ngFor with index as value in attribute

395

48

I have a simple `ngFor` loop which also keeps track of the current `index`. I want to store that `index` value in an attribute so I can print it. But I can't figure out how this works.

I basically have this:

```
<ul *ngFor="#item of items; #i = index" data-index="#i">
    <li>{{item}}</li>
</ul>
```

I want to store the value of `#i` in the attribute `data-index`. I tried several methods but none of them worked.

I have a demo here: http://plnkr.co/edit/EXpOKAEIFlI9QwuRcZqp?p=preview

How can I store the `index` value in the `data-index` attribute?

angular ngfor

edited Jan 19 '18 at 15:44
Wesley Coetzee
3,304 1 20 38

asked Feb 15 '16 at 9:27
Vivendi
6,208 21 73 153

Delete the edit. I though that I was trying to edit one of my questions. It has been an error – Mario Jan 30 '18 at 9:44

6 Answers

I would use this syntax to set the index value into an attribute of the HTML element:

```
<ul>
    <li *ngFor="let item of items; let i = index" [attr.data-index]="i">
        {{item}}
    </li>
</ul>
```

Angular = 1

```
<ul>
    <li *ngFor="#item of items; #i = index" [attr.data-index]="i">
        {{item}}
    </li>
</ul>
```

Here is the updated plunkr: http://plnkr.co/edit/LiCeyKGUapS5JKkRWnUJ?p=preview.

edited Jun 18 at 18:31
tobias47n9e
1,047 3 13 39

answered Feb 15 '16 at 9:29
Thierry Templier
154k 29 335 307

3 In fact, the `attr.` is a syntax to tell Angular2 to set the value of the expression to the attribute name. It's not just like evaluating JSON, I guess ;-) See this link: angular.io/docs/ts/latest/guide/template-syntax.html – Thierry Templier Feb 15 '16 at 9:54

1 Thanks, that does the trick. Also, I meant to put the `ngFor` on the `li` instead of the `ul`. – Vivendi Feb 15 '16 at 10:10

1 I rolled back the recent edit in order to have both the original answer and the new updated code value. I was searching for some early beta stuff and saw that this answer was modified, so it no longer applied to the beta version of angular 2. – ps2goat Nov 17 '16 at 18:58

4 UPDATE: 08/2017 <div *ngFor="let hero of heroes; let i=index; trackBy: trackById" > – Maxi Aug 28 '17 at 22:21

1 as of Oct 17, 2018 Angular 6 is using *ngFor="let item of items; index as i" See Leo's answer below. – Gel Sisaed Oct 18 '18 at 2:33

In Angular 5/6/7:

edited Jan 11 at 21:52

answered Dec 4 '17 at 14:29

Leo
4,367 2 29 35

8 this is it! Working as of Oct 17, 2018 . Thank you. (why they keep changing syntax, so annoying? – Gel Sisaed Oct 18 '18 at 2:31

1 this is a better and simple answer than the first one. – Kenry Sanchez Feb 19 at 19:41

2 `index as i` works great, thank you. But isn't the `*ngFor` meant to go on the element to be repeated (e.g. the `<li>` in this case)? The code you suggested creates multiple `<ul>` s rathers than multiple `<li>` s. – Liran H Apr 8 at 11:54

88

Just an update to this, Thierry's answer is still correct, but there has been an update to Angular2 with regards to:

```
<ul *ngFor="let item of items; let i = index" [attr.data-index]="i">
  <li>{{item}}</li>
</ul>
```

The `#i = index` should now be `let i = index`

EDIT/UPDATE:

The `*ngFor` should be on the element you're wanting to foreach, so for this example it should be:

```
<ul>
  <li *ngFor="let item of items; let i = index" [attr.data-index]="i">{{item}}</li>
</ul>
```

EDIT/UPDATE

Angular 5

```
<ul>
  <li *ngFor="let item of items; index as i" [attr.data-index]="i">{{item}}</li>
```

```
<ul *ngFor="let item of items; index as i">
  <li [attr.data-index]="i">{{item}}</li>
</ul>
```

edited Jun 7 at 6:55

answered Jun 28 '16 at 9:23
Wesley Coetzee
3,304 1 20 38

3 And `#item` should be `let item` ;-) – Günter Zöchbauer Jun 28 '16 at 10:00

Ah yes - that too, sorry I'll go correct that – Wesley Coetzee Jun 28 '16 at 10:01

48

I think its already been answered before, but just a correction if you are populating an unordered list, the `*ngFor` will come in the element which you want to repeat. So it should be insdide `<li>` . Also, Angular2 now uses let to declare a variable.

```
<ul>
    <li *ngFor="let item of items; let i = index" [attr.data-index]="i">
                {{item}}
    </li>
</ul>
```

answered Jul 25 '16 at 20:49
monica
999 9 21

In case of angular 5 right solution – Oswald Jan 8 '18 at 12:07

I think that it is best practice to specify the *ngFor directive in ul tag <ul *ngFor="let item of items; index as i" [attr.data-index]="i"> <li > {{item}} </li> </ul> – bajran Jul 7 '18 at 15:42

1 If you specify ngFor in the ul tag, then the entity that will be repeated will be ul, but here u dont want to repeat the ul, you just want to iterate on the list elements i.e. li. – monica Jul 9 '18 at 11:07

19

```
<ul>
    <li *ngFor="let item of items; let i = index">{{i + 1}}</li>
</ul
```

edited May 21 '18 at 18:21
Chris Farmer
14.8k 28 101 149

answered Nov 10 '17 at 20:59
Alf Moh
4,142 3 24 37

Try this

1

```
<div *ngFor="let piece of allPieces; let i=index">
{{i}} // this will give index
</div>
```

answered Apr 15 at 11:00
Chirag
254 2 11
