



FRONT-END FRAMEWORKS

BÀI 5: \$SCOPE VÀ CHỈ THỊ

- ⊙ Hiểu và sử dụng \$scope và \$rootScope
- ⊙ Hiểu và sử dụng các chỉ thị cơ bản
- ⊙ Hiểu và sử dụng các chỉ thị điều khiển
- ⊙ Sử dụng các chỉ thị để tạo các trang web thông dụng



- 📖 Tìm hiểu \$scope
- 📖 \$scope lồng nhau
- 📖 Tìm hiểu \$rootScope
- 📖 Tìm hiểu các chỉ thị cơ bản
- 📖 Tìm hiểu các chỉ thị điều khiển giao diện
- 📖 Làm việc với bảng
- 📖 Tạo trang sản phẩm



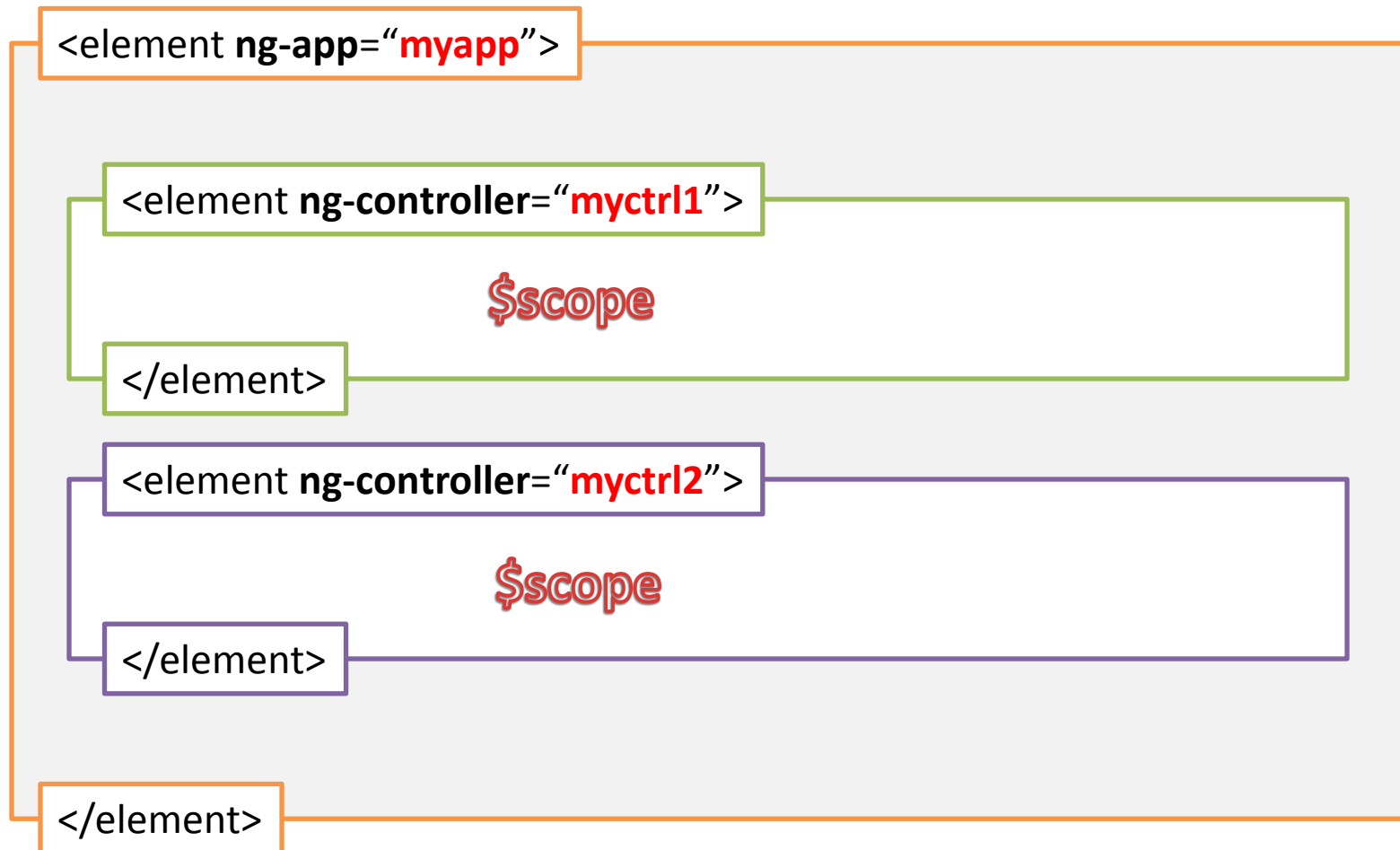
- ❑ Trong AngularJS, Controller là một hàm có nhiệm vụ thực hiện xử lý dữ liệu trong đối tượng \$scope. Dữ liệu sẽ được đồng bộ lên giao diện (View) thông qua cơ chế buộc dữ liệu.
- ❑ Khi một controller được gán vào một thẻ HTML bằng thuộc tính chỉ thị **ng-controller** thì AngularJS sẽ tạo ra một đối tượng **\$scope** mới chứa dữ liệu và các phương thức dành riêng cho controller làm việc với phạm vi giao diện của thẻ đó.
- ❑ Trong phạm vi ứng dụng angular có thể có nhiều controller. Mỗi controller điều khiển 1 view riêng và có \$scope quản lý dữ liệu riêng

