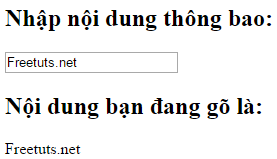
1. AngularJS Directives là gì?

Directives là một thành phần mở rộng HTML, hay nói cách khác là các thuộc tính (properties) của các thẻ HTML mà Angular nó định nghĩa thêm, vì nó của riêng của Angular nên phải tuân thủ theo nguyên tắc của nó là chữ bắt đầu luôn luôn là ký tự mà tôi bôi đỏ ng-prefix, trong đó tiền tố prefix là tên của derective mà chúng ta sử dụng. Như ở các ví dụ trước, để khai báo là một **Directive Controller** thì chúng ta khai báo ng-controller, quá đơn giản phải không nào.

**Ví dụ dưới đây chúng ta sử dụng ba Directives (ng-controller, ng-model, ng-bind)**:

|  |  |
| --- | --- |
|  | <!doctype html>  <html lang="en">      <head>          <meta charset="UTF-8">          <title>Ví dụ sử dụng Directive</title>          <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.2/angular.min.js">  </script>          <script language="javascript">                  angular.module('myapp', [])                  .controller('MyController', function($scope) {                      $scope.name = 'Chào mừng các bạn đến với website freetuts.net';                  });          </script>      </head>      <body ng-app="myapp">          <div ng-controller="MyController">              <h2>Nhập nội dung thông bao:</h2>              <input ng-model="name"> <br/>              <h2>Nội dung bạn đang gõ là:</h2>              <span ng-bind="name"></span>        </div>      </body>  </html> |

**Giao diện**:



Trong ví dụ này mục đích là khi nhập dữ liệu vào ô input thì dòng ở dưới sẽ thay đổi theo. Nếu chúng ta sử dụng [javascript](http://freetuts.net/javascript) để thực hiện thì ta sẽ bắt sự kiện onkeypress cho input và đổi giá trị cho thẻ span. Nhưng với AngularJS thì rất đơn giản, chúng ta sẽ sử dụng**Directive ng-model** để khai báo model và dùng **Directive ng-bind** để lấy data từ model. Nhưng để hai Directives ng-model và ng-bind "*biết đến nhau*"  thì giá trị của nó phải giống nhau nhé (ng-model="name", ng-bind="name").

Các bạn để ý rằng trong đoạn JS tôi chỉ khai báo module, controller nhưng không xử lý gì bên trong. Vậy bạn có thắc mắc là tại sao không code xử lý gì mà lại khai báo chi cho mệt? Lưu ý với bạn đấy là điều bắt buộc nhé, nghĩa là ta phải khai báo để AngularJS hiểu, nếu không nó sẽ báo lỗi đấy.

2. Danh sách AngularJS Directives

Nói đến danh sách **Directives trong AngularJS** thì mình không thể kể ra hết được, để xem được hết danh sách bạn lên trang chủ[Angular API](https://docs.angularjs.org/api) để đọc nhé. Trong bài này mình chỉ nêu ra một vài ví dụ đơn giản cho các bạn dễ hiểu hơn thôi.

ng-app

Khi chúng ta khai báo một Directive ng-app thì AngularJS sẽ hiểu là bắt đầu một ứng dụng Angular, nó sẽ xác định đây là thẻ gốc (element root) và tiếp theo sẽ khởi tạo các thông số cấu hình bên trong mà ta gọi là bootstraps. Ngoài ra ng-app còn được sử dụng để mô tả các module khác nhau trong ứng dụng. Về ví dụ thì chúng ta demo rất nhiều rồi nên tôi không trình bày thêm về nó nũa.

ng-init

Trong lập [trình hướng đối tượng](http://freetuts.net/tag/lap-trinh-huong-doi-tuong) thì chúng ta có khái niệm hàm khởi tạo (constructor), thì ng-init cũng vậy, nó có ý nghĩa là khai báo dữ liệu khởi tạo khi ứng dụng vừa được chạy. Các dữ liệu này sẽ được dùng cho toàn bộ phạm vi của controller mà nó thuộc về. Ví dụ:

|  |  |
| --- | --- |
|  | <!doctype html>  <html lang="en">      <head>          <meta charset="UTF-8">          <title>Ví dụ sử dụng Directive</title>          <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.2/angular.min.js">  </script>          <script language="javascript">                  angular.module('myapp', [])                  .controller('MyController', ['$scope', function($scope) {}]);          </script>      </head>      <body ng-app="myapp">         <div ng-controller="MyController">              <div my-customer ng-init="demo='Nội dung'">{{demo}}</div>              {{demo}}          </div>          {{demo}}      </body>  </html> |

