How to use *ngIf else?

Asked 2 years, 5 months ago Active 16 days ago Viewed 565k times

519

99

I'm using Angular and I want to use `*ngIf else` (available since version 4) in this example:

```
<div *ngIf="isValid">
  content here ...
</div>

<div *ngIf="!isValid">
 other content here...
</div>
```

How can I acheive the same behavior with `ngIf else` ?

angular if-statement angular-template

edited Oct 8 '18 at 21:15
Alexander Abakumov
5,689 5 50 79

asked Mar 24 '17 at 18:18
kawli norman
2,598 2 6 6

Check here for Angular 8 explained with Example NgIf, NgIf Else and NgIf Then Else freakyjolly.com/… – Code Spy Aug 7 at 15:02

14 Answers

840

Angular 4 and 5:

using `else` :

```
<div *ngIf="isValid;else other_content">
```

```
<ng-template #other_content>other content here...</ng-template>
```

you can also use `then else` :

```
<div *ngIf="isValid;then content else other_content">here is ignored</div>
<ng-template #content>content here...</ng-template>
<ng-template #other_content>other content here...</ng-template>
```

or `then` alone :

```
<div *ngIf="isValid;then content"></div>
<ng-template #content>content here...</ng-template>
```

Demo :

Plunker

Details:

`<ng-template>` : is Angular’s own implementation of the `<template>` tag which is according to MDN :

> The HTML `<template>` element is a mechanism for holding client-side content that is not to be rendered when a page is loaded but may subsequently be instantiated during runtime using JavaScript.

edited Jun 25 '18 at 6:55
Iswar
948 4 23 49

answered Mar 24 '17 at 18:20
Bougarfaoui El houcine
17.2k 4 29 29

8 I hoped there was a way just to use <ng-template> without another tag like div, but oddly it is not... I know the <div> gets removed as you use it, but it's kinda weird as implementation I think. – andreas Mar 30 '17 at 20:03

13 @andreas You can use `<ng-container>` for the if-clause – Martin Schneider Jun 17 '17 at 11:08

2 Note: you can use `ng-container` for the container containing *ngIf, but not for the template – Simon_Weaver Jan 18 '18 at 1:32

@Simon_Weaver I figured it out the hard way. But why? why didn't they allow `*ngIf` to work on `ng-template` ? – Eran Medan Mar 15 '18 at 6:02

149

In Angular 4.x.x You can use ngIf in four way to achieve simple if else procedure:

1. Just Use **If**

```
<div *ngIf="isValid">
    If isValid is true
</div>
```

2. Using **If with Else** (Please notice to **templateName**)

```
<div *ngIf="isValid; else