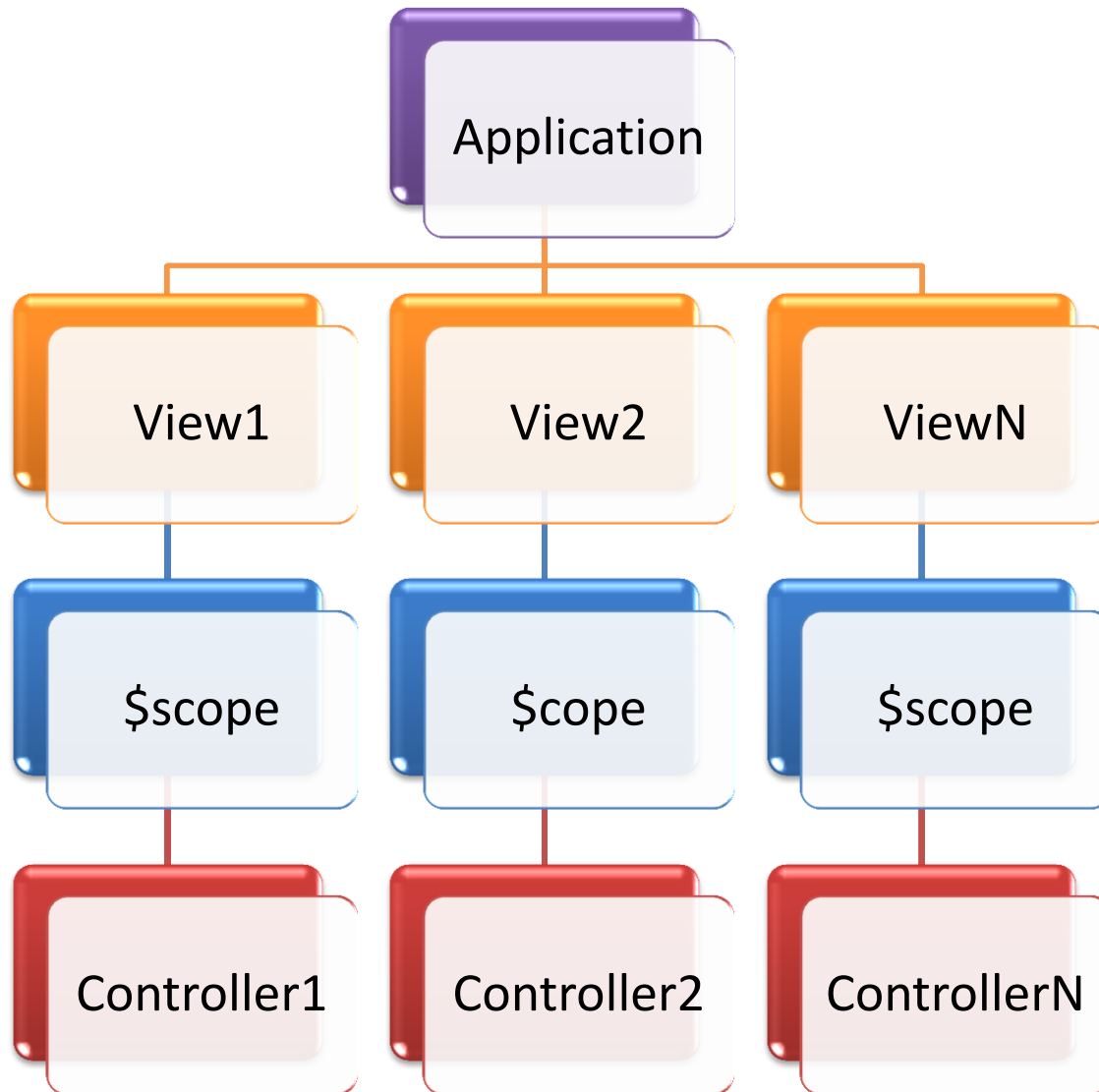
```
<body ng-app="myapp">
  <div ng-controller="myctrl1">
    ...
  </div>
  <div ng-controller="myctrl2">
    ...
  </div>
  <script>
    var app = angular.module('myapp', []);
    app.controller('myctrl1', function ($scope) {...});
    app.controller('myctrl2', function ($scope) {...});
  </script>
</body>
```

