



LẬP TRÌNH JAVASCRIPT

BÀI 8: SỬ DỤNG ANGULARJS

- ❑ `undefined` và `null` được ghi thành `' '`
- ❑ Sử dụng `toString()` nếu như giá trị của `object` không phải là `Number`, `Date` hay `Array`

```
myObject.toString !== Object.prototype.  
toString
```

❑ Tiếp đầu ngữ ng phía trước các thuộc tính disabled, required, selected, checked, read và open

Disabled:

```
<input type="checkbox" ng-model="isDisabled" />
```

```
<button ng-disabled="isDisabled">Disabled</button>
```

- ❑ Định nghĩa: Directives là thành phần mở rộng của HTML (Properties của HTML mà Angular định nghĩa thêm)
- ❑ Directives đánh dấu trực tiếp lên DOM (thuộc tính, tên thẻ, comment, css): HTML Compiler (`$compile`)
- ❑ Các tiêu chuẩn
 - ❖ Thêm -x và data- phía trước elements/attributes (thuộc tính/tên thẻ)
 - ❖ Chuyển đổi tên :, - hoặc _

□ Directives chính:

- ❖ ng-app: Khởi tạo ứng dụng AngularJs
- ❖ ng-init: Khởi tạo dữ liệu
- ❖ ng-model: Gắn kết dữ liệu ứng dụng với các đối tượng HTML (input, select, textarea)

❑ Default directives (directives được xây dựng sẵn) :

```
<body ng-app="docsBindExample">
  <div ng-controller="Controller">
    Hello <input ng-model='name'>
    1.<span ng-bind="fail_name"></span>
    2.<span ng:bind="name"></span>
    3.<span ng_bind="name"></span>
    4.<span data-ng-bind="name"></span>
    5.<span x-ng-bind="name"></span>
  </div>
</body>
```

Application

- ✓ ng-app
- ✓ ng-controller

Form

- ✓ ng-pattern
- ✓ ng-minlength
- ✓ ng-maxlength
- ✓ ng-required
- ✓ ng-list
- ✓ ng-true-value
- ✓ ng-false-value
- ✓ ng-options
- ✓ ng-submit

Template

- ✓ ng-csp
- ✓ ng-disabled
- ✓ ng-hide/show
- ✓ ng-if
- ✓ ng-mouse
- ✓ ng-repeat
- ✓ ng-switch
- ✓ ng-transclude
- ✓ ng-view
- ✓ ng-include

Operation

- ✓ ng-change
- ✓ ng-checked
- ✓ ng-click
- ✓ ng-href
- ✓ ng-selected

Binding

- ✓ ng-bind
- ✓ ng-model
- ✓ ng-init
- ✓ ng-src
- ✓ ng-style

❑ Tự định nghĩa directive:

- ❖ Sử dụng hàm **.directive()**
- ❖ Sử dụng trên thẻ html, trên form. Khai báo một directive phải tạo một thẻ html có tên giống tên directive đã tạo.

```
.directive('myDirective', function() {  
    return {  
        template: '<h1>Chào mừng các bạn đến với FPT</h1>'  
    };  
});
```


❑ Có thể sử dụng filter trong controller, service hoặc trực tiếp

❖ Cú pháp

- {{ expression | **filter** }}
- {{ expression | **filter1** | **filter2** | ... }}
- {{ expression | **filter:argument1:argument2:...** }}

❑ Các filter thông dụng

- ❖ currency: định dạng số thành kiểu tiền tệ
- ❖ date: định dạng ngày tháng năm
- ❖ filter: trả về tập con từ mảng
- ❖ json: định dạng đối tượng thành kiểu json
- ❖ limitTo: giới hạn mảng/chuỗi
- ❖ lowercase: chuyển chuỗi sang chữ thường
- ❖ number: chuyển số sang chữ
- ❖ orderBy: sắp xếp mảng theo thứ tự
- ❖ uppercase: chuyển chuỗi sang chữ hoa

□ Ví dụ

- `<p> Cao đẳng {{ 'fpt' | uppercase }} </p>`
- Hoặc `<h1> Price: {{ price | currency }} </h1>`

□ Filter và mảng

```
(function(angular) {  
    'use strict';  
    angular.module('FilterInControllerModule', []).  
        controller('FilterController', ['filterFilter',  
            function FilterController(filterFilter) {  
                this.array = [  
                    {name: 'Cao đẳng'},  
                    {name: 'FPT'},  
                    {name: 'Polytechnic'}  
                ];  
  
                this.filteredArray = filterFilter(this.array, 'a');  
            }  
        ]  
    })(window.angular);
```

