

■ <http://stats.js.org/>

1) 현재 github에서 가장 인기있는 javascript 라이브러리의 순위



RANK	NAME	DAY	WEEK	MONTH
0000	angular.js	5	20	113
0001	d3	2	12	60
0002	node	-3	-11	-30
0003	jquery	2	-1	-8
0004	html5-boilerplate	-2	-12	-33
0005	reveal.js	-1	2	19
0006	impress.js	-1	-7	-31
0007	react	8	33	140
0008	three.js	1	2	7
0009	backbone	-1	-5	-22
0010	meteor	2	11	68
0011	brackets	1	-3	-9
0012	jQuery-File-Upload	1	-3	-2
0013	javascript	3	18	98
0014	todomvc	1	6	35
0015	express	0	2	7
0016	moment	2	5	15
0017	socket.io	4	10	59
0018	Chart.js	3	7	32
0019	node-js-sample	-2	-8	-43
0020	atom	1	6	43
0021	frontend-nanodegree-resume	7	42	175
0022	ionic	2	11	69
0023	Ghost	3	4	13
0024	Semantic-UI	4	12	47
0025	underscore	0	0	7

2) angular가 1위

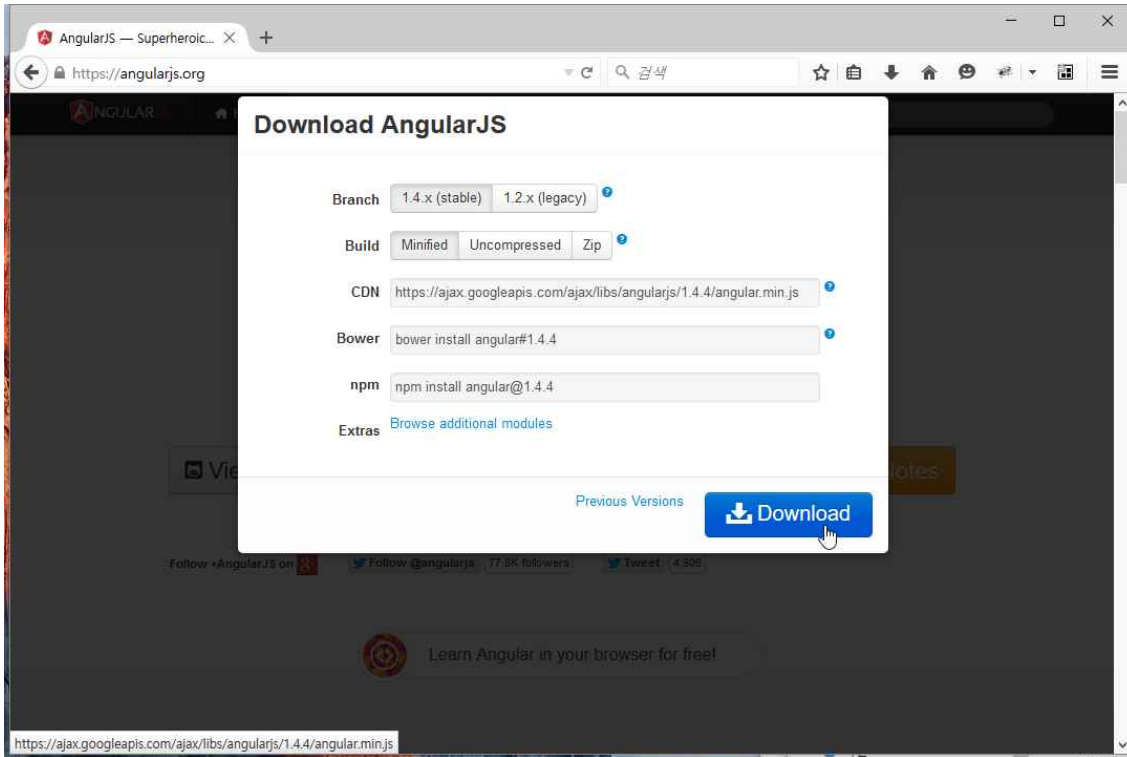
3) ionic이 22위(하이브리드 프레임워크로는 1위)

■ **angular.js**

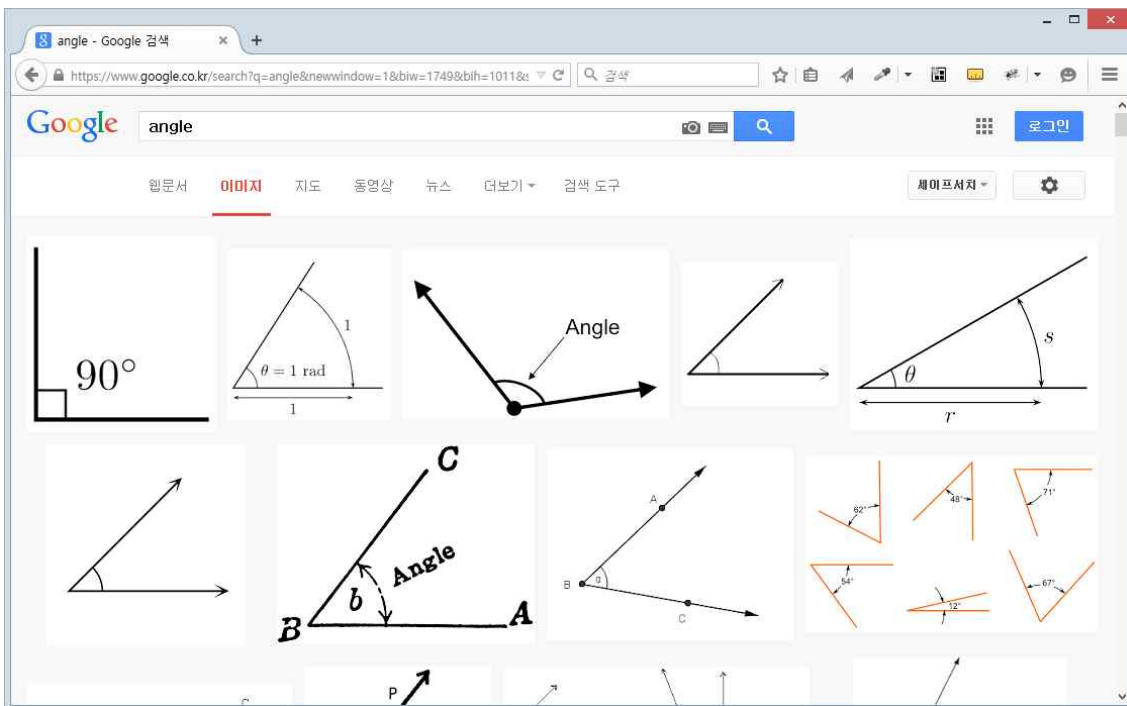
1) angularjs.org



## 2) angularjs의 다운로드



## 3) angular란? 각진이라는 뜻으로 각도기 즉, '∠' 이 태그를 의미함



## 4) 'ng-' 지시어를 잘 알아야 함

## 5) ng-app 속성을 부여하면 그 태그안에서 angular.js가 실행됨

## 왜 ng인가?

홈페이지의 fag를 보면

'Because HTML has Angular brackets and "ng" sounds like "Angular".'라고 쓰여있음.  
한글로 해석하면 '<'태그의 발음이 ng와 비슷하다고 함