**Giao diện**:

ng-init angularjs

Các bạn thấy rõ ràng mình khai báo ba vị trí khác nhau với cùng key là demo ({{demo}}) nhưng chỉ có hai vị trí nhận được dữ liệu, lý do đơn giản là vị trí cuối cùng năm ngoài [Controller](http://freetuts.net/tim-hieu-controller-trong-angularjs-202.html) của data-init nên không thể nhận được.

ng-...

Còn rất nhiều nữa nhưng bài này mình chỉ mang tính chất giới thiệu nên sẽ không nói thêm về vụ này nữa, chúng ta sẽ lần lượt tìm hiểu từng Directives cụ thể trong các bài sau nhé.

3. Tự định nghĩa Directives

Cái này thú vị nhất nè, lúc trước mình cứ nghĩ là **danh sách AngularJS Directives**là cố đinh, điều này là một sai lầm nhé. Nghĩa là ngoài các Directives mặc định có sẵn trong Angular thì chúng ta có thể tự tạo được.

**Ví dụ 1**: Xây dựng một Directive xuất hiện lời chào với nội dung "*Chào mừng các bạn đến với website freetuts.net*"

|  |  |
| --- | --- |
|  | <!doctype html>  <html lang="en">      <head>          <meta charset="UTF-8">          <title>Ví dụ sử dụng Directive</title>          <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.2/angular.min.js">  </script>          <script language="javascript">                  angular.module('myapp', [])                  .controller('MyController', function($scope) {                      //                  })                  .directive('myDirective', function() {                      return {                          template: '<h1>Chào mừng các bạn đến với website freetuts.net</h1>'                      };                  });          </script>      </head>      <body ng-app="myapp">         <div ng-controller="MyController">              <div my-directive></div>          </div>      </body>  </html> |

**Giao diện**:



Trong ví dụ này mình đã xây dựng một div và khai báo nó là một Directive <div my-directive></div>. Trong đoạn code JS mình sử dụng hàm khai báo một directive là:

|  |  |
| --- | --- |
|  | .directive('myDirective', function() {      return {          template: '<h1>Chào mừng các bạn đến với website freetuts.net</h1>'      };  }); |

Trong đó **myDirective** tương ứng với **my-directive**, nghĩa là khi bạn tạo tên directive trong đoạn code JS thì tiền tố đầu tiên và tiền tố sau được nối lại và tiền tố sau viết chữ in hoa. Bên trong mình xử lý return về một object có key là **template,**key này chính là nội dung HTML bên trong của directive này, không tin bạn đổi giá trị return này và xem lại giao diện nhé.

**Ví dụ 2**: Xây dựng form login bằng cách định nghĩa Directive tên my-formregister.

|  |  |
| --- | --- |
|  | <!doctype html>  <html lang="en">      <head>          <meta charset="UTF-8">          <title>Ví dụ sử dụng Directive</title>          <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.2/angular.min.js">  </script>          <script language="javascript">                  angular.module('myapp', [])                  .controller('MyController', function($scope) {                      //                  })                  .directive('myFormregister', function() {                        var html = '<table>';                          html += '<tr>';                              html += '<td>Username:</td>';                              html += '<td><input type="text"/></td>';                          html += '</tr>';                          html += '<tr>';                              html += '<td>Password:</td>';                              html += '<td><input type="password"/></td>';                          html += '</tr>';                          html += '<tr>';                              html += '<td></td>';                              html += '<td><input type="Button" value="Login"/></td>';                          html += '</tr>';                      html += '</table>';                        return {                          template: html                      };                  });          </script>      </head>      <body ng-app="myapp">         <div ng-controller="MyController">              <div my-formregister></div>          </div>      </body>  </html> |
|  |  |

**Giao diện**:

:

Rất rõ ràng phải không nào, sự khác biệt so với **ví dụ 1** là đoạn code trong javascript return về mã html form login.