❑ Đoạn mã trên có thể diễn giải như sau

- ❖ `var app = angular.module("myapp", [])`
 - Tạo một module làm việc với `<body ng-app="myapp">`
- ❖ `app.controller("myctrl2", function($scope){...})`
 - Chỉ định hàm xử lý `<div ng-controller="myctrl2">`
- ❖ `$scope` xử lý dữ liệu từng vùng giao diện

- ❑ Trong một ứng dụng có thể chứa nhiều \$scope. Mỗi \$scope có dữ liệu và hàm xử lý riêng





```

<div ng-controller="redctrl">
  <h3>Red: {{redmsg}}</h3>
  <h3>Blue: {{bluemsg}}</h3>
</div>
<hr />
<div ng-controller="bluectrl">
  <h3>Red: {{redmsg}}</h3>
  <h3>Blue: {{bluemsg}}</h3>
</div>
<script>
  var app = angular.module("myapp", []);
  app.controller("redctrl", function ($scope) {
    $scope.redmsg = "I am in scope of redctrl";
  });
  app.controller("bluectrl", function ($scope) {
    $scope.bluemsg = "I am in scope of bluectrl";
  });
</script>

```

Red: I am in scope of redctrl

Blue:

Red:

Blue: I am in scope of bluectrl

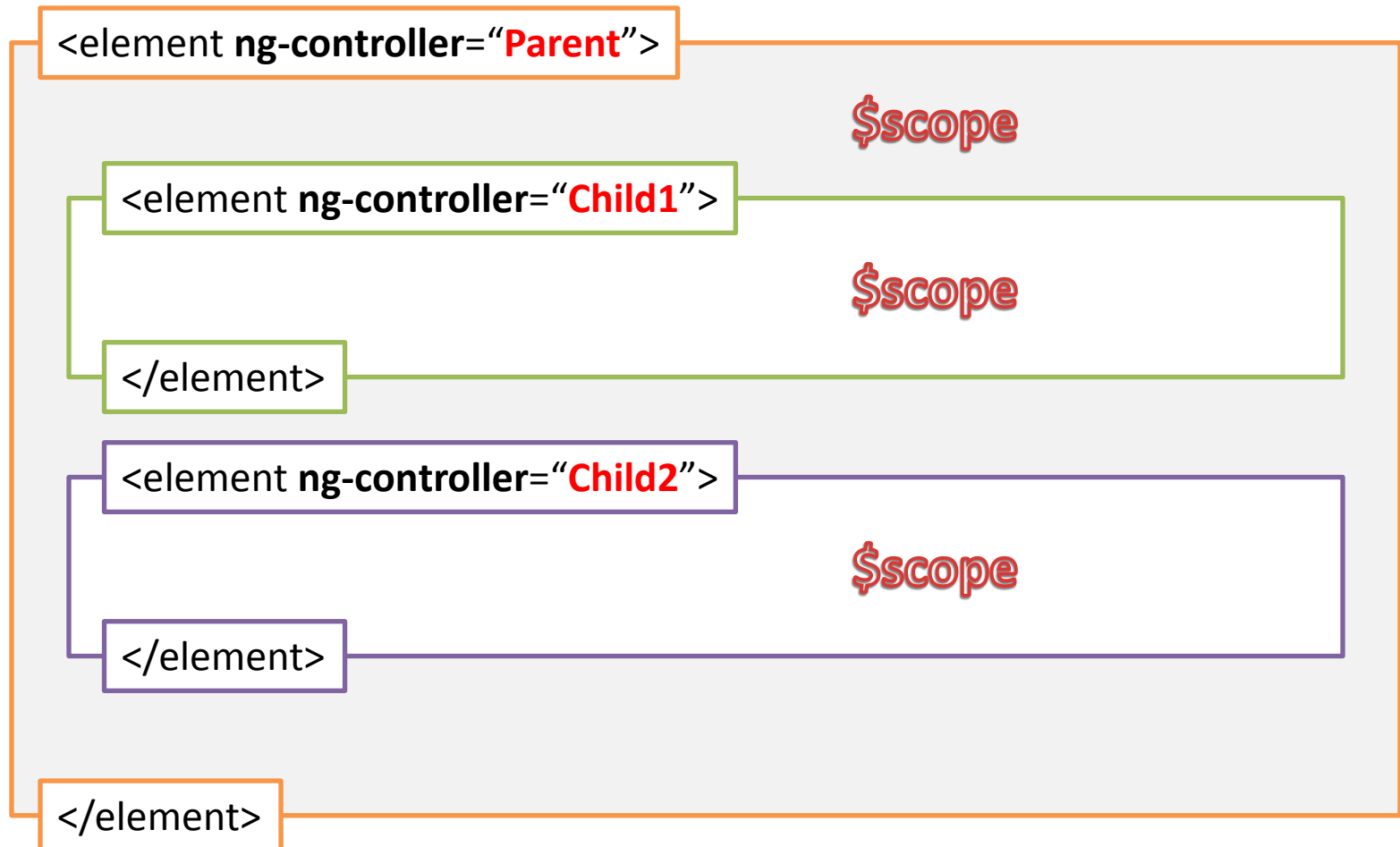
- ❑ Dữ liệu trong mỗi phạm vi là khác nhau. Không thể truy xuất dữ liệu của phạm vi khác.



