

Use Stack Overflow for Teams at work to find answers in a private and secure environment. Get your first 10 users free. [Sign up.](#)

# Check if value in array in angular template?

Asked 5 years, 8 months ago   Active 4 years ago   Viewed 70k times



Is there any way to check if a value is in an array in an angular template? I'm looking to something like this:

36

```
<div ng-class="{ 'myClass': 1 in [1,2,5] }">Yay</div>
```



Where myClass is applied if 1 is in the array.



Is this possible?

4

And as an aside, what templating engine exactly is used with Angularjs? Where can I find documentation for it? Whenever I try and search (even official docs), I only seem to turn up docs on directives or data binding.

javascript

angularjs

template-engine

edited Aug 27 '15 at 12:32



Nicolas Zozol

5,315   1   41   60

asked Jan 14 '14 at 20:44



Anonymous

3,083   6   35   64

## 3 Answers



You can use `indexOf()` to test whether a value is in an array and then use it inside your `ngClass` like this (to conditionally add "newclass"):

71

```
<div ng-class="{ 'newclass': ([1,2,5].indexOf(2) > -1) }">Yay</div>
```



Or more likely you'll want to test against an array on your scope:



```
<div ng-class="{ 'newclass': (tarray.indexOf(1) > -1) }">Yay</div>
```

Assuming, for instance, you have declared `tarray` in your controller:

```
$scope.tarray=[1,2,5];
```

[demo](#)

As far as a template engine, it's built in to Angular. So there's not really something separate to search for. But here's the [top level template docs](#) and there's some good [video tutorials here](#) that cover templates (for instance the first one is on data binding) as well as much more.

edited Jan 14 '14 at 21:03

answered Jan 14 '14 at 20:58



[KayakDave](#)

24.1k 3 59 67

1 Thanks, exactly what I was looking for - didn't realize quite how much JS I could use in the templates. – [Anonymous](#) Jan 14 '14 at 21:03

1 This doesn't work for object inside arrays though – [Mauro De Giorgi](#) Feb 28 '14 at 23:24 ✎

You can create a filter to check this. Something like:

13

```
app.filter('inArray', function() {  
  return function(array, value) {  
    return array.indexOf(value) !== -1;  
  };  
});
```

Then you can use this in your template with:

```
<div ng-class="{newclass': tarray | inArray : 1}">Yay</div>
```

The advantage to this is that it makes your template more readable.

edited Jun 12 '15 at 20:57



[dansalmo](#)


8,779 4 43 46


answered Aug 21 '14 at 8:06



[Martin](#)

11.5k 3 34 50

2 Nice solution, but I needed to add parenthesis around the filter expression to avoid an angular syntax error. `<div ng-class="{ 'newclass': (tarray | isArray : 1) }">Yay</div>` – [dansalmo](#) Jun 12 '15 at 21:05 

You could add the verification if `array` is an `Array` object (it might not always be an array, especially on initialization): `return Array.isArray(array) && array.indexOf(value) !== -1;` – [Dmitri Pavlutin](#) Dec 13 '15 at 15:47 

Try this:

5 `ng-class="{ 'myClass': !!~[1,2,5].indexOf(1) }"`

if the value isn't in array you get 0 therefore with `!!` you get false

answered Jan 14 '14 at 21:09



[Whisher](#)

12k 24 86 162

`indexOf` returns -1 when the item is not in the array – [Gruff Bunny](#) Jan 14 '14 at 21:12

@Gruff Bunny `~[1,2,5].indexOf(3)` with the bit operations you get 0 – [Whisher](#) Jan 14 '14 at 21:13 

Oops - didn't see the not there - i do apologise – [Gruff Bunny](#) Jan 14 '14 at 21:16

Great idea- but unfortunately bitwise operators are not supported in Angular expressions: [github.com/angular/angular.js/issues/2838](https://github.com/angular/angular.js/issues/2838) – [KayakDave](#) Jan 14 '14 at 21:22