❑ Filter và mảng

```
<body ng-app="FilterInControllerModule">
  <div ng-controller="FilterController as ctrl">
    <div>
      All entries:
      <span ng-repeat="entry in ctrl.array">{{entry.name}} </span>
    </div>
    <div>
      Entries that contain an "a":
      <span ng-repeat="entry in ctrl.filteredArray">{{entry.name}}
    </span>
    </div>
  </div>
</body>
```

❑ Filter tự định nghĩa

```
(function(angular) {  
    'use strict';  
    angular.module('myReverseFilterApp', [])  
        .filter('reverse', function() {/*custom filter*/  
            //code here  
        })  
        .controller('MyController', ['$scope', 'reverseFilter', function($scope, reverseFilter) {/*filter in control*/  
            //code here  
        }]);  
})(window.angular);
```



LẬP TRÌNH JAVASCRIPT

BÀI 8: SỬ DỤNG ANGULARJS

- ❑ Các điều khiển (control) trong form gồm: input, select, textarea
- ❑ Sử dụng **ngModel** để ràng buộc dữ liệu 2 chiều giữa ***model*** và ***view***

Demo

index.html

```

1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Example - example-forms-simple-production</title>
6
7    <script src="//code.angularjs.org/snapshot/
      angular.min.js"></script>
8
9  </head>
10
11
12  <body ng-app="formExample">
13    <div ng-controller="ExampleController">
14      <form novalidate class="simple-form">
15        <label>Name: <input type="text" ng-model="user.name" /></
          label><br />
16        <label>E-mail: <input type="email" ng-model="user.email" /
          ></label><br />
17        <input type="button" ng-click="reset()" value="Reset" />
18        <input type="submit" ng-click="update(user)" value="Save"
          />
19      </form>
20      <pre>user = {{user | json}}</pre>
21      <pre>master = {{master | json}}</pre>
22    </div>
23

```

Preview

Name:

E-mail:


```

user = {
  "name": "FPT Poly",
  "email": "phunhuan@fe.edu.vn"
}

master = {
  "name": "FPT Poly",
  "email": "phunhuan@fe.edu.vn"
}

```

❑ Sử dụng **ngModel** để ràng buộc dữ liệu 2 chiều giữa *model* và *view*

```
<script>
angular.module('formExample', [])
    .controller('ExampleController', ['$scope',
        function($scope) {
            $scope.master = {};
            $scope.update = function(user) {
                $scope.master = angular.copy(user);
            };
            $scope.reset = function() {
                $scope.user = angular.copy($scope.master);
            };
            $scope.reset();
        }
    ])
);
</script>
```

❑ Sử dụng class CSS trong AngularJS

- ❖ `ng-valid`
- ❖ `ng-invalid`
- ❖ `ng-valid-[key]`
- ❖ `ng-invalid-[key]`
- ❖ `ng-pristine`
- ❖ `ng-dirty`
- ❖ `ng-touched`
- ❖ `ng-untouched`
- ❖ `ng-pending`

❑ Sử dụng class CSS trong AngularJS

```
<div ng-controller="ExampleController">
<form novalidate class="css-form">
<label>Name: <input type="text" ng-model="user.name" required
/></label><br />
<label>E-mail: <input type="email" ng-model="user.email" required
/></label><br />
</form> </div>
```

```
<style type="text/css">
.css-form input.ng-invalid.ng-touched { background-color: #FA787E; }
.css-form input.ng-valid.ng-touched { background-color: #78FA89; }
</style>
```

□ Kết nối form và các lệnh điều khiển

```
<label>E-mail:
<input type="email" ng-model="user.email" name="uEmail"
required="" /></label><br />
<div ng-show="form.$submitted || form.uEmail.$touched">
<span ng-show="form.uEmail.$error.required">Tell us your
email.</span>
<span ng-show="form.uEmail.$error.email">This is not a valid
email.</span>
</div>
```

❑ Kết nối form và các lệnh điều khiển

Gender:

```
<label><input type="radio" ng-model="user.gender" value="male"
/>male</label>
<label><input type="radio" ng-model="user.gender" value="female"
/>female</label>
<br />
<label>
<input type="checkbox" ng-model="user.agree" name="userAgree"
required="" />
```

❑ Validation cho các thẻ html5 (text, number, url, email, date, radio, checkbox)
gồm

- ❖ required
- ❖ pattern
- ❖ minlength
- ❖ maxlength
- ❖ min
- ❖ max

❑ Validation

- ❖ `ngModelController,`
`ngModelController.$error`
- ❖ `$validators (modelValue, viewValue)`
- ❖ `$setValidity (true: valid, false:`
`invalid)`

- ❑ SV có thể tham khảo tại địa chỉ
 - ❖ <http://docs.angularjs.org/guide/>

FPT POLYTECHNIC



KẾT THÚC