## 4) 웹표준을 위하여 'data-'를 사용함

## ■ angular.js의 특징

### 1) 양방향 데이터 바인딩

#### - 양방향 바인딩의 장점

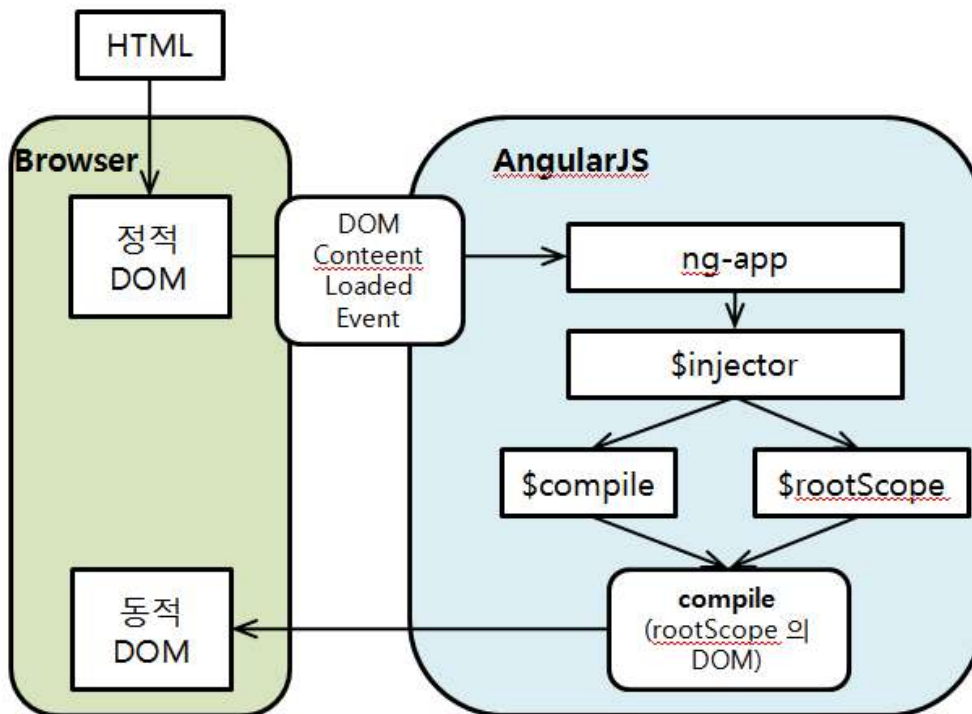


#### - 양방향 바인딩이 아닌 경우 일일이 페이지 내부를 모두 변경해야 함

### 2) MVC 구조(MVW : Model View Whatever)

- 3) 지시자(directive)를 이용한 HTML 확장
- 4) 의존관계 주입(DI)
- 5) 단일페이지 웹 애플리케이션을 위한 라우터

## ■ angular.js의 작동방식



- 1) 브라우저가 html을 로드(DOM을 파싱)
- 2) Angular.js를 로드
- 3) DOM Content Loaded Event를 기다림
- 4) DOM이 모두 로드되면 ng-app 지시자를 찾음
- 5) ng-app에서는 dependency injection 을 위해 사용되는 \$injector를 생성
- 6) injector 지시어는 어플리케이션의 모델을 위한 컨텍스트가 되는 루트 스코프 생성
- 7) 최종적으로 ng-app을 기준으로 하위DOM을 컴파일하고 rootScope와 링크

## ■ jquery와의 비교

## 1) input에 있는 글자를 h1요소에 출력하는 예제

```
<div id="txtBox">
  <input type="text" />
  <h1>안녕하세요?</h1>
</div>
<script src="js/zepto.js"></script>
<script>

  $("input").keyup(function() {
    var txt = $(this).val();

    $("h1").text(txt);
  });

</script>
```

## ■ angular.js의 시작

```
<!DOCTYPE html>
<html lang="ko" data-ng-app>
<head>
  <meta charset="UTF-8">
  <title>Angular.js 예제</title>
</head>
<body>
</body>
</html>
```

1) data-ng-app속성이 있어야 angular.js가 작동

2) 책이나 예제소스에 있는 ng-app과 같음(웹표준때문에 data-ng-app을 추천)

'ng-' 지시어(directive)의 이름 표기법

예) ng-model 을,

ng-model,ng:model,ng\_model,x-ng-model,x-ng:model,x\_ng:model,data-ng-model,data-ng:model,data-ng\_model 등으로 전부 사용할 수 있음

