

## GIỚI THIỆU TỔNG QUAN

Cùng với sự phát triển như vũ bão của cộng đồng JavaScript, chúng ta có cảm giác như mỗi tuần lại có thêm các framework, library JavaScript mới được ra mắt. Thật khó có thể để liệt kê hết ra đây các framework, library, tool hiện có trong cộng đồng. Bài viết này chỉ giới thiệu với các bạn những framework, library, tool nổi tiếng nhất và có ảnh hưởng nhất trong cộng đồng JavaScript, cũng như khi nào nên áp dụng các framework, library hay tool đó.

### 1. ANGULARJS

AngularJS là framework phổ biến nhất, được nhiều developer sử dụng để xây dựng phần front-end cho những web application có phần front-end phức tạp. Nó là một opensource được hỗ trợ bởi Google.

AngularJS là một MVC framework. Nó cung cấp two-way data binding giữa model và view. Cách binding dữ liệu như thế này cho phép tự động update ở cả hai phía bất kể khi nào dữ liệu bị thay đổi. Vì thế hỗ trợ các bạn xây dựng các thành phần view có thể tái sử dụng ở nhiều chỗ. Ngoài ra nó cũng cung cấp cho các bạn một services framework để hỗ trợ tối đa giao tiếp giữa back-end và front-end.

Khi nào thì nên sử dụng?

Bất kể khi nào bạn cần xây dựng một web application có phần front-end phức tạp và cần sử dụng một framework có thể handle tất cả mọi thứ.

### 2. REACTJS

Đây có lẽ là thứ tốn giấy mực nhiều nhất trong thời gian gần đây. Tất cả mọi người ai ai cũng nói về ReactJS. ReactJS là một opensource được phát triển bởi Facebook. Bản thân nó là một JavaScript library dùng để xây dựng user interface.

React cung cấp một phương thức dễ dàng để xây dựng giao diện thông qua việc phát triển các component và kết hợp chúng lại với nhau. Nó cũng cung cấp một phương thức trừu tượng hoá

DOM để tối ưu việc rendering cũng như cho phép có thể render thông qua Node.js. Một điểm nữa đó là nó implement one-way reactive data flow, giúp cho chúng ta có một cái nhìn dễ dàng hơn.

Trong một số trường hợp, người ta thường kết hợp React để làm view layer trong một số MVC framework như Angular, Ember.

Khi nào thì nên sử dụng?

Khi bạn muốn một tầng giao diện mạnh mẽ nhưng không cần framework nào ở tầng front-end hay đơn giản bạn muốn xây dựng một view layer mạnh mẽ hơn thứ có sẵn của các framework đang sử dụng như Angular, Ember hay Backbone hoặc khi bạn muốn xây dựng một isomorphic web framework.

### **3. BACKBONEJS**

BackboneJS nổi tiếng là một framework đơn giản gói gọn trong một file JavaScript. Nó cực kì phổ biến đối với các web application nhỏ.

BackboneJS cung cấp một MVC framework đầy đủ với cả phần routing. Phần model trong BackboneJS cho phép key-value binding và sử dụng event để handle khi dữ liệu thay đổi. Phần view đảm nhận việc handle các event và router dùng để handle URL cũng như trạng thái trong application. Quá đầy đủ cho bạn để xây dựng một Single Page Application mà không cần phải apply quá nhiều thứ phức tạp.

Khi nào thì nên sử dụng?

Backbone sẽ là một framework cực kì tốt để đi với những web application đơn giản.

### **4. EMBERJS**

EmberJS là một framework chú trọng vào hiệu quả của developer khi xây dựng web application. Bản thân Ember mô tả nó là một framework không làm lãng phí thời gian của bạn.

Ember là một MVC framework. Nó bao gồm template và view engine, tự động update khi data thay đổi. Nó mang concept tương tự như Angular, cho phép bạn tạo ra những HTML tag của riêng mình. Ngoài ra nó cũng bao gồm routing để handle URL, model engine để làm việc với RESTful API.

Khi nào thì nên sử dụng?

Khi bạn cần một framework có thể hoạt động và không cần phải quá linh hoạt bởi vì bạn bị giới hạn về thời gian.

## **5. METEOR**

Meteor là một fullstack framework, hay cũng có thể nói là một platform khá nổi tiếng trong thời gian gần đây. Nó hỗ trợ bạn xây dựng một web application hoàn chỉnh từ back-end, front-end, đến cả mobile app sử dụng cross platform. Hỗ trợ bạn deploy một cách nhanh chóng bằng nền tảng PaaS của nó. Đặc biệt Meteor hỗ trợ cực kì tốt đối với các real time web application. Hỗ trợ tích hợp tốt với các framework khác như AngularJS hay ReactJS ở front-end.