FPT Polytechnic HCM

Username Password

Tạo 2 controller quản lý
2 vùng giao diện gồm
+ Thông tin hình ảnh
+ Thông tin form

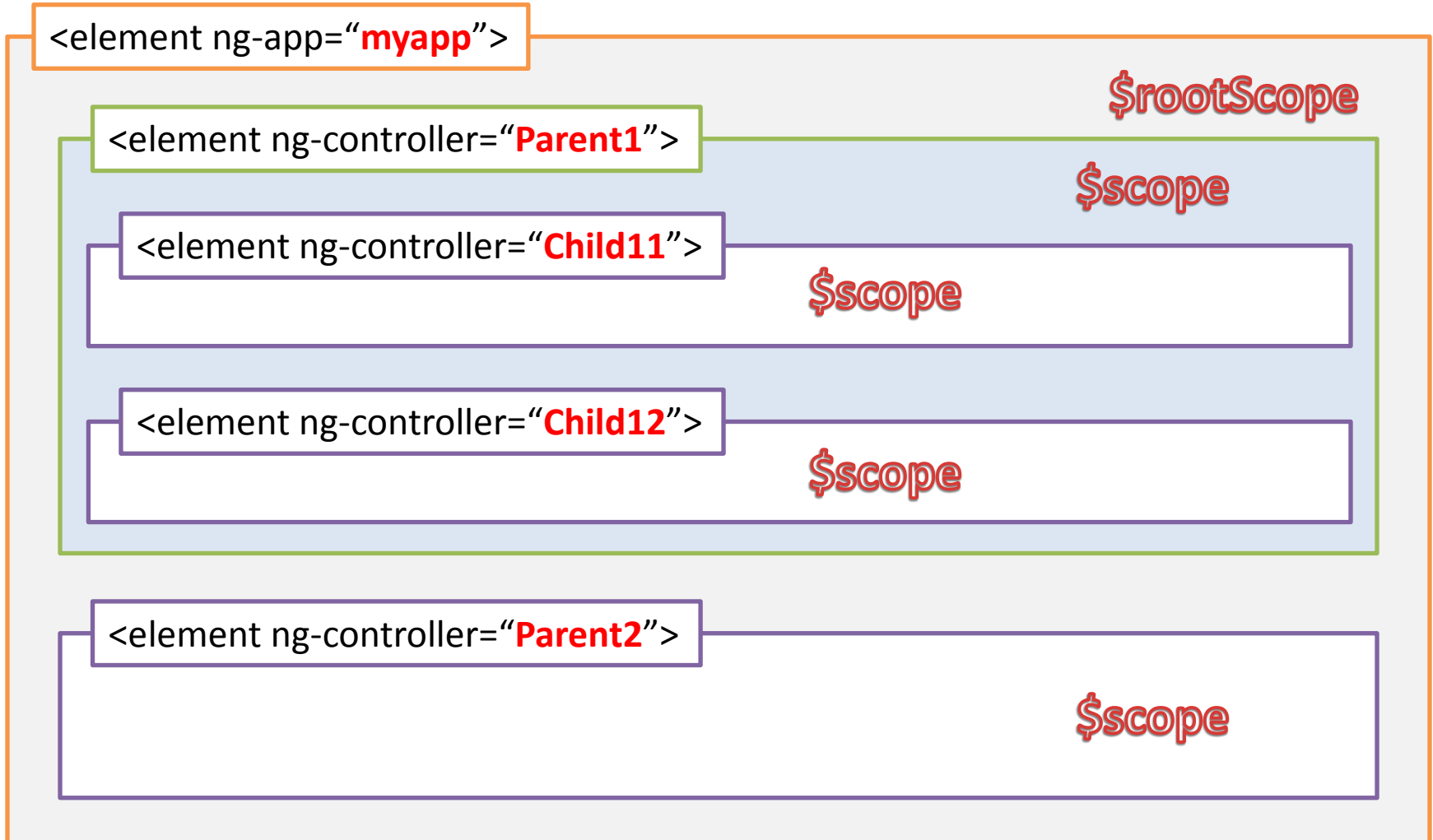


Trong \$scope con có thể truy xuất trực tiếp thành viên của \$scope cha hoặc thông qua \$parent

```
<ul ng-controller="parentCtrl">
  <li>Parent: {{parentmsg}}</li>
  <li>Child: {{childmsg}}
    <ul ng-controller="childCtrl">
      <li>Child: {{childmsg}}</li>
      <li>Parent1: {{parentmsg}}</li>
      <li>Parent2: {{$parent.parentmsg}}</li>
    </ul>
  </li>
</ul>
<script>
  var app = angular.module("myapp", []);
  app.controller("parentCtrl", function ($scope) {
    $scope.parentmsg = "I am parent";
  });
  app.controller("childCtrl", function ($scope) {
    $scope.childmsg = "I am child";
  });
</script>
```

- Parent: I am parent
- Child:
 - Child: I am child