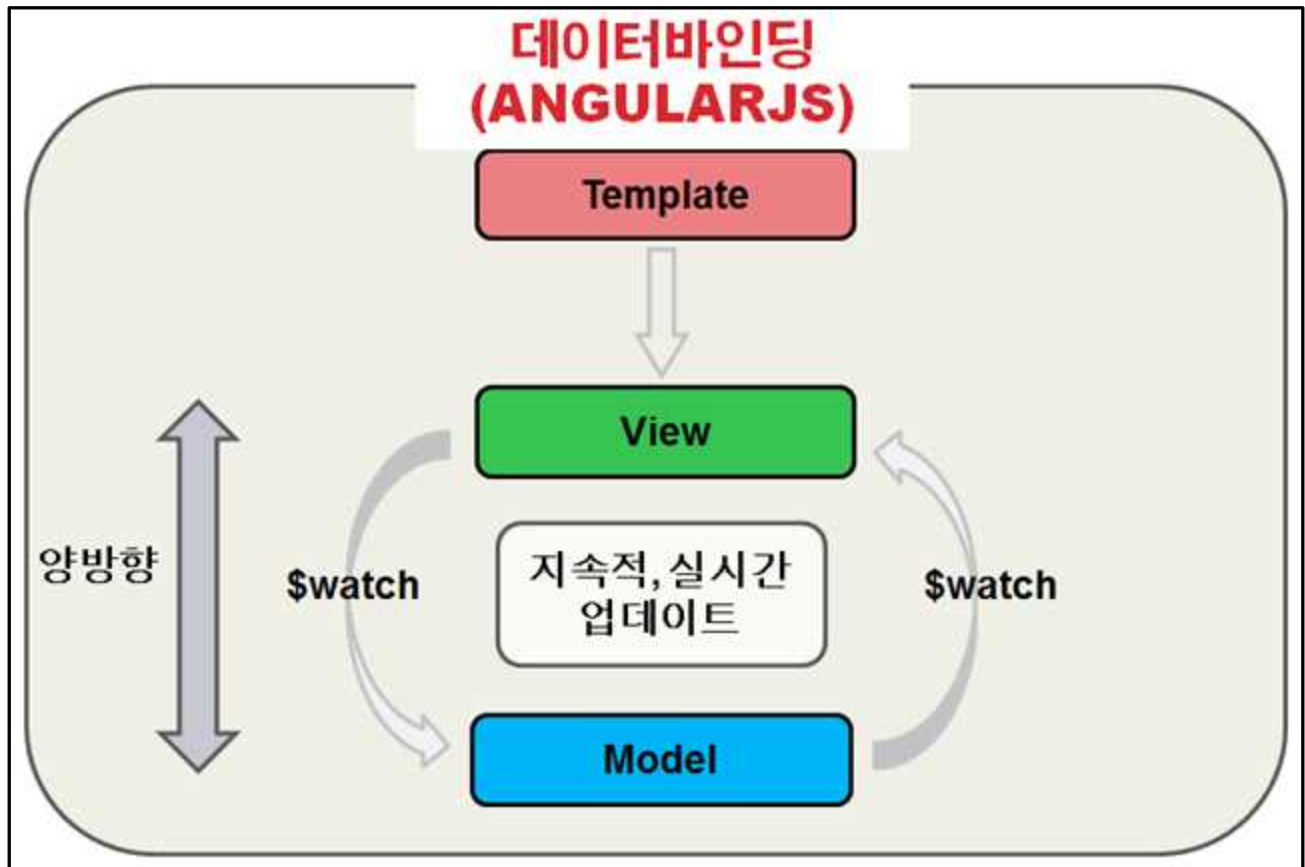
추천 :data-ng-model (HTML5표준)

## ■ angular.js의 완전기초

1) 'ng-' 지시어(directive)를 많이 알아야 함

2) {{ }} : 모델과 바인딩 (ng-bind와 비슷함)

3) 데이터바인딩 : 모델과 양방향 바인딩



4) data-ng-controller : 컨트롤러 함수의 설정

5) \$scope : 적용범위

```
<!DOCTYPE html>
<html lang="ko" data-ng-app>
<head>
  <meta charset="UTF-8">
  <title>Angular.js 예제</title>
</head>
<body>
  <div data-ng-controller="controller">
    <h1>{{title}}</h1>
  </div>
  <script src="js/angular.min.js"></script>
  <script>

    function controller($scope) {
      $scope.title = '안녕하세요?';
    }
  </script>
</body>
</html>
```

```
    </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="ko" data-ng-app>
<head>
  <meta charset="UTF-8">
  <title>Angular.js 예제</title>
</head>
<body>
  <h1>{{title}}</h1>
  <div data-ng-controller="controller">
  </div>
  <script src="js/angular.min.js"></script>
  <script>
    function controller($scope) {
      $scope.title = '안녕하세요?';
    }
  </script>
</body>
</html>
```

## ■ 많이 사용하는 ng-repeat 사용해보기

### 1) jquery 이용예제

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>영화내용 출력페이지</title>
</head>
<body data-ng-app>
<h1>영화목록</h1>
<ul data-ng-controller="movieContrl">

</ul>
<script type="text/javascript" src="js/zepto.js"></script>
<script type="text/javascript">
    var movieList = [
        {poster:"Cinderella.jpg",director:"케네스
브레너",name:"신데렐라"},
        {poster:"CityLights.jpg",director:"찰리채
플린",name:"시티라이트"},
```

```

        {poster:"EleanorRigby.jpg",director:"네드
벤슨",name:"엘리노어 릭비"},
        {poster:"Fast_Furious7.jpg",director:"제임
스 완",name:"분노의 질주7"},
        {poster:"Kingsman.jpg",director:"매튜 본
",name:"킹스맨"}
    ];

    $.each(movieList,function(idx,movie){

        var $li = $("<li>");

        var $img = $("<img>").attr({width:200,src:"poster/"+movie.poster});

        var $h2 = $("<h2>").text(movie.name);

        var $p = $("<p>").text(movie.director);

        $li.append($img,$h2,$p).appendTo("ul");

    });

</script>
</body>
</html>

```

- \$.each() 함수를 이용하여 json만큼 반복하여 jquery요소객체 생성후 서로 append 하여 DOM을 생성

## 2) angular.js를 이용한 예제

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>영화내용 출력페이지</title>
</head>
<body data-ng-app>
    <h1>영화목록</h1>
    <ul data-ng-controller="movieContrl">
        <li data-ng-repeat="movie in movieList">
            