Khi nào thì nên sử dụng?

Như giới thiệu, khi bạn cần một fullstack framework, hỗ trợ việc deploy. Đặc biệt khi bạn xây dựng những real time web application.

## **6. SAILJS**

SailJS là một web framework phổ biến trong các NodeJS framework. Nó đưa ra một loạt các tính năng tuyệt vời để xây dựng web application với solid structure. Nó hỗ trợ convention over configuration (bạn đặt code trong folder /view thì nó là view, hỗ trợ websocket built-in, hỗ trợ generate RESTful API, hỗ trợ ORM với waterline).

Khi nào thì nên sử dụng?

Khi bạn cần một web framework hỗ trợ bạn mọi thứ nhanh nhất ở phía server side sử dụng NodeJS.

## 7. JQUERY

Có lẽ nó chính là JavaScript library được sử dụng phổ biến nhất trên thế giới. Hỗ trợ DOM traversal, event handling, animation, ajax,...

Khi nào thì nên sử dụng?

Luôn luôn.

## 8. LODASH

lodash là một JavaScript library cung cấp các tiện ích dành cho developer để đơn giản hoá code, cũng như code hiệu quả hơn.

Khi nào thì nên sử dụng?

Khi bạn cần một library đưa ra các tiện ích giúp việc coding hiệu quả hơn.

## 9. KARMA & MOCHA & CHAI

Mocha là một JavaScript test framework hỗ trợ việc test các phần async code một cách dễ dàng.

Chai là một behavior-driven/test-driven development assertion library, dùng kết hợp với Mocha để viết test script dễ đọc hơn.

Karma là tool được thiết kế để giúp tự động chạy test trên nhiều browser khác nhau.

Khi nào thì nên sử dụng?

Luôn luôn.

## 10. GRUNT & GULP

Khi xây dựng web application, trên môi trường production bạn thường phải minify JavaScript, CSS hay trước đó bạn phải biên dịch CoffeeScript/TypeScript, chạy unit test,... Đó là lý do vì sao bạn cần những task runner như Grunt hay Gulp.

Khi nào thì nên sử dụng?

Khi bạn muốn mọi thứ chỉ thông qua config và task runner thực hiện cho bạn.

## 11. BABEL

ES6 hay ECMAScript 2015 ra đời cùng hàng loạt các tính năng vô cùng tuyệt vời. Tuy nhiên chỉ những version mới nhất của browser mới hỗ trợ được và chúng ta không ở trong một thế giới tuyệt vời đến thế. Vẫn còn rất nhiều browser thế hệ cũ chỉ hỗ trợ được tới ES5. Vì thế chúng ta cần một công cụ để biên dịch ES6 về ES5 giúp web application của chúng ta có thể chạy được trên tất cả các browser.

Khi nào thì nên sử dụng?

Khi bạn muốn sử dụng các tính năng mới của Javascript và vẫn muốn support các browser thế hệ cũ.

## TÓM LẠI

Trên đây là những framework, library, tool nổi bật và phổ biến nhất trong JavaScript cũng như trường hợp nên sử dụng. Hy vọng bài viết này đem lại cho bạn cái nhìn sơ bộ về các framework, library hay tool trong cộng đồng JavaScript và hiểu được khi nào nên áp dụng chúng.

## **I. TỔNG QUAN ANGULARJS**

### **1. ANGULARJS LÀ GÌ?**

- Angular là một bộ Javascript Framework rất mạnh và thường được sử dụng để xây dựng project Single Page Application (SPA).
- Nó hoạt động dựa trên các thuộc tính mở rộng HTML (các attributes theo quy tắc của Angular). Đây là một Framework mã nguồn mở hoàn toàn miễn phí và được hàng ngàn các lập trình viên trên thế giới ưa chuộng và sử dụng.
- Framework này được thế hệ Web 2.0 phát triển khá mạnh ở nước ngoài, tuy nhiên ở Việt Nam thì vẫn chưa thông dụng lắm.

### **2. NHỮNG AI CẦN NẮM BẮT ANGULARJS**

- Về thành phần đối tượng học serie AngularJS là những bạn muốn đi theo nghề lập trình web nói chung và những bạn muốn sử dụng AngularJS để làm dự án nói riêng.
- Hy vọng rằng sau loạt serie này bạn sẽ tự mình nghiên cứu thêm nhé, vì bản chất học lập trình không thể nào trình bày full được mà chỉ ở mức basic, chỉ đường cho bạn đi thôi.

