas-especiais />
* <div abas-dinamicas-especiais></div>
* **<abas-dinamicas-especiais></abas-dinamicas-especiais> e <div abas-dinamicas-especiais></div>**
* <abasdinamicasespeciais></abasdinamicasespeciais> e <div abas-dinamicas-especiais />

**Explicação**

**Mesmo usando *camelCase* na definição da diretiva, devemos utilizar o hífen na marcação HTML. Com a propriedade restrict declaramos que ela pode ser utilizada tanto como atributo (A) quanto elemento (E).**

**Ou seja, podemos usar a diretiva como elemento:**

**<abas-dinamicas-especiais></abas-dinamicas-especiais>**

**E como atributo:**

**<div abas-dinamicas-especiais></div>**

**Quando usamos como elemento, somos obrigados a abrir e fechar a tag da diretiva, sendo assim, esta forma é inválida:**

**<abas-dinamicas-especiais/>**

**Directive Definition Object**

**Para definir a diretiva no Angular devemos criar um *Directive Definition Object* (DDO), por exemplo:**

**angular.module('minhasDiretivas', [])  
 .directive('meuPainel', function() {  
 var ddo = {};  
  
 //propriedades aqui  
  
 return ddo;  
 });**

**Qual das propriedades abaixos não fazem parte de um DDO?**

* **url**
* scope
* restrict
* template
* transclude

**Explicação**

**A propriedade url não faz parte de um DDO, no entanto existe a propriedadetemplateUrl. A templateUrl permite criar um arquivo HTML separado para toda a marcação HTML dessa diretiva. Ou seja, na diretiva fica apenas o caminho para aquele HTML. Alternativamente podemos usar a propriedade template com toda marcação necessária da diretiva.**

**Lembrando também, a propriedade restrict é para dizer onde a diretiva é valida (Elemento, Atributo, Comentário). A transclude usamos quando a diretiva quer manter elementos filhos e a scope é para capturar informações da diretiva. Não confunda-o com$scope do controller!**

**No exercício vimos a diretiva que representa um painel, usando restrict, transclude,scope e templateUrl:**

**angular.module('minhasDiretivas', [])  
 .directive('meuPainel', function() {  
  
 var ddo = {};  
  
 ddo.restrict = "AE";  
 ddo.transclude = true;  
  
 ddo.scope = {  
 titulo: '@'  
 };  
  
 ddo.templateUrl = 'js/directives/meu-painel.html';   
  
 return ddo;  
 });**

**Segue também o link da documentação oficial sobre diretivas (em inglês) com mais exemplos:**[**https://docs.angularjs.org/guide/directive**](https://docs.angularjs.org/guide/directive)

**Capturar informações da diretiva**

**Veja o uso de uma diretiva para mostrar os dados de um usuario logado:**

**<usuario-logado nome-completo="{{usuario.nomeCompleto}}"></usuario-logado>**

**Como sabemos, para capturar dados da diretiva, usamos a propriedade scope:**

**angular.module('minhasDiretivas', [])  
 .directive('usuarioLogado', function() {  
  
 var ddo = {};  
 ddo.restrict = "AE";  
  
 ddo.scope = {  
 nome: 'AQUI'  
 };  
  
 return ddo;  
 });**

**O que devemos usar no lugar de AQUI para a diretiva funcionar corretamente?**

* @
* **@nome-completo**
* @nome
* nome-completo
* nome

**Explicação**

**Repare que o atributo no elemento HTML se chama nome-completo e no JavaScript da diretiva usamos apenas nome. Se os dois fossem iguais, bastaria colocar @ no lugar de *AQUI*. No entanto, como possuem nomes diferentes, é preciso ser explicito na declaração doscope. Devemos usar @nome-completo para capturar o valor do atributo:**

**ddo.scope = {  
 nome: '@nome-completo'  
};**

**Lembrando também que @ siginfica que o valor é copiado como *string*.**

**É importante reforçar que cada diretiva possui o seu escopo privado!**

**Diretiva, marcação e definição**

**Diretiva, marcação e definição**

**Vejamos a seguinte marcação HTML:**

**<p meu-paragrafo titulo="Bem-vindo ao alura"></p>**

**Qual das opções abaixo cria corretamente a diretiva que nada mais faz do que exibir um parágrafo com otitulo recebido?**

1. **angular.module('meuParagrafo').directive(function() {  
    var ddo = {};  
    ddo.restrict = 'A';  
    ddo.scope = {  
    titulo : '@'  
    };  
    ddo.template = '<p>{{titulo}}</p>';  
    return ddo;  
   }):**
2. **angular.module('minhasDiretivas').directive('meuParagrafo', function() {  
     
    var ddo = {};  
    ddo.restrict = 'A';  
     
    ddo.scope = {  
    titulo : 'titulo'  
    };  
     
    ddo.template = '<p>{{titulo}}</p>';  
    return ddo;  
   }):**
3. **angular.module('minhasDiretivas').directive('meuParagrafo', function() {  
    var ddo = {};  
    ddo.restrict = 'A';  
    ddo.scope = {  
    titulo : '@'  
    };  
    ddo.templateUrl = '<p>{{titulo}}</p>';  
    return ddo;  
   }):**
4. **angular.module('minhasDiretivas').directive('meuParagrafo', function() {  
     
    var ddo = {};  
    ddo.restrict = 'A';  
     
    ddo.scope = {  
    titulo : '@titulo'  
    };  
     
    ddo.template = '<p>{{titulo}}</p>';  
    return ddo;  
   }):**

**Resposta correta: d**

**Explicação**

**Vamos destrinchar a resposta correta:**

**angular.module('minhasDiretivas').directive('meuParagrafo', function() {  
 var ddo = {};  
 ddo.restrict = 'A';  
 ddo.scope = {  
 titulo : '@titulo'  
 };  
 ddo.template = '<p>{{titulo}}</p>'  
 return ddo;  
}):**

**Uma diretiva é criada através da função .directive, porém ela deve ser chamada para um módulo já existente ou para um novo módulo. No código acima, estamos partindo da ideia de que o módulo minhasDiretivas já existe, porque não passamos um array como segundo parâmetro. Muito bem, toda diretiva deve retornar um DDO, caso contrário como o Angular saberá de suas configurações?**

**Diretivas também possuem escopo privado, característica que permite seu reuso, inclusive a existência da mesma diretiva mais de uma vez na marcação HTML. No exemplo acima, parameu-paragrafo terá seu próprio título, sem bagunçar os dos demais. Se temos atributos isolados em uma diretiva, como passamos valores do mundo externo para dentro dela? Isso é feito para uma API, interface de uso baseada em atributos.**

**Se no HTML (uso da diretiva) usamos o atributo titulo, podemos capturar o valor passado para o título da seguinte maneira:**

**scope.titulo = '@titulo'**

**Existe o atributo privado scope.titulo da diretiva e através de @titulo copiamos o valor em string do atributo no HTML para dentro da diretiva. Mas quando o atributo no HTML é igual à propriedade no escopo isolado, podemos deixar apenas @:**

**scope.titulo = '@'**

**Como estamos definindo o template diretamente na diretiva e não em um arquivo externo, usamos a propriedade ddo.template.**