```



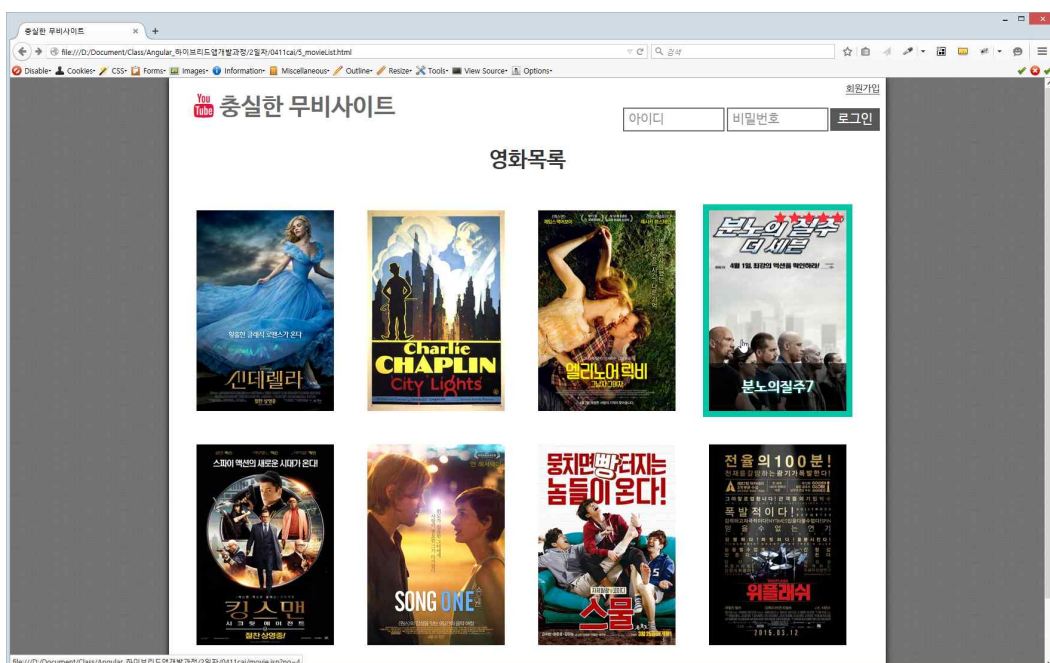
```

        <h2>{{movie.name}}</h2>
        <p>감독:{{movie.director}}</p>
    </li>
</ul>
<script type="text/javascript" src="js/angular.min.js"></script>
<script type="text/javascript">
    function movieContrl($scope) {
        $scope.movieList = [
            {poster:"Cinderella.jpg",director:"케네스 브레너",name:"신데렐라"},
            {poster:"CityLights.jpg",director:"찰리채플린",name:"시티라이트"},
            {poster:"EleanorRigby.jpg",director:"네드 벤슨",name:"엘리노어 릭비"},
            {poster:"Fast_Furious7.jpg",director:"제임스 완",name:"분노의 질주7"},
            {poster:"Kingsman.jpg",director:"매튜 본",name:"킹스맨"}
        ];
    }
</script>
</body>
</html>

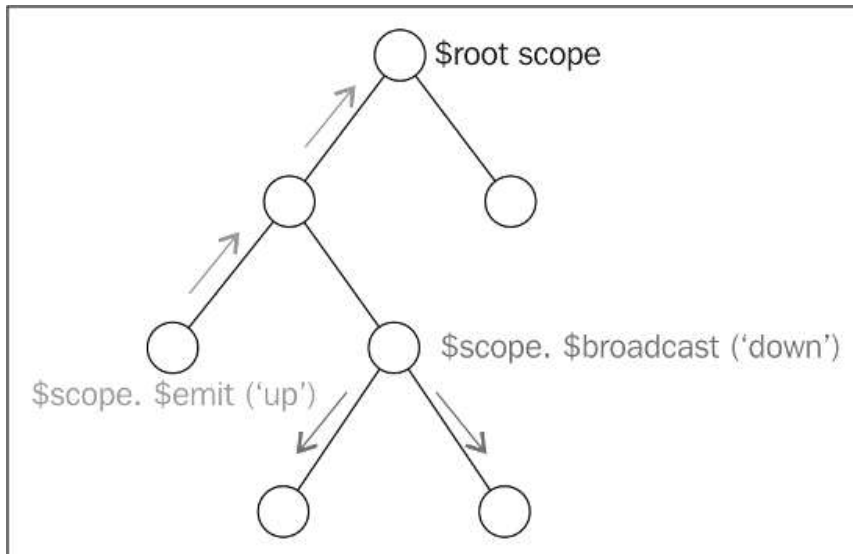
```

- 템플릿 형식으로 간단하게 데이터 바인딩이 가능함

## ■ 실전 영화목록 페이지 만들기

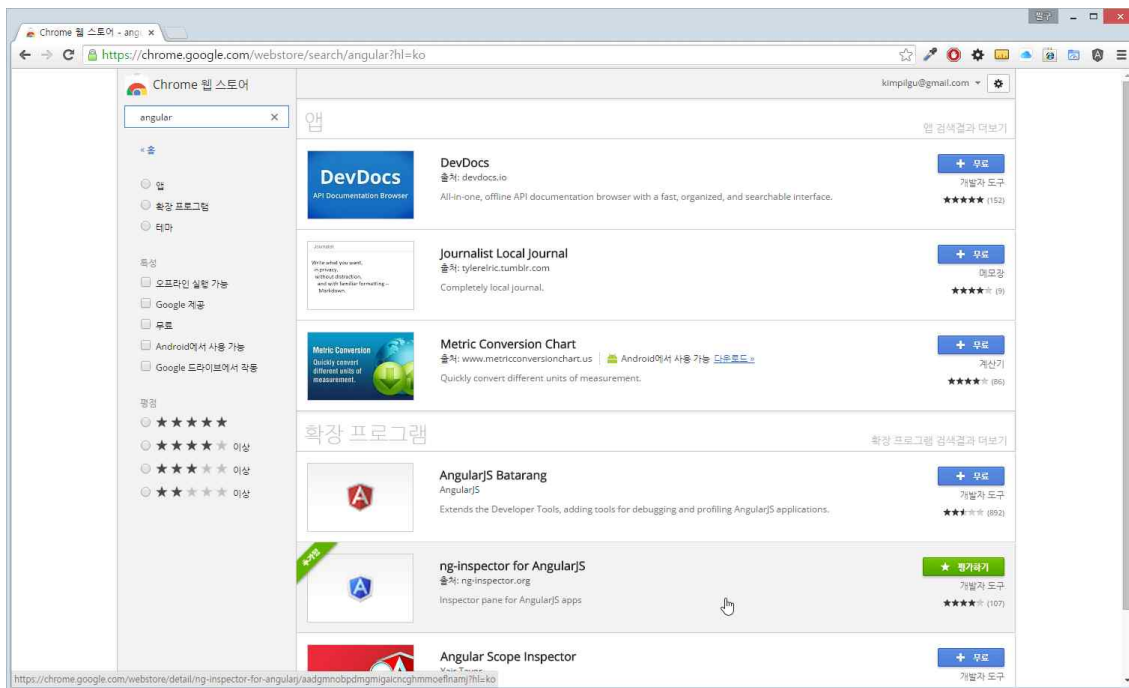


## ■ scope의 범위와 계층구조



1) 부모 \$rootScope객체가 있고 자식 \$scope가 tree구조로 생김

2) 크롬 확장프로그램 : ng-inspector for AngularJS





## ■ 스코프내의 함수를 이용하는 방법

### HTML 마크업 영역

```
<ul id="movieList" data-ng-controller="moviesController">
  <li data-ng-repeat="movie in movies">
    <a href="movie.jsp">
      
      <div >
        <h3>{{movie.name}}</h3>
        <div class="box_grade">
          <strong
            style="width:{{getStarWidth(movie)}}px">{{getAvgGrade(movie)}}</strong>
          </div>
        </div>
      </a>
```

```

    </li>
</ul><!-- //movieList -->