### **3. BẠN CẦN CHUẨN BỊ NHỮNG KIẾN THỨC GÌ ĐỂ HỌC ANGULARJS**

- Để học được Angular thì bạn phải có kiến thức cơ bản về javascript, biết cách xử lý object, string, ... Hay thậm chí nếu như bạn biết chuyên sâu Javascript thì đó là một lợi thế rất nhiều.
- Vì bản chất Angular hoạt động dạng Single Page, sử dụng dạng API để lấy dữ liệu, vì vậy bạn phải biết các kỹ thuật xử lý DHTML, Ajax.

### **4. CÁC ĐẶC TÍNH CỦA ANGULARJS**

- AngularJS là một Framework phát triển dựa trên Javascript để tạo các ứng dụng web
- AngularJS thường dùng để phát triển frontend (giao diện khách hàng) thông qua các API để gọi data, sử dụng mô hình MVC rất mạnh mẽ
- Mã nguồn AngularJS tự động fix với các trình duyệt khác nhau nên bạn không cần phải lo vấn đề tương thích trình duyệt
- Angular là mã nguồn mở, hoàn toàn miễn phí và được phát triển bởi hàng ngàn các lập trình viên trên thế giới.

## 5. CÁC TÍNH NĂNG CODE LỖI CỦA ANGULARJS

- Sau đây là các tính năng cốt lõi quan trọng trong AngularJS
  - **Data-binding:** (liên kết dữ liệu) tự động đồng bộ dữ liệu giữa model và view
  - **Scope:** (Phạm vi) Đây là những đối tượng kết nối giữa Controller và View
  - **Controller:** Đây là những hàm javascript xử lý kết hợp với bộ điều khiển Scope
  - **Service:** AngularJS sử dụng các API được xây dựng từ các web service (PHP, ASP) để thao tác với DB.
  - **Filters:** Bộ lọc lọc ra các thành phần của một mảng và trả về mảng mới
  - **Directives:** đánh dấu vào các yếu tố của DOM, nghĩa là sẽ tạo ra các thẻ HTML tùy chỉnh
  - **Templates:** hiển thị thông tin từ controller, đây là một thành phần của views
  - **Routing:** chuyển đổi giữa các action trong controller
  - **MVC:** Mô hình chia thành phần riêng biệt thành Model, View, Controller. Đây là một mô hình khá hay nhưng trong Angular thì nó được chế biến lại một chút gần giống với MVVM (Model View View Model)
  - **Deep Linking:** Liên kết sâu, cho phép bạn mã hóa trạng thái của ứng dụng trong các URL để nó có thể đánh dấu được với công cụ tìm kiếm.
  - **Dependency Injection:** Angular giúp các nhà phát triển tạo ứng dụng dễ dàng hơn để phát triển, hiểu và thử nghiệm dễ dàng.

## 6. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA ANGULARJS

- Ưu điểm:
  - Angular cho phép tạo ra các ứng dụng một cách đơn giản, code sạch
  - Angular sử dụng data bind giống .NET với tính năng liên kết với HTML nên giúp người dùng cảm thấy dễ chịu.
  - Angular đang ở giai đoạn thử nghiệm
  - Angular có thể chạy trên hầu hết các trình duyệt điện thoại thông minh.
- Nhược điểm:

- Không an toàn: Được phát triển từ javascript nên nó không an toàn, phía máy chủ phải thường xuyên xác nhận quyền để hệ thống chạy trơn tru.
- Phụ thuộc: Nếu người dùng vô hiệu hóa javascript thì coi như ko dung được

## 7. CÁC COMPONENTS CHÍNH TRONG ANGULARJS

- Angular JS được chia làm ba thành phần chính sau đây:
  - ng-app: định nghĩa này chỉ thị một kết nối ứng dụng Angular JS tới HTML
  - ng-model: chỉ thị này liên kết với dữ liệu của ứng dụng Angular
  - ng-bind: chỉ thị này dùng đưa dữ liệu vào HTML tags

## II. CÀI ĐẶT ANGULARJS

- Có 2 cách để cài đặt thư viện AngularJS vào ứng dụng của bạn :

### 1. Import trực tiếp từ đường dẫn đến thư viện angularjs, bằng cách bỏ vào thẻ <head> như sau :

<head>

<script src =<https://ajax.googleapis.com/ajax/libs/angularjs/1.3.3/angular.min.js>>

</head>

- Cách này nhược điểm là không có internet thì không chạy nếu các bạn chỉ thực hành tại localhost, hoặc mang máy ra quán cà phê thư giãn và học AngularJS mà wifi yếu quá =)).

