

Angular: conditional class with *ngClass

Asked 3 years, 7 months ago Active 4 days ago Viewed 517k times



What is wrong with my Angular code? I am getting:

388

Cannot read property 'remove' of undefined at BrowserDomAdapter.removeClass ...



HTML



91

```
<ol class="breadcrumb">
  <li *ngClass="{active: step==='step1'}" (click)="step='step1; '">Step1</li>
  <li *ngClass="{active: step==='step2'}" (click)="step='step2'">Step2</li>
  <li *ngClass="{active: step==='step3'}" (click)="step='step3'">Step3</li>
</ol>
```

javascript

css

 angular

angular-template

angular-ng-class

edited Nov 13 '18 at 7:37



MostafaMashayekhi

8,604 2 12 32

asked Feb 8 '16 at 12:00



daniel

10.8k 32 79 135

angulartutorial.net/2017/12/... – Prashobh Dec 12 '17 at 6:15

14 Answers



Angular 2,...,7 provides several ways to add classes conditionally:

790

type one

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



type two

```
[ngClass]="{'my-class': step=='step1'}"
```

and multiple option:

```
[ngClass]="{'my-class': step=='step1', 'my-class2':step=='step2' }"
```

type three

```
[ngClass]="{1:'my-class1',2:'my-class2',3:'my-class4'}[step]"
```

type four

```
[ngClass]="(step=='step1')?'my-class1':'my-class2'"
```

edited Mar 8 at 6:38

answered Feb 1 '17 at 7:50



[Mostafa Mashayekhi](#)

8,604 2 12 32

4 Perfect answer, just fix the type 2 to: `[ngClass]="{'my-class': step=='step1'}"` With the " int the class name – [Adriano Galesso Alves](#) Mar 28 '17 at 6:38



I was looking for the type four, but I'm wondering if I can add another class with another condition into that expression? Thanks in advance – [Louay Hamada](#) May 15 '18 at 17:48

For type three, the order of the class name and check is wrong. It should be class name first such as `[ngClass]="{'my-class1': 1, 'my-class2': 2 }"` – [obaylis](#) Jul 18 '18 at 11:33

looks like "type three" and "type four" are specific usages of `[ngClass]="js expression returning html class string"` so that are the same in this sense – [YakovL](#) Jul 26 at 12:04



`[ngClass]=...` instead of `*ngClass` .

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
<div *ngFor="let item of items">{{item}}</div>
```

instead of the longer equivalent version

```
<template ngFor let-item [ngForOf]="items">
  <div>{{item}}</div>
</template>
```

See also <https://angular.io/docs/ts/latest/api/common/index/NgClass-directive.html>

```
<some-element [ngClass]="['first second']">...</some-element>
<some-element [ngClass]="['first', 'second']">...</some-element>
<some-element [ngClass]="{'first': true, 'second': true, 'third': false}">...</some-
element>
<some-element [ngClass]="stringExp|arrayExp|objExp">...</some-element>
<some-element [ngClass]="{'class1 class2 class3' : true}">...</some-element>
```

See also <https://angular.io/docs/ts/latest/guide/template-syntax.html>

```
<!-- toggle the "special" class on/off with a property -->
<div [class.special]="isSpecial">The class binding is special</div>

<!-- binding to `class.special` trumps the class attribute -->
<div class="special"
  [class.special]="!isSpecial">This one is not so special</div>

<!-- reset/override all class names with a binding -->
<div class="bad curly special"
  [class]="badCurly">Bad curly</div>
```

edited Mar 28 '17 at 7:51

answered Feb 8 '16 at 12:02



Günter Zöchbauer

369k 85 1179 1070

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

property binding,[ngIf]. The rest of the <div>, including its class attribute, moved inside the <ng-template> element." - more info @ angular.io/guide/structural-directives#the-asterisk--prefix – Combine May 23 at 8:56 ✎

Actually, it's nothing more complicated, * just allows a simplified syntax instead of canonical form. – Günter Zöchbauer Jun 28 at 17:47

Another solution would be using `[class.active]` .

70

Example :

```
<ol class="breadcrumb">
  <li [class.active]="step==='step1'" (click)="step='step1'">Step1</li>
</ol>
```

answered Feb 8 '16 at 12:22



Joel Almeida

6,294 3 19 49

7 I think this should be the accepted answer as this is the Angular2 way to set the html class (which I didn't know about and google brought me here). – kub1x Nov 21 '16 at 14:07

That's the normal structure for `ngClass` is:

56

`[ngClass]="{'classname' : condition}"`

So in your case, just use it like this...