```

## script영역

```

function moviesController($scope) {
    $scope.movies = [
        {poster:"Cinderella.jpg",director:"케네스 브레너",name:"신데렐라",grade:13,gradeNum:2},
        {poster:"CityLights.jpg",director:"찰리 채플린",name:"시티라이트",grade:37,gradeNum:4},
        {poster:"EleanorRigby.jpg",director:"네드 벤슨",name:"엘리노어 릭비",grade:51,gradeNum:6},
        {poster:"Fast_Furious7.jpg",director:"제임스 완",name:"분노의 질주 7",grade:68,gradeNum:11},
        {poster:"Kingsman.jpg",director:"매튜 본",name:"킹스맨",grade:0,gradeNum:0},
    ];

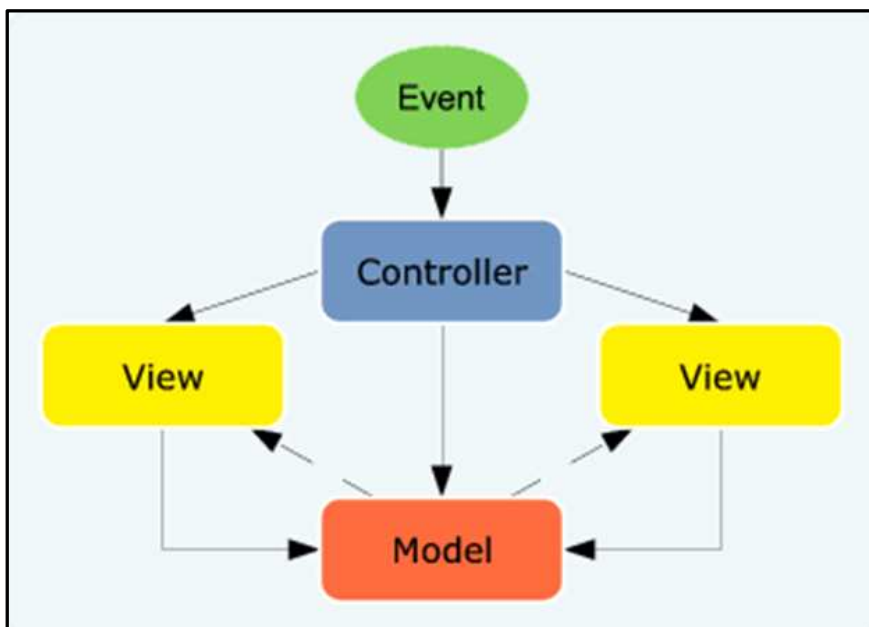
    $scope.getAvgGrade=function(movie) {

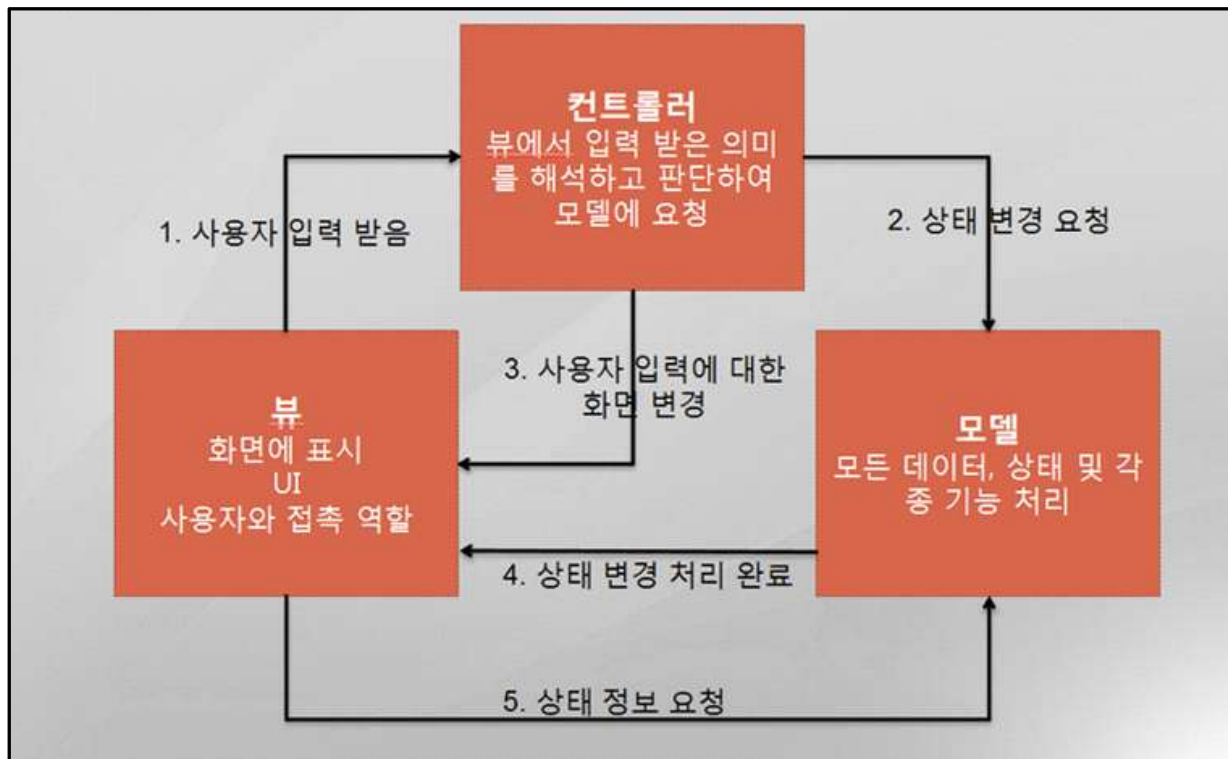
        if(movie.gradeNum!=0) {
            return Math.floor(movie.grade/movie.gradeNum*100)/100;
        }
        return movie.gradeNum;
    }

    $scope.getStarWidth=function(movie) {
        return this.getAvgGrade(movie)*12;
    }
}

```

## ■ MVC의 이해





1. Model(모델) : 데이터 그 자체
2. View(뷰) : 유저인터페이스
3. Controller(컨트롤러) : 뷰와 모델의 연결(로직을 수행)

## ■ Model

- 1) 데이터 자체
- 2) backbone.js의 Model과 다름
- 3) 프레임워크 종속성이냐? 아니면 편리함이냐?