### 2. Truy cập vào <https://angularjs.org/> và download thư viện cho ứng dụng của bạn, theo tôi các bạn nên download về bản mới nhất và là bản release, không nên down bản thử nghiệm vì có thể gây ra chạy không đúng trong quá trình học của các bạn.





### 3. Các bạn có thể chọn lấy từ GitHub

- Download AngularJS 1, hoặc click vào button có hình Download AngularJS 1 như hình



- Chọn:

- Branch là latest version.
  - Build là Minified (đây là file đã nén giúp web chạy nhanh hơn, bởi vì chúng ta không cần đọc file thư viện và tuyệt đối không sửa đổi gì trong thư viện của nó, nên chọn bản nén cho có lợi performance).
  - CDN : nếu dùng cách import trực tiếp như đề cập ở trên.
  - Bower và npm : nếu cài đặt bằng terminal, hữu ích nếu bạn dùng Linux.
  - Rồi click Download tải thư viện về thôi.
- Nếu đã import thư viện bằng file download về thì các bạn cũng bỏ vào thẻ <head> và sửa đường dẫn src lại là đường dẫn trỏ đến file trong ứng dụng của bạn, nơi chứa file thư viện AngularJS.

### III. VIẾT ỨNG DỤNG

- Các Bước Tạo Ứng Dụng AngularJS:

=====

#### Step 1 – Load Framework

Cách nhanh là ta sử dụng thẻ script

```
<script src = "http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">
```

```
</script>
```

#### Step 2 – Định nghĩa nơi bắt đầu sử dụng AngularJS

```
<div ng-app= "">
```

```
...
```

```
</div>
```

### Step 3 – Định nghĩa tên model

```
<p>Enter your name: <input type="text" ng-model = "name"></p>
```

### Step 4 – Ràng buộc giá trị của model và hiển thị

```
<p>Hello <span ng-bind = "name"></span>!</p>
```

---

## Các bước tạo một ứng dụng AngularJS

### Bước 1: Tải framework

Là một khung JavaScript thuần, thư viện AngularJS có thể được thêm vào bởi sử dụng thẻ

<script>, theo một trong hai cách sau:

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"> </script>
```

Hoặc sau khi download file AngularJS:

```
<head>
```

```
  <script src="D:\angular.min.js"></script>
```

```
</head>
```

### Bước 2: Định nghĩa ứng dụng AngularJS sử dụng ng-app directive.