```
<ol class="breadcrumb">
  <li [ngClass]="{'active': step==='step1'}" (click)="step='step1'">Step1</li>
  <li [ngClass]="{'active': step==='step2'}" (click)="step='step2'">Step2</li>
  <li [ngClass]="{'active': step==='step3'}" (click)="step='step3'">Step3</li>
</ol>
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



5,244

9

60

90



61.6k

15

197

133

with the following examples you can use 'IF ELSE'

36

```
<p class="{condition ? 'checkedClass' : 'uncheckedClass'}">  
<p [ngClass]="condition ? 'checkedClass' : 'uncheckedClass'">  
<p [ngClass]="[condition ? 'checkedClass' : 'uncheckedClass']">
```

edited Dec 21 '18 at 8:40

answered Jun 8 '18 at 13:07



Chaitanya Nekkhalapudi

539

4

6

I tried the first and the second solution. Only the second worked for me – [user1238784](#) Jul 2 '18 at 7:03

2 check your quotations! first also should work. – [Chaitanya Nekkhalapudi](#) Jul 3 '18 at 7:50

You can use ngClass to apply the class name both conditionally and not in Angular

30

For Example

```
[ngClass]=" 'someClass' ">
```

Conditional

```
[ngClass]="{'someClass': property1.isValid}">
```

Multiple Condition

```
[ngClass]="{'someClass': property1.isValid && property2.isValid}">
```

Method expression

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

This method will inside of your component

```
getSomeClass(){
    const isValid=this.property1 && this.property2;
    return {someClass1:isValid , someClass2:isValid};
}
```

edited Dec 16 '17 at 9:49



sonance207

350 1 6 15

answered Aug 4 '17 at 18:44



Code-EZ

4,004 9 36 61

You should use something ([ngClass] instead of *ngClass) like that:

12

```
<ol class="breadcrumb">
  <li [ngClass]="{active: step==='step1'}" (click)="step='step1; '">Step1</li>
  (...)
```

answered Feb 8 '16 at 12:05



Thierry Templier

158k 29 342 311

In Angular 7.X

6

The CSS classes are updated as follows, depending on the type of the expression evaluation:

- string - the CSS classes listed in the string (space delimited) are added
- Array - the CSS classes declared as Array elements are added
- Object - keys are CSS classes that get added when the expression given in the value evaluates to a truthy value, otherwise they are removed.

```
<some-element [ngClass]="['first second']">...</some-element>
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

element>

```
<some-element [ngClass]="stringExp|arrayExp|objExp">...</some-element>
```

```
<some-element [ngClass]="{'class1 class2 class3' : true}">...</some-element>
```

edited May 27 at 5:36



span

3,805 3 41 87

answered Dec 5 '18 at 14:24



Rohit.007

2,084 2 10 21

While I was creating a reactive form, I had to assign 2 types of class on the button. This is how I did it:

5

```
<button type="submit" class="btn" [ngClass]="(formGroup.valid)?'btn-info':''"
[disabled]="!formGroup.valid">Sign in</button>
```

When the form is valid, button has btn and btn-class (from bootstrap), otherwise just btn class.

answered Sep 6 '17 at 23:49



Sarvar Nishonboev

4,412 4 31 42

1 This worked for me when merging classes – dc10 Sep 19 '17 at 21:59

to extend MostafaMashayekhi his answer for option two> you can also chain multiple options with a ','

5

```
[ngClass]="{'my-class': step=='step1', 'my-class2':step=='step2' }"
```

Also *ngIf can be used in some of these situations usually combined with a *ngFor

```
class="mats p" *ngIf="mat=='painted'"
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

ngClass syntax:

3 `[ngClass]='{'classname' : conditionFlag}'`

You can use like this:

```
<ol class="breadcrumb">
  <li [ngClass]='{'active': step==='step1'}' (click)="step='step1'">Step1</li>
  <li [ngClass]='{'active': step==='step2'}' (click)="step='step2'">Step2</li>
  <li [ngClass]='{'active': step==='step3'}' (click)="step='step3'">Step3</li>
</ol>
```

answered Jun 19 at 5:53



Chirag

331 2 11

why duplicate several already existing answers? – YakovL Jul 26 at 12:09

This is what worked for me:

3 `[ngClass]='{'active': dashboardComponent.selected_menu == 'profile'}'`

edited Jun 28 at 10:38



Prashant Pimpale

5,425 4 12 44

answered Mar 23 '18 at 4:36



Ninad Kulkarni

164 1 5

Let, YourCondition is your condition or a boolean property, then do like this

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

answered Mar 20 at 6:15



Abdus Salam Azad

1,216 13 12



Not relevant with `[ngClass]` directive but I was also getting the same error as

0

Cannot read property 'remove' of undefined at...



and I thought to be the error in my `[ngClass]` condition but it turned out the property I was trying to access in the condition of `[ngClass]` was not initialized.

Like I had this in my typescript file

```
element: {type: string};
```

and In my `[ngClass]` I was using

```
[ngClass]="{'active', element.type === 'active'}"
```

and I was getting the error

Cannot read property 'type' of undefined at...

and the solution was to fix my property to

```
element: {type: string} = {type: 'active'};
```

Hope it helps somebody who is trying to match a condition of a property in `[ngClass]`

answered Sep 18 at 9:18



Hamza Khanzada

325 2 10

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).