



## MVC 디자인 패턴을 통한 angularjs 기초 다지기

### Table of Contents

- MVC 디자인 패턴을 통한 angularjs 기초 다지기
  - MVC 패턴에 관한 간단한 예제
  - 지시어 - ngpp 지시어
  - 뷰 - View
  - 모델 - Model
  - 컨트롤러 - Controller
  - 최종 요약 정리

## MVC 패턴에 관한 간단한 예제

우선 아래와 같은 간단한 예제를 작성해봅시다.

```
1 <html ng-app>
2 <head>
3 <title>AngularJS MVC 디자인 패턴 구현 및 구조 파악</title>
4 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.10/angular.min.js"></script>
5 <script src="controller.js"></script>
6 </head>
7 <body ng-controller="StudyMemberListCtrl">
8 <ul>
9 <li ng-repeat="member in members">
10 {{member.name}}
11 <p>{{member.message}}</p>
12 </li>
13 </ul>
14 </body>
15 </html>
```

자, 다음 Controller.js 파일을 아래와 같이 작성해보자.

```
1 function StudyMemberListCtrl($scope){
2 $scope.members = [
3 {"name":"Groovy","message":"I love Groovy"},
4 {"name":"Grails","message":"I love Grails"},
5 {"name":"Grandle","message":"I love Grandle"}
6 ];
7 }
```

-결과 화면-



## 지시어 - ng-app 지시어

위에 코드 예제에서 처음으로 확인해야 할 코드는 바로 ng-app 라는 지시어이다.

ng-app 지시어를 사용하여 Angularjs 프레임워크가 어떤 위치로 부터 DOM을 컴파일 해야하는지 알려주는 역할을 한다고 볼 수 있다.

현재 위 코드에서 보면 어서 지시어를 작성하였으므로, angularjs의 스코프 범위는 문서 전체가 된다.

그 후 Angularjs 라이브러리가 로딩 된 후 2번째 위치에서 controllerjs 및 다른 가티 js 파일을 로드하게 된다.

controller.js 파일안에는 ng-controller에서 명시한 StudyMemberListCtrl 함수가 포함되어 있다.



1. Controller를 외부JS 파일로 제외하여 작성한 것을 집중하여 보기 바란다.
2. 물론! controller.js 의 내용을 외부js로 분리하지 않고 내부에 작성해도 정상적으로 작동한다!

## 뷰 - View

```
<body ng-controller="StudyMemberListCtrl">
  <ul>
    <li ng-repeat="member in members">
      {{member.name}}
      <p>{{member.message}}</p>
    </li>
  </ul>
</body>
```

Angularjs에서 View(뷰)는 Data Model(모델)의 값들이 브라우저를 통해서 HTML 파싱을 통해서 보여진 결과 값이라 볼 수 있다. 즉, 언제든지 모델이 변화하면 Angularjs는 미리 연결된 요소를 즉시 반영하여 View를 업데이트 한다.



- 1.ng-repeat="phone in phones" 문은 컨트롤러에서 설명하도록 한다.
- 2.for(var i in array) 문법이란 비슷한 용도로 알아두면 좋다.

## 모델 - Model

```
1 members = [  
2   {"name":"Groovy","message":"I love Groovy"},  
3   {"name":"Grails","message":"I love Grails"},  
4   {"name":"Grandle","message":"I love Grandle"}  
5 ];
```

여기에서의 모델은 단순히 스터디 과목 이름의 배열로 표시를 해보았다.



위 모델들은 StudyMemberListCtrl 컨트롤러에 의해 제어되는 것을 알 수 있다.  
아직 이해가 안된다면, 차근차근 공부해 보기로 한다.

## 컨트롤러 - Controller

```
function StudyMemberListCtrl($scope){  
    $scope.members = [  
        {"name":"Groovy","message":"I love Groovy"},  
        {"name":"Grails","message":"I love Grails"},  
        {"name":"Grandle","message":"I love Grandle"}  
    ];  
}
```

컨트롤러는 AngularJs에서 가장 중요한 부분이며, 하는 역할은 바로 뷰와 모델을 상호 연결하여 데이터 바인딩을 설정해 주는 작업을 하는 것이다.

위의 예제를 통해서 정리해보면,

StudyMemberListCtrl - 컨트롤러 함수의 이름이며, 별도의 외부 JS파일에 위치하고 있다.

Body 태그 내에 ng-controller 지시어를 통해서 제작한 컨트롤러의 함수의 이름과 매치시킨다.

그 후 작동되는 순서는 아래와 같다.

- 1.스터디 모델 데이터는 컨트롤러 Scope 영역에 주입되게 된다.  
-> scope는 어플리케이션이 시작 되는 순간 만들어진 Root Scope를 상속받은 형태이다.
- 2.컨트롤러 Scope는 body ng-controller="컨트롤러 이름" 과 같이 설정하여 사용한다.  
-> 해당 영역에서 {{ }} 문자를 통해서 모델과 Binding 하여 사용 가능하다.

## 최종 요약 정리

- 1.AngularJs파일 로드 -> HTML 페이지 로드 -> **브라우저에 의해 콜백함수 실행** -> ng-app 지시어 찾을  
-> ng-app 지시어가 정의된 요소에 **부트스트랩 활성화(Scope)**
- 2.AngularJs에서 가장 중요한 개념은 바로 **Scope** 라고 볼 수 있다.  
-> **모델, 뷰, 컨트롤러가 서로 연동**되게 하는 중요한 요소라고 볼 수 있다.
- 3.AngularJs는 모델을 뷰와 분리하지만, Sync 는 유지한채로 있지만, 모델에서 일어나는 **변화를 감지하여 즉시 뷰에 반영하도록 해주는 기능이 있다.**



만약 Javascript 또는 jQuery 를 활용하여 위의 예제랑 같은 페이지를 제작한다고 가정하고, 필요한 메소드 및 코딩 방법에 대해서 생각해보면 확연히 줄어든 코딩 라인, 아올러 사용법에 있어서 AngularJs가 가지고 있는 유니크한 장점을 확인할 수 있다.