 - Parent1: I am parent
 - Parent2: I am parent

- ☐ \$scope cha không thể truy xuất \$scope con
- ☐ \$scope con có thể truy xuất trực tiếp \$scope cha hoặc thông qua \$parent



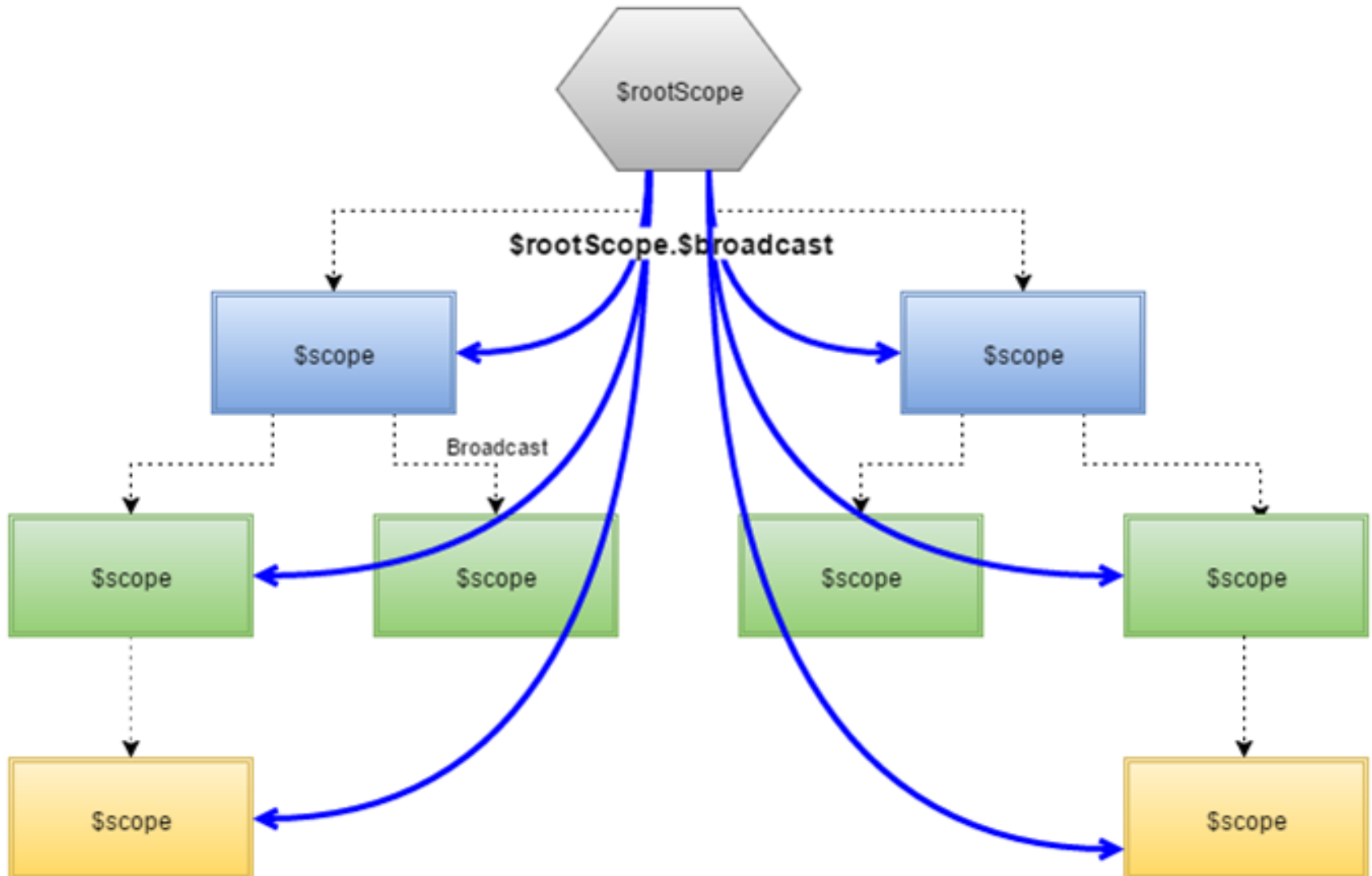
\$rootScope quản lý dữ liệu và hàm dùng chung cho mọi \$scope bên trong

```
<h1>Root: {{rootmsg}}</h1>
<ul ng-controller="parentCtrl">
  <li>Parent: {{parentmsg}}
    <ul ng-controller="childCtrl">
      <li>Child: {{childmsg}}</li>
      <li>Root1: {{rootmsg}}</li>
      <li>Root2: {{$parent.$parent.rootmsg}}</li>
    </ul>
  </li>
</ul>
<script>
  var app = angular.module("myapp", []);
  app.controller("parentCtrl", function ($scope, $rootScope) {
    $rootScope.rootmsg = "I am root";
    $scope.parentmsg = "I am parent";
  });
  app.controller("childCtrl", function ($scope) {
    $scope.childmsg = "I am child";
  });
</script>
```