## ■ View

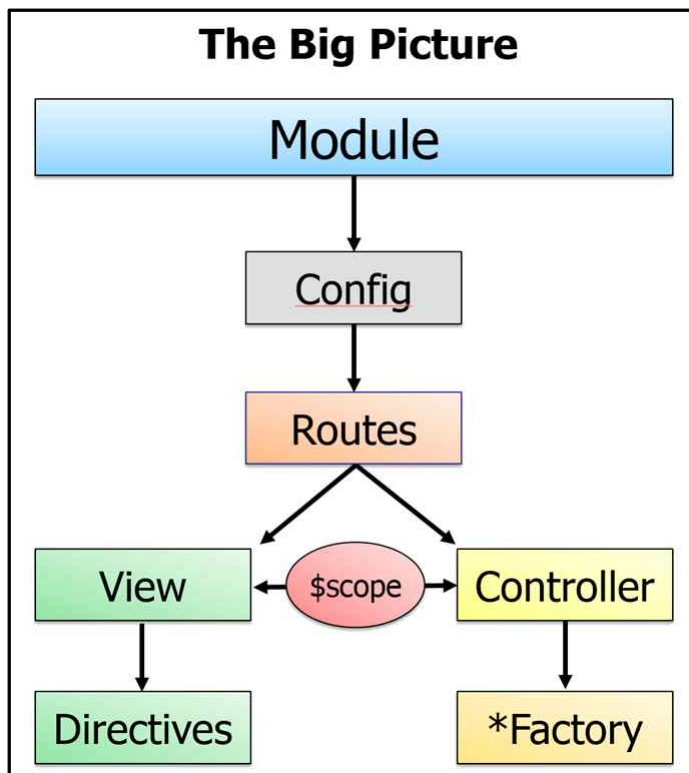
- 1) 유저 인터페이스
- 2) 모델의 정보가 출력되는 곳
- 3) 유저로부터 이벤트를 받는 곳
- 4) 선언적 프로그래밍을 해야 함

## ■ Controller

- 1) 이벤트의 감지 및 연결
- 2) 뷰의 이동(라우터 이용)
- 3) 서버와의 연결(\$http 이용)
- 4) 모델의 설정 및 초기화

## ■ 모듈의 이용

- 1) 전역상태로 모두 사용하면 유지보수나 코딩을 이해하는데 많은 어려움이 있음
- 2) 하나의 페이지에 모듈을 여러개로 나눠서 개발하면 편리함



```
<body data-ng-app="myApp">
  <div data-ng-controller="boxController">
    <h1>{{title}}</h1>

  </div>
  <script type="text/javascript" src="js/angular.min.js"></script>
  <script type="text/javascript">
```

```
//{{{ <- 인터플레이션

var module = angular.module("myApp",[]);//[ ]필요한 모듈의 의존성 입력

module.controller("boxController",function($scope){
    $scope.title="모듈을 이용합니다.";
});
</script>
</body>
```

■ bind 지시어

지시어	설 명
ng-model	Model객체 생성(input,textarea,select)
ng-bind	모델을 바인딩({})와 같음
ng-show	true면 안보임 / false면 보임
ng-hide	true면 숨기고 / false면 안보임
ng-disabled	true면 disable / false면 반대
ng-readonly	true면 readonly / false면 반대
ng-checked	true면 checked / false면 반대
ng-multiple	true면 multiple속성 / false면 반대

```
<!DOCTYPE html>
<html lang="ko" data-ng-app>
<head>
    <meta charset="utf-8">
    <title>bind</title>
</head>
<body data-ng-init="txt='a'">
    <h1 data-ng-bind="txt">Test</h1>
    <input data-ng-model="txt" />
    <input type="checkbox" data-ng-model="chk"/>
    <h2 data-ng-show="chk">안녕하세요?</h2>
    <h2 data-ng-hide="chk">반갑습니다?</h2>
    <input data-ng-disabled="chk" value="못씀"/>
    <input data-ng-readonly="chk" value="읽기만 함"/>
    <script src="js/angular.min.js"></script>
```

```
</body>
</html>
```

## ■ filter의 사용

1) 데이터의 출력에 다양한 필터를 활용하거나 직접 filter를 만들 수 있음

필터명	설 명
currency	통화기호와 숫자로 표현
date	날짜를 특정형식으로 표현함
number	소수점 자리수로 숫자를 표현
lowercase	소문자로 처리
uppercase	대문자로 처리
json	이 필터를 이용하여 json 디버깅이 편함

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>필터의 사용</title>
</head>
<body data-ng-app data-ng-controller="movieController">
<h1>영화검색</h1>
<p>
<label>이름으로 검색 : <input type="text" data-ng-model="search" /></label>
</p>
<ul >
<li data-ng-repeat="movie in movies | filter:search">제목: {{movie.name}} 평점 :
{{getAvgGrade(movie)|number:2}}</li>
</ul>
<p>
<h2>여자는 {{femaleNum}}명입니다.</h2>
<ul>
<li data-ng-repeat="person in people">{{person.name}}</li>
</ul>
<script type="text/javascript" src="js/angular.min.js"></script>
<script type="text/javascript">

function movieController($scope,$filter) {
```



```

$scope.movies = [
    {poster:"Cinderella.jpg",director:"케네스 브레너",name:"신데렐라",grade:13,gradeNum:2},
    {poster:"CityLights.jpg",director:"찰리채플린",name:"시티라이트",grade:37,gradeNum:4},
    {poster:"EleanorRigby.jpg",director:"네드 벤슨",name:"엘리노어 릭비",grade:51,gradeNum:6},
    {poster:"Fast_Furious7.jpg",director:"제임스 완",name:"분노의 질주7",grade:68,gradeNum:11},
    {poster:"Kingsman.jpg",director:"매튜 본",name:"킹스맨",grade:0,gradeNum:0},
];

$scope.getAvgGrade=function(movie) {
    if(movie.gradeNum!=0) {
        return movie.grade/movie.gradeNum;
    }
    return movie.gradeNum;
}

$scope.people= [
    {name:"김연아",gender:"여자"},
    {name:"류현진",gender:"남자"},
    {name:"박지성",gender:"남자"},
    {name:"수지",gender:"여자"}
];