```
<div ng-app="">
```

```
...
```

```
</div>
```

### Bước 3: Định nghĩa tên một model sử dụng ng-model directive.

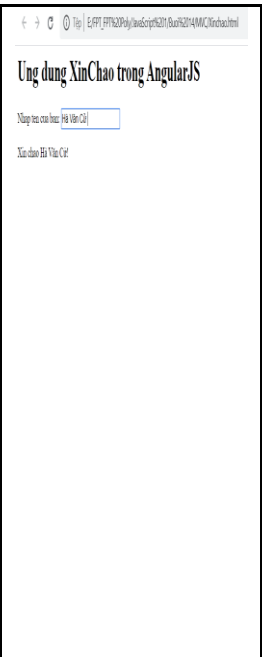
<p>Nhập tên của bạn: <input type="text" ng-model="ten"></p>

**Bước 4: Gắn kết giá trị của model đã được định nghĩa ở trên sử dụng ng-bind directive.**

<p>Xin chào <span ng-bind="ten"></span>!</p>

**Sử dụng các bước bên trên để tạo ra một trang HTML**

- Vidu\_Xinchao.html

<pre>&lt;html&gt; &lt;title&gt;Tao ung dung AngularJS dau tien&lt;/title&gt; &lt;body&gt; &lt;h1&gt;Ung dung XinChao trong AngularJS&lt;/h1&gt; &lt;div ng-app=""&gt;   &lt;p&gt;Nhập tên của bạn: &lt;input type="text" ng-model="ten"&gt;&lt;/p&gt;   &lt;p&gt;Xin chào &lt;span ng-bind="ten"&gt;&lt;/span&gt;!&lt;/p&gt; &lt;/div&gt; &lt;script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"&gt;&lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
--	--

**DIỄN GIẢI ANGULARJS TÍCH HỢP VỚI HTML:**

- ng-app chỉ bắt đầu một ứng dụng AngularJS

- ng-model sau đó tạo một biến model đặt tên là “name” mà có thể được sử dụng với trang html và bên trong thẻ div có chỉ thị ng-app. Tức là ng-model sẽ sử dụng được nếu nó nằm trong ng-app
- ng-bind thì sử dụng tên của model để hiển thị trong thẻ html bất kể nơi nào khi user gõ vào trong text box.
- Đóng </div> cũng là kết thúc ứng dụng AngularJS. Nếu sau </div> bạn gọi ng-bind thì sẽ không có tác dụng vì nó nằm ngoài ng-app rồi.

#### IV. ANGULARJS – DIRECTIVES

- AngularJS cho bạn mở rộng HTML với các thuộc tính mới gọi là Directives. Có vài thuộc tính đặc biệt bắt đầu bằng ng-, các directive này AngularJS đã xây dựng sẵn, ngoài ra AngularJS cũng cho bạn tự định nghĩa directives của riêng bạn.
- Chúng ta xem một số directives được xây dựng sẵn :
  - ng-app – directive: này bắt đầu một ứng dụng AngularJS
  - ng-init – directive : này khởi tạo dữ liệu
  - ng-model – directive : này định nghĩa model, đó là các biến để sử dụng trong AngularJS
  - ng-repeat – directive : này lặp lại các nhân tố html cho mỗi item trong một bộ. Nó xem như vòng lặp

#### - Giải thích các directive xây dựng sẵn:

##### 1. Giới thiệu ng-app directive trong AngularJS:

- ng-app directive bắt đầu một ứng dụng AngularJS. Nó tự động khởi tạo ứng dụng và khởi tạo ứng dụng web có sử dụng AngularJS. Ở đây, ta định nghĩa một ứng dụng AngularJS mặc định sử dụng thuộc tính ng-app của một phần tử div.

```
<div ng-app="">
...
</div>
```

## 2. Giới thiệu ng-init directive:

- ng-init khởi tạo các dữ liệu cho ứng dụng AngularJS. Nó được sử dụng để khởi tạo các giá trị cho một biến được sử dụng trong ứng dụng. Ở đây, ta định nghĩa một mảng các Quốc gia. Chúng ta sử dụng cú pháp JSON để định nghĩa cho mảng các Quốc gia.

```
<div ng-app="" ng-init="cacquocgia=[{locale:'en-VN',ten:'Cong Hoa XHCN Viet Nam'},
                                   {locale:'en-GB',ten:'Vuong Quoc Anh'},
                                   {locale:'en-FR',ten:'Cong Hoa Phap'}]">
...
</div>
```

## 3. Giới thiệu ng-model directive trong AngularJS:

- ng-model directive định nghĩa các model/biến được sử dụng trong ứng dụng AngularJS. Ở đây, chúng ta sẽ định nghĩa một model dưới tên "ten".

```
<div ng-app="">
...
    <p>Nhap ten cua ban: <input type="text" ng-model="ten"></p>
</div>
```

## 4. Giới thiệu ng-repeat directive trong AngularJS:

- ng-repeat directive lặp lại các phần tử HTML cho mỗi item trong một tập hợp. Ở đây, chúng ta sẽ duyệt qua một mảng các Quốc gia.

```
<div ng-app="">
```

```
...
```

```
<p>Danh sach cac Quoc gia voi Locale tuong ung:</p>
```

```
<ol>
```

```
<li ng-repeat="quocgia in cacquocgia">
```

```
{{ 'Quoc Gia: ' + quocgia.ten + ', Locale: ' + quocgia.locale }}
```

```
</li>
```

```
</ol>
```

```
</div>
```

## Vidudirective.html

```
<html>
```

```
<head>
```

```
<title>Directive trong AngularJS</title>
```

```
</head>
```

```
<body>
```

```
<h1>Tao ung dung AngularJS</h1>
```

```
<div ng-app = "" ng-init = "cacquocgia = [{locale:'en-VN',ten:'CH  
XHCN Viet Nam'}, {locale:'en-GB',ten:'Vuong Quoc Anh'}, {locale:'en-  
FR',ten:'Cong Hoa Phap'}]">
```

