

Filter

필터는 사용자에게 표시하는 식의 값을 필터링합니다. 뷰 템플릿 컨트롤러 서비스에서 사용할 수 있으며, 기존의 필터를 정의하는 것은 간단합니다. 기본적인 API는 `filterProvider` 에서 제공됩니다.

View Template에서 필터의 사용

View Template에서 아래의 문법에서 식에 적용 할 수 있습니다.

```
{{ expression | filter }}
```

예) `{{12 | currency}}` 은 수치 인 12을 `currency` 필터를 사용하여 통화로 간주합니다. 결과는 \$12.00입니다.

필터의 결과에 다른 필터를 적용 할 수 있습니다. 이것을 "체인(chaining)"라고 하며 다음과 같은 문법으로 합니다.

```
{{ expression | filter1 | filter2 | ... }}
```

필터는 인수를 취할 수도 있으며, 문법은 아래와 같습니다.

```
{{ expression | filter:argument1:argument2:... }}
```

예) `{{1234 | number : 2}}` 태그는 수치 인 1234 `number` 필터를 사용하여 소수 둘째 자리까지 표시합니다. 결과는 1,234.00입니다.

Controller와 Service에서 필터의 사용

필터는 컨트롤러와 서비스에서 사용할 수 있습니다. 사용하기 위해서는 <필터 이름> `Filter`의 이름으로 자신의 컨트롤러 또는 서비스에 종속성을 추가합니다.

예) `numberFilter`의 의존성을 사용하면 `number` 필터가 주입됩니다. 인수로 주입되는 함수는 첫 번째 인수에서 형식 값을 두 번째 인수 이후 필터의 인수를 받습니다.

아래의 예에서는 `filter` 라는 필터를 사용하고 있습니다. 이 필터는 조건에 따라 배열을 필터링하고 그 결과 항목이 짜진 배열을 가져옵니다. 이 필터는 View Template에서 `{{ctrl.array | filter : 'a'}}` 과 같은 마크 업에 사용이 경우 내용이 문자열이 'a'만 항목으로 좁혀지게 됩니다. 그러나 View Template에서 필터의 사용은 digest마다 필터의 재평가가 이루어지기 때문에 큰 배열의 경우 부하가 커지므로 주의가 필요합니다.

따라서 아래의 예에서는 이 필터를 컨트롤러에서 직접 호출합니다. 이렇게 하면 필요한 경우에만 컨트롤러가 필터를 호출합니다. (예를 들어, backend에서 데이터를 가져올 경우 필터 식의 내용이 변경된 경우)

script.js

```
angular.module('FilterInControllerModule', []).
controller('FilterController', ['filterFilter', function(filterFilter) {
  this.array = [
    {name: 'Tobias'},
    {name: 'Jeff'},
    {name: 'Brian'},
    {name: 'Igor'},
    {name: 'James'},
    {name: 'Brad'}
  ];
  this.filteredArray = filterFilter(this.array, 'a');
}]);
```

index.html

```
<div ng-controller="FilterController as ctrl">
  <div>
    All entries:
    <span ng-repeat="entry in ctrl.array">{{entry.name}} </span>
  </div>
  <div>
    Entries that contain an "a":
    <span ng-repeat="entry in ctrl.filteredArray">{{entry.name}} </span>
  </div>
</div>
```

사용자정의 필터 만들기

자신의 필터를 만드는 것은 매우 간단합니다. 모듈에 새로운 필터 팩토리 함수를 등록하면 됩니다. 내부에서 filterProvider가 사용되게 됩니다. 이 Factory는 입력 값을 첫 번째 인수로 새로운 필터 함수를 반환해야 합니다. 이어지는 인수가 그대로 필터의 인수로 필터 함수에 전달됩니다.

아래의 예 필터는 텍스트 문자열을 뒤집어 표시합니다. 또한 조건을 만족하면 텍스트를 대문자로 변환합니다.

script.js

```
angular.module('myReverseFilterApp', [])
.filter('reverse', function() {
  return function(input, uppercase) {
    input = input || '';
    var out = '';
    for (var i = 0; i < input.length; i++) {
      out = input.charAt(i) + out;
    }
    // conditional based on optional argument
    if (uppercase) {
      out = out.toUpperCase();
    }
    return out;
  };
})
.controller('MyController', ['$scope', function($scope) {
  $scope.greeting = 'hello';
}]]);
```

index.html

```
<div ng-controller="MyController">
  <input ng-model="greeting" type="text"><br>
  No filter: {{greeting}}<br>
  Reverse: {{greeting|reverse}}<br>
  Reverse + uppercase: {{greeting|reverse:true}}<br>
</div>
```