Root: I am root

- Parent: I am parent
 - Child: I am child
 - Root1: I am root
 - Root2: I am root

- ❑ \$rootScope có thể truy xuất trực tiếp trong phạm vi ng-app hoặc sử dụng \$parent
- ❑ Cần khai báo đối số \$rootScope nếu controller muốn làm việc với \$rootScope



Cập nhật dữ liệu trong scope cha sẽ tự động broadcast đến các scope con



FRONT-END FRAMEWORKS

BÀI 5 (PHẦN 2)

❑ AngularJS định nghĩa rất nhiều thuộc tính HTML mở rộng phục vụ cho mục đích riêng. Các thuộc tính này thường bắt đầu ng-* và được gọi là chỉ thị

❑ Các chỉ thị cơ bản

❖ @ng-app

➤ Khai báo phạm vi View cho ứng dụng AngularJS

❖ @ng-controller

➤ Khai báo phạm vi View được điều khiển bởi một Controller

❖ @ng-init

➤ Khởi tạo thuộc tính dữ liệu trong \$scope

❖ @ng-bind

➤ Buộc dữ liệu vào 1 thẻ HTML thay cho biểu thức {{}}


```
<body ng-app="myapp" ng-init="a=2000">
  <h1>Root: {{a}}</h1>
  <ul ng-controller="parentCtrl" ng-init="b=1000">
    <li>Parent: {{b}}
      <ul ng-controller="childCtrl" ng-init="c=3000">
        <li>Child: {{a}}</li>
        <li>Parent: {{b}}</li>
        <li>Root: {{c}}</li>
      </ul>
    </li>
  </ul>
  <script>
    var app = angular.module("myapp", []);
    app.controller("parentCtrl", function ($scope) {
    });
    app.controller("childCtrl", function ($scope, $rootScope) {
      alert($rootScope.a);
    });
  </script>
</body>
```

Root: 2000

- Parent: 1000
 - Child: 2000
 - Parent: 1000
 - Root: 3000

❑ @ng-init ở phạm vi nào thì thuộc tính tạo ra thuộc phạm vi đó

❑ Trong AngularJS định nghĩa các chỉ thị giúp điều khiển giao diện

❖ @ng-show="expression"

➤ Hiển thị khi biểu thức có giá trị true

❖ @ng-hide="expression"

➤ Ẩn khi biểu thức có giá trị true

❖ @ng-if="expression"

➤ Thực hiện khi biểu thức có giá trị là false

❖ @ng-switch="expression"