```
<p>Nhap ten cua ban: <input type = "text" ng-model = "ten"></p>
```

```
<p>Xin chao <span ng-bind = "ten"></span>!</p>
```

```
<p>Danh sach cac quoc gia va locale tuong ung:</p>
```

```
<ol>
```

```
<li ng-repeat = "quocgia in cacquocgia">
```



<pre>         {{ 'Quoc Gia: ' + quocgia.ten + ', Locale: ' + quocgia.locale }}       &lt;/li&gt;     &lt;/ol&gt;   &lt;/div&gt;    &lt;script src = "http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"&gt;&lt;/ script&gt; &lt;/body&gt; &lt;/html&gt; </pre>	
---	--

- Ví dụ trên khởi tạo một mảng các quốc gia. Chúng ta sẽ sử dụng cú pháp JSON để định nghĩa mảng các quốc gia. Nếu các bạn chưa biết JSON là gì thì có thể tham khảo <http://www.json.org/>.
- (**JSON** là viết tắt của **JavaScript Object Notation**, là một kiểu định dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được. **JSON** là một tiêu chuẩn mở để trao đổi dữ liệu trên web.)



# JSON

```
{  "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

## - Cú pháp JSON:

```
<div ng-app = "" ng-init = "countries = [{locale:'en-US',name:'United States'},
{locale:'en-GB',name:'United Kingdom'}, {locale:'en-FR',name:'France'}]">
```

...

```
</div>
```

- Và một điều lưu ý là để **bind dữ liệu**, liên hệ tới **directive ng-bind**. Một cách khác để làm điều này mà không cần sử dụng thẻ html là ta dùng ký hiệu 2 dấu ngoặc nhọn {{.
- Việc đổ dữ liệu trong {{ cũng như trong javascript, chuỗi sẽ nằm trong dấu nháy và muốn in ra giá trị của biến có kết hợp chuỗi thì tao dùng dấu +, như sau:

```
{{ 'Country: ' + country.name + ', Locale: ' + country.locale }}
```

## V. ANGULARJS – EXPRESSIONS

- Expression được sử dụng để buộc dữ liệu ứng dụng tới html. Expression được ghi bên trong 2 dấu ngoặc móc {{ express }}. Expression có cùng cách làm việc như ng-bind directives.
- Expression trong AngularJS là một biểu thức javascript và đưa ra dữ liệu nơi chúng được sử dụng.

### Sử dụng số

<p> Chi phí quyền sách: {{cost \* quantity}} \$ </p>

### Sử dụng chuỗi

<p> Hello {{firstname + “ “ + lastname}}!</p>

### Sử dụng đối tượng

<p>ID: {{student.id}}</p>

### Sử dụng mảng

<p>Mark(match): {{mark[3]}}</p>

### Mô tả:

- Sử dụng số trong Expression trong AngularJS

<p>Chi phi muon sach: {{gia \* soluong}}.000 VND</p>

- Sử dụng chuỗi trong Expression trong AngularJS

<p>Xin chao {{sinhvien.ho + " " + sinhvien.ten}}!</p>

- Sử dụng đối tượng trong Expression trong AngularJS

<p>MSSV: {{sinhvien.mssv}}</p>

- Sử dụng mảng trong Expression trong AngularJS

<p>Diem thi (Mon Toan): {{diemthi[3]}}</p>

### - **ViduExpression.html**

```
<html>
  <title>Vi du Expression trong AngularJS</title>
<body>
  <h1>Ung dung AngularJS</h1>
  <div ng-app="" ng-init="soluong=2;gia=40;sinhvien={ho:'Trần Anh
',ten:'Dũng',mssv:123456789};diemthi=[8.5,9,7.5,9,9.5]">
    <p>Xin chao {{sinhvien.ho + " " + sinhvien.ten}}!</p>
    <p>MSSV: {{sinhvien.mssv}}</p>
    <p>Chi phi muon sach: {{gia * soluong}}.000 VND</p>
    <p>Diem thi (Mon Toan): {{diemthi[3]}}</p>
  </div>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
</body>
</html>
```

# Ung dung AngularJS

Xin chao Trần Anh Dũng!

MSSV: 123456789

Chi phi muon sach: 80.000 VND

Diem thi (Mon Toan): 9

## Ghi chú: So sánh giữa biểu thức AngularJS và Javascript

- Giống như biểu thức Javascript, AngularJS có thể chứa đựng chuỗi, toán tử, và biến.
- Không giống như Javascript, biểu thức AngularJS có thể ghi bên trong HTML.
- AngularJS expression không hỗ trợ điều kiện, lặp, và xử lý ngoại lệ, trong khi Javascript làm được điều này.
- AngularJS expression hỗ trợ filter, còn Javascript thì không.