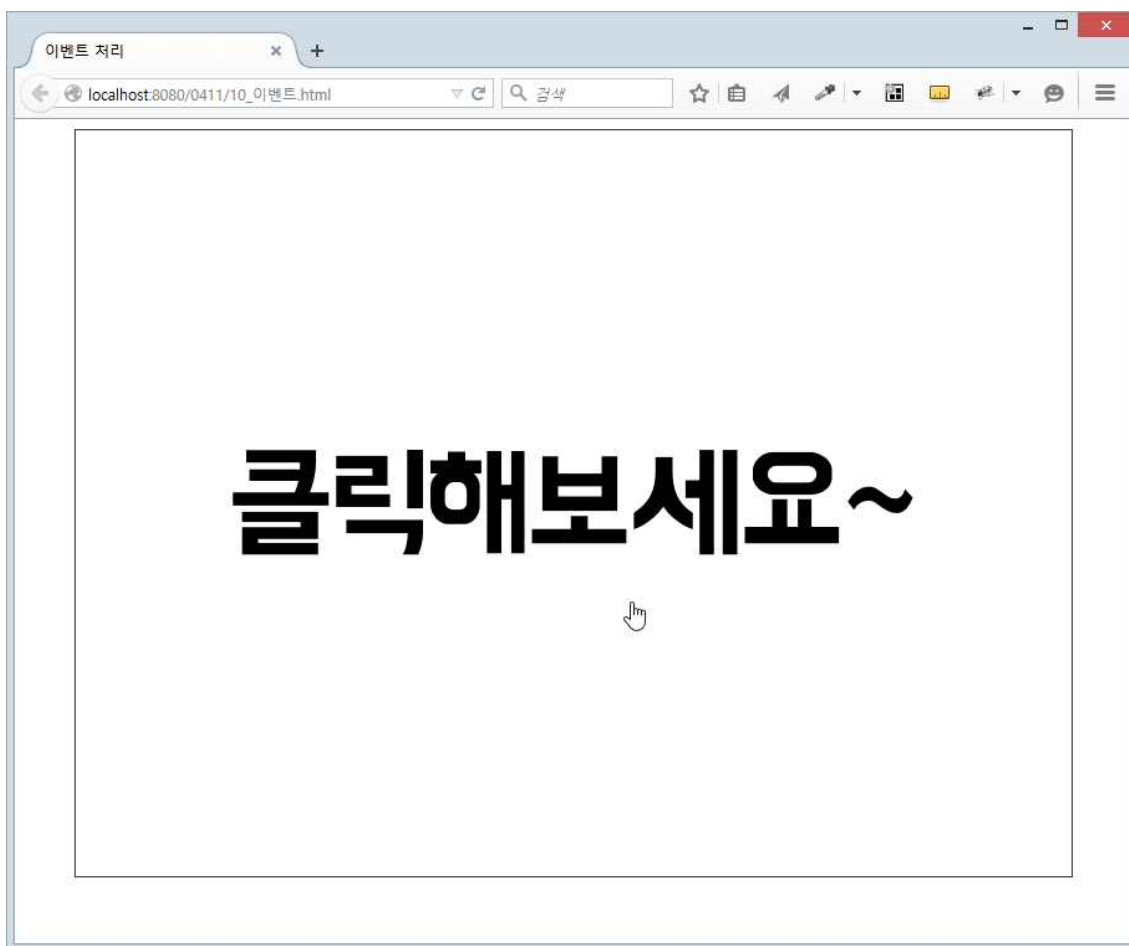
$scope.femaleNum = $filter('filter')($scope.people,{gender:"여자"}).length;
}

</script>
</body>
</html>

```

## ■ 이벤트 지시어

지시어	설 명
ng-change	change이벤트
ng-click	click이벤트
ng-dbclick	더블클릭이벤트
ng-mousedown	마우스누름
ng-submit	form에서 submit
ng-keydown	키 누름
그 외	우리가 아는 이벤트에 ng-붙이기



```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>이벤트 처리</title>
<link rel="stylesheet"
href="http://fonts.googleapis.com/earlyaccess/hanna.css" />
<style>
```

```

        #clickBox {
            font-family: 'Hanna', serif;
            font-size:100px;
            width:800px;
            height:600px;
            text-align:center;
            line-height:600px;
            border:1px solid #333;
            margin:auto;
            cursor: pointer;
        }
</style>
</head>
<body data-ng-app="eventApp" data-ng-controller="eventController" >
    <div id="clickBox" data-ng-click="clickBox($event)">
        <span data-ng-bind="msg"></span>
    </div>
    <script type="text/javascript" src="js/angular.min.js"></script>
    <script type="text/javascript">
        var module = angular.module("eventApp",[]);

        module.controller("eventController",function($scope){

            $scope.msg="클릭해보세요~";

            $scope.clickBox = function($event) {
                $scope.msg      =      "x:"      +      $event.clientX+"/
y:"+$event.clientY;
            }

        });

    </script>
</body>
</html>

```

## ■ 자바스크립트 배열 공부하기

### 1) 배열이 가진 메서드

메서드명	설 명
push(데이터)	배열에 새 요소를 추가하고 배열의 새 길이를 반환합니다.
pop()	배열의 마지막 요소를 제거하여 반환합니다.
unshift(데이터)	배열 시작에 새 요소를 삽입합니다.
shift()	배열에서 첫 번째 요소를 제거하여 반환합니다.
splice(index,갯수)	index부터 갯수만큼 제거

### 2) 예제

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>array</title>
</head>
<body>
<script>
    //var arr = new Array("김연아","손흥민","류현진");
    //alert(arr.length);
    var arr = ["김연아","손흥민","류현진"];
    arr.push("박지성");
    //alert(arr);//김연아,손흥민,류현진,박지성
    arr.unshift("박태환");
    //alert(arr);//박태환,김연아,손흥민,류현진,박지성
    arr.splice(2,1);//2번째index의 손흥민 제거
    //alert(arr);//박태환,김연아,류현진,박지성
    arr.pop();//박태환,김연아,류현진
    arr.shift();//김연아,류현진
    arr.shift();//류현진
    alert(arr);
</script>
</body>
</html>
```

## ■ angular Todo MVC 예제



<http://todomvc.com/>

### 1) todo목록 애플리케이션 제작

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8">
    <title>Todo App만들기</title>
  </head>
  <body>
    <div>
      <h1>제목</h1>
      <p>전체 할일 <em>2</em>개 / 남은 할일은 <strong>1</strong>개</p>
      <ul>
        <li class="to-do-list"><label><input type="checkbox" />독서</label></li>
        <li class="to-do-list"><label><input type="checkbox" />공부</label></li>
      </ul>
      <form>
        <fieldset>
          <legend>할일 입력폼</legend>
          <div>
            <input id="txt" placeholder="새로운 할일 입력" />
            <button>추가</button>
          </div>
        </fieldset>
      </form>
    </div>
  </body>
</html>
```

```
        </fieldset>
    </form>
</div>
</body>
</html>
```

## 2) data-ng-app 속성추가

```
<html lang="ko" data-ng-app>
```

## 3) controller 만들기

```
<div data-ng-controller="controller">

function controller($scope) {
    //alert("test");
}


```

## 4) 타이틀 동적으로 부여

```
<h1>{{title}}</h1>

function controller($scope) {
    $scope.title = '필구의 Todo!';
}


```

## 5) todo리스트를 json으로 만들기

```
$scope.todoList = [
    { end : true, title : "독서"},
    { end : false, title : "달팽이 구워먹기"},
    { end : false, title : "프로젝트 완성"}
];
}
```

## 6) 리스트 출력

- ng-repeat은 for문과 같음(jstl의 forEach와 같음)
- ng-model은 해당요소에 모델을 바인딩시키는 지시어  
(ng-model은 input,textarea,select요소에만 가능)
- {{}}는 표현식