➤ Thực hiện @ng-switch-when="value" khi biểu thức có giá trị là value

❖ @ng-repeat="array"

➤ Duyệt các phần tử trong mảng

❑ <element **ng-show**="expr">...</element>

❖ Hiển thị thẻ <element> nếu biểu thức expr là true

❖ Ví dụ:

`Loading`

❑ <element **ng-hide**="expr">...</element>

❖ Ẩn thẻ <element> nếu biểu thức expr là true

❖ Ví dụ:

`Loading`

❑ <element **ng-if**="expr">...</element>

❖ Bổ sung thẻ <element> nếu biểu thức expr là true

❖ Ví dụ:

`= 5">Passed`

CHỈ THỊ NG-IF, NG-SHOW, NG-HIDE

```
<div ng-app="">
  <div ng-controller="myctrl">
    <ul>
      <li>Họ và tên: {{student.name}}</li>
      <li>Giới tính:
        <span ng-show="student.gender">Nam</span>
        <span ng-hide="student.gender">Nữ</span>
      </li>
      <li>Xếp loại:
        <span ng-if="student.mark < 5">Yếu/kém</span>
        <span ng-if="student.mark >= 5 && student.mark < 6.5">Trung bình</span>
        <span ng-if="student.mark >= 7.5">Khá/giỏi</span>
      </li>
    </ul>
  </div>
</div>

<script>
  function myctrl($scope) {
    $scope.student = {
      name: "Phạm Đức Cường",
      mark: 7.5,
      gender: true
    };
  }
</script>
```



- Họ và tên: Phạm Đức Cường
- Giới tính: Nam
- Xếp loại: Khá/giỏi

```
<element ng-switch="expr">  
  <element ng-switch-when="value1"></element>  
  <element ng-switch-when="value2"></element>  
  ...  
  <element ng-switch-default></element>  
</element>
```

- ❑ Bổ sung vào trang web thẻ <element> có giá trị của chỉ thị **ng-switch-when** bằng với giá trị của chỉ thị **ng-switch**. Nếu không có <element> nào có ng-switch-when bằng với ng-switch thì bổ sung <element> chứa chỉ thị **ng-switch-default**

```
<ul ng-app="" ng-controller="myctrl">
  <li>Họ và tên: {{student.name}}</li>
  <li>
    Giới tính:
    <span ng-switch="student.gender">
      <span ng-switch-when="true">Nam</span>
      <span ng-switch-default>Nữ</span>
    </span>
  </li>
  <li>
    Ngành học:
    <span ng-switch="student.major">
      <span ng-switch-when="UDPM">Ứng dụng phần mềm</span>
      <span ng-switch-when="TKW">Thiết kế trang web</span>
      <span ng-switch-default>Ngành mới</span>
    </span>
  </li>
  <li>Điểm TB: {{student.mark}}</li>
</ul>
```

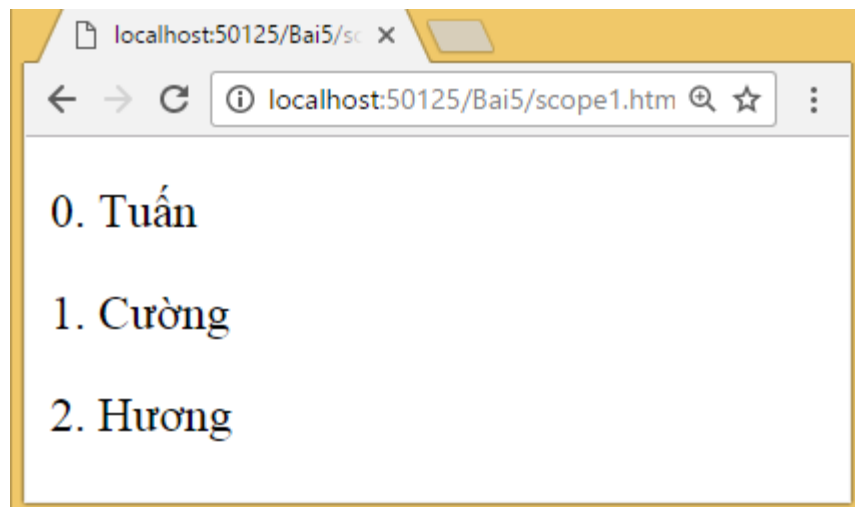
- Họ và tên: Phạm Đức Cường
- Giới tính: Nam
- Ngành học: Ứng dụng phần mềm
- Điểm TB: 7.5



```
<script>
  function myctrl($scope) {
    $scope.student = {
      name: "Phạm Đức Cường",
      mark: 7.5,
      gender: true,
      major: "UDPM"
    };
  }
</script>
```

