

# TEMA 3

Daniel Estévez  @  & 



&



# Directivas propias

- Al igual que los controladores, las directivas se registran en los módulos
- Para registrar una directiva, se utiliza la API `module.directive`

```
myApp.directive('greeting', function() {...});
```



# Directivas propias (template)

- Una directiva es un markup nuevo a la que se le asocia un comportamiento
- Es decir que lleva asociado un template angular.



&



# Directivas propias (template)

**index.html**

```
<body ng-app="myApp">  
  <div ng-controller="greetingCtrl">  
    <greeting></greeting>  
  </div>  
</body>
```



&



# Directivas propias (template)

## scripts.js

```
var myApp = angular.module('myApp', []);  
myApp.controller('greetingCtrl', function ($scope){  
    $scope.greeting = function(){ $scope.greeting = 'Hola!!! Waaasssup?';}  
});  
myApp.directive('greeting', function() {  
    return {  
        restrict: 'E',  
        template: '<button type="submit" class="btn btn-default" ng-click="greeting()"> {{ greeting }} </button>'  
    };  
});
```



&



# Directivas propias: Template

## Ejercicio

1) Utilizando "gna-como-directiva.html" crea una app Angular que esté compuesta de una directiva `<gna></gna>` creada por ti, que agrupe el código html que tenías hasta ahora



&



# Directivas propias: link

- Las directivas que quieren manipular el DOM utilizan la opción link.
- function `link(scope, element, attrs) { ... }` donde:
  - `scope` es un objeto scope Angular.
  - `element` es el elemento 'jqLite-incrustado'.
  - `attrs` es un objeto hash con pares clave-valor de los nombres de los atributos names y sus correspondientes valores.

# Directivas propias (template)

**index.html**

```
<body ng-app="myApp">  
  <greeting></greeting>  
</body>
```



&





# Directivas propias (link)

## scripts.js

```
var myApp = angular.module('myApp', []);  
myApp.directive('greeting', function() {  
  return {  
    restrict: 'E',  
    template: '<button type="submit" class="btn btn-default" ng-click="greeting()"> {{ greeting }} </button>',  
    link: function (scope) {  
      scope.greeting = function(){scope.greeting = 'Hola!!!  
Waaasssup?'};  
    };  
  });  
});
```



&



# Directivas propias: link

## Ejercicio

2) Utiliza la función link, para poder asociar la lógica de generar números aleatorios a la directiva, y sin hacer más falta el controlador



# Directivas propias (attrs)

- Y desde la directiva podemos acceder a los atributos del markup que hemos definido

## index.html

```
<body ng-app="myApp">  
  <greeting name="Dani"></greeting>  
</body>
```



# Directivas propias (link)

## scripts.js

```
var myApp = angular.module('myApp', []);  
myApp.directive('greeting', function() {  
    return {  
        restrict: 'E',  
        template: '<button type="submit" class="btn btn-default" ng-click="greeting()"> {{ greeting }} </button>',  
        link: function (scope, attrs) {  
            scope.greeting = function(){scope.greeting = 'Hola!!!' + attrs +  
                'Waaasssup?'};  
        };  
    });  
});
```



&



# Directivas propias: link

## Ejercicio

3) Pásale a tu gna a través de un atributo, el intervalo de tiempo en el que quieres que se genere un nuevo número



&

