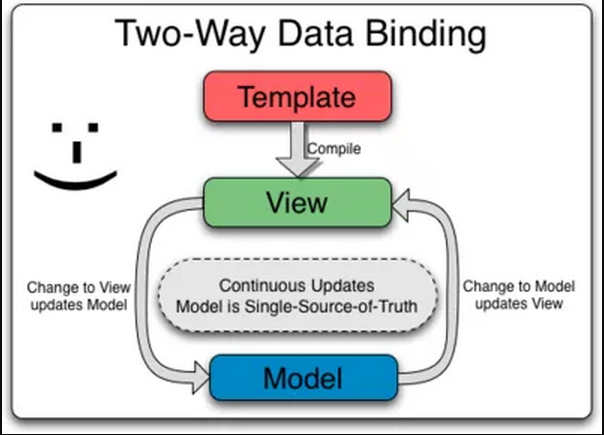
**Data-binding**: (liên kết dữ liệu) tự động đồng bộ dữ liệu giữa model và view

Data Binding trong AngularJs chính là các thức tự động đồng bộ (synchronization) dữ liệu giữa Model và View trong AngularJs. Data Binding cho phép bạn tạo sự liên kết, ràng buộc giữa Model và Template thông qua một ứng dụng AngularJs (ng-app). Khi có bất kỳ sự thay đổi nào ở model, dữ liệu sẽ được phản ánh ngay tức thì ra Template và ngược lại.

Cách thức hoạt động của Data Binding trong AngularJs



Ở hình trên bạn sẽ thấy rằng, đầu tiên AngularJS compile Template ra ngoài View cho người dùng thấy được (chẳng hạn như là 1 ô textbox). Khi có bất kì thày đổi trên View này, chẳng hạn như trong ô textbox mình sẽ nhập là abc. Thì tự động dữ liệu abc này sẽ ngay lập tự được đồng bộ vào bên trong Model. Và tiếp theo bên trong Model nếu có bất kỳ sự thay đổi nào sẽ ngay lập tức được đồng bộ ngược lại ra View. Vì thế dữ liệu được trung chuyển từ Model sang View hay từ View sang Model là một thể thống nhất.

**Scope**: (Phạm vi) Đây là những đối tượng kết nối giữa Controller và View

Scope (biến phạm vi $scope) là một đối tượng trung gian mà thông qua nó các thuộc tính hay phương thức bên trong Controller được truyền tải ra ngoài View. Chính vì thế chỉ có những thuộc tính và phương thức sử dụng $scope mới có thể tương tác với lớp View của một ứng dụng AngularJs.

Một ví dụ dễ hiểu đó là các bạn hãy tưởng tượng $scope là một chiếc xe vận chuyển, lớp Controller là người quản lý nhà kho, còn View là nơi nhận hàng. Người quản lý nhà kho sẽ thực hiện việc chuyển giao hàng hóa tới nơi nhận hàng thì phải thông qua chiếc xe vận chuyển. Và trong ứng dụng AngularJs cũng vậy, $scope như một chiếc xe tải vận chuyển các phương thức, thuộc tính bên trong Controller và gửi các thuộc tính, phương thức này ra bên ngoài View.

**Controller**: Đây là những hàm javascript xử lý kết hợp với bộ điều khiển Scope

Trong AngularJs, Controller giống như một hàm contructor của Javascript, dùng để khởi tạo các thuộc tính, phương thức cần thiết và gửi ra ngoài view thông qua đối tượng phạm vi $scope.

Khi chúng ta tạo ra 1 controller và đính kèm nó vào DOM của HTML (có thể là thẻ html, body, div, span,…) thông qua ng-controller directives.  AngularJs sẽ khởi tạo một đối tượng Controller sử dụng tất cả các thuộc tính và phương thức của Controller đã được tạo ra đó thông qua đối tượng phạm vi $scope, từ đối tượng $scope này. Toàn bộ thuộc tính (biến js) hoặc phương thức (function) sẽ được bind ra ngoài View tùy theo cách mà chúng ta xử lý.

### Một số lưu ý khi sử dụng Controller trong AngularJs

* Không nên sử dụng Controller để thao tác DOM của HTML. Controller chỉ nên chứa các business logic.
* Không nên sử dụng Controller để xử lý dữ liệu đầu vào, vì điều này sẽ làm rối rắm Controller của bạn. Thay vì đó bạn có thể sử dụng form controls trong AngularJs để làm điều đó
* Không nên sử dụng Controller để xử lý dữ liệu đầu ra. Thay vì đó bạn có thể sử dụng cơ chế filters của AngularJs
* Không nên sử dụng Controller để chia sẽ các đoạn mã (code) với các Controller khác. Thay vì đó bạn hãy sử dụng cơ chế service trong AngularJs
* Không nên sử dụng Controller để quản lý vòng đời (life cycle) của các bộ điều khiển (components). Thay vì đó hãy sử dụng cơ chế service trong AngularJs

Tham khảo: <http://freetuts.net/angular>