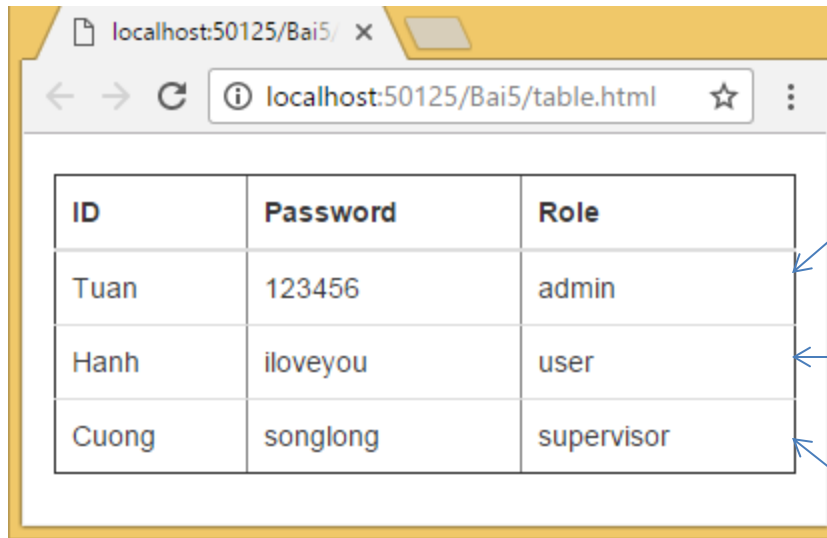
```
<element ng-repeat="item in array">{{$index}}. {{$item}}</element>
```

- ❑ Chỉ thị ng-repeat được sử dụng để duyệt mảng.
- ❑ Ví dụ trên duyệt các phần tử trong mảng array.
Cứ mỗi phần tử sẽ bổ sung một thẻ <element>
vào trang web
 - ❖ **item** là biến chứa phần tử hiện tại
 - ❖ **\$index** chứa vị trí của phần tử hiện tại đang duyệt



```
<div ng-app="" ng-controller="myctrl">
  <p ng-repeat="sv in students">{{$index}}. {{$sv}}</p>
</div>
<script>
  function myctrl($scope) {
    $scope.students = ["Tuấn", "Cường", "Hương"];
  }
</script>
```


❑ Sử dụng ng-repeat để tạo ra các hàng dữ liệu



ID	Password	Role
Tuan	123456	admin
Hanh	iloveyou	user
Cuong	songlong	supervisor

```
<tr ng-repeat="u in users">  
  <td>{{u.id}}</td>  
  <td>{{u.password}}</td>  
  <td>{{u.role}}</td>  
</tr>
```

```
$scope.users = [  
  {  
    id: "Tuan",  
    password: "123456",  
    role: "admin"  
  },  
  {  
    id: "Hanh",  
    password: "iloveyou",  
    role: "user"  
  },  
  {  
    id: "Cuong",  
    password: "songlong",  
    role: "supervisor"  
  }  
];
```

```
$scope.users = [
  {
    id: "Tuan",
    password: "123456",
    role: "admin"
  },
  {
    id: "Hanh",
    password: "iloveyou",
    role: "user"
  },
  {
    id: "Cuong",
    password: "songlong",
    role: "supervisor"
  }
];
```

```
<table border="1" class="table">
  <thead>
    <tr>
      <th>ID</th>
      <th>Password</th>
      <th>Role</th>
    </tr>
  </thead>
  <tbody>
    <tr ng-repeat="u in users">
      <td>{{u.id}}</td>
      <td>{{u.password}}</td>
      <td>{{u.role}}</td>
    </tr>
  </tbody>
</table>
```

Cứ 1 phần tử user
sẽ tạo 1 thẻ <tr>

- ❑ Cho `$scope.products=[{name:"", image:""}, {}, ...]`
- ❑ Hãy thảo luận cách tạo ra trang web sau

Swiss Watch



Sony Vaio Star



Korean Perfume



Motorola Modern



Canon Fast Flip



Dell 2 Flies





DEMO

Tạo trang sản phẩm đã
thảo luận ở slide trước



- ✓ Tìm hiểu \$scope
- ✓ \$scope lồng nhau
- ✓ Tìm hiểu \$rootScope
- ✓ Tìm hiểu các chỉ thị cơ bản
- ✓ Tìm hiểu các chỉ thị điều khiển giao diện
- ✓ Làm việc với bảng
- ✓ Tạo trang sản phẩm





Cảm ơn