```
<li data-ng-repeat="todo in todoList">
  <label><input type="checkbox" data-ng-model="todo.end"/>{{todo.title}}</label>
</li>
```

## 7) 새로운 할일 입력

- ng-ng-submit은 submit이벤트리스너
- \$event객체와 / newDo MODEL을 DI(의존성 주입)

```
<form data-ng-submit="addTodo($event,newDo)">
<fieldset>
  <legend>할일 입력폼</legend>
  <div>
    <input id="txt" placeholder="새로운 할일 입력" data-ng-model="newDo" />
    <button>추가</button>
  </div>
</fieldset>
</form>
```

- \$event.preventDefault()는 submit막기
- newDo 내용이 없다면 아무일도 안하기
- 내용이 있다면 todoList에 할일 객체 바인딩
- \$scope.newDo를 비우기
- input에 focus()주기

```
$scope.addTodo=function($event,newDo) {
  $event.preventDefault();
  if(newDo.length==0) return false;
  $scope.todoList.push({end:false,title:newDo});
  $scope.newDo = ""; //newDo=""와는 완전히 다름
  angular.element("#txt").focus();
};
```

## 8) 전체 할일 갯수와 남은 할일 갯수 구하기

- data-ng-bind를 이용해 todoList 모델의 length 바인딩
- 남은 할일은 처리를 해야 하므로 remainDo()함수와 바인딩

<p>  
전체 할일 <em data-ng-bind="todoList.length"></em>개  
/ 남은 할일은 <strong data-ng-bind="remainDo(todoList)"></strong>개  
</p>

- todoList를 DI
- angular.forEach메서드를 이용해 for each 수행
- 각 todo모델의 end속성이 false인 갯수를 리턴

```
$scope.remainDo = function(todoList) {  
    var cnt = 0;  
    angular.forEach(todoList, function(todo, index) {  
        if(!todo.end) {  
            cnt++;  
        }  
    });  
    return cnt;  
};
```

## 9) 완료된 할일 목록 비우기

- clearEnd(todoList) 이벤트리스너 붙이기

<button data-ng-click="clearEnd(todoList)">완료없애기</button>

- 기존의 todoList를 받고, 새로운 배열을 만들어 완료되지 않은 todo모델만 push

```
$scope.clearEnd=function(todoList) {  
    var oldList = todoList;  
    $scope.todoList = [];  
    angular.forEach(oldList, function(value, key) {  
        if(!value.end) {  
            $scope.todoList.push(value);  
        }  
    });  
};
```



## 10) 할일 개별삭제 버튼 만들기

- delete(\$index) 에서 \$index는 요소의 index

```
<li data-ng-repeat="todo in todoList">
  <label><input type="checkbox" data-ng-model="todo.end"/>
    {{todo.title}}
  </label>
  <button data-ng-click="delete($index)">삭제</button>
</li>
```

todoList모델에서 해당 모델 삭제  
(양방향 바인딩이 되어 있기 때문에 자동으로 삭제됨)

```
$scope.delete = function($index) {
  $scope.todoList.splice($index,1);
};
```

### ■ 전체 소스

```
<!DOCTYPE html>
<html lang="ko" data-ng-app>
  <head>
    <meta charset="UTF-8">
    <title>Todo App만들기</title>
  </head>
  <body>
    <div data-ng-controller="controller">
      <h1 data-ng-bind="title"></h1>
      <p>전체 할일 <em data-ng-bind="todoList.length"></em>개
        / 남은 할일은 <strong data-ng-bind="remainDo(todoList)"></strong>
        개
        <button data-ng-click="clearEnd(todoList)">완료없애기</button></p>
      <ul>
        <li data-ng-repeat="todo in todoList">
          <label><input
            data-ng-model="todo.end"/>{{todo.title}}</label>
          <button data-ng-click="delete($index)">삭제</button>
        </li>
      </ul>
      <form data-ng-submit="addTodo($event,newDo)">
```

```

        <fieldset>
            <legend>할일 입력폼</legend>
            <div>
                <input id="txt" placeholder="새로운 할일 입력"
data-ng-model="newDo" />
                <button>추가</button>
            </div>
        </fieldset>
    </form>
</div>
<script src="js/jquery.js"></script>
<script src="js/angular.min.js"></script>
<script>

    function controller($scope) {
        $scope.title = '필구의 Todo!';
        $scope.todoList = [
            { end : true, title : "독서"},
            { end : false, title : "달팽이 구워먹기"},
            { end : false, title : "프로젝트 완성"}
        ];

        $scope.addTodo=function($event,newDo) {

            $event.preventDefault();

            if(newDo.trim().length==0) return false;

            $scope.todoList.push({end:false,title:newDo});

            //newDo="";와는 완전히 다름
            $scope.newDo = "";

            angular.element("#txt").focus();

        };

        $scope.delete = function($index) {

```

```
        $scope.todoList.splice($index,1);
    };

    $scope.remainDo = function(todoList) {

        var cnt = 0;

        angular.forEach(todoList, function(todo, index) {

            if(!todo.end) {
                cnt++;
            }

        });

        return cnt;
    };

    $scope.clearEnd=function(todoList) {

        var oldList = todoList;

        $scope.todoList = [];

        angular.forEach(oldList, function(value, key) {

            if(!value.end) {
                $scope.todoList.push(value);
            }

        });

    };
}
```

</script>

</body>

</html>

## ■ route 구현

- 1) 라우트란 해당 주소에 따라 뷰 템플릿이 변경되는 것을 말함
- 2) \$routeProvider를 이용하여 라우팅
- 3) when() / otherwise()를 이용하여 if~else처럼 라우팅이 가능
- 4) nav 만들기

```
<h1><a href="#home">라우팅 예제</a></h1>
<ul>
  <li><a href="#list">목록</a></li>
  <li><a href="#detail">디테일</a></li>
  <li><a href="#my">마이페이지</a></li>
</ul>
```

a요소를 이용하여 링크이동

## 5) 라우팅 모듈 생성

```
angular.module("ngModule", ['ngRoute']).config(function($routeProvider){
  $routeProvider.when("/list",{templateUrl:"list.html"})
    .when("/home",{templateUrl:"home.html"})
    .when("/detail",{templateUrl:"detail.html"})
    .when("/my",{templateUrl:"my.html"})
    .otherwise({redirectTo:"/home"});
});
```

\$routeProvider에 when(), otherwise()로 주소와 페이지 연결

## 6) a요소가 아닌경우 location.hash를 이용해서 주소 변경

```
<button data-ng-click="changeLocation()">리스트페이지로 이동</button>
function controller($scope) {
  $scope.changeLocation=function() {
    location.hash="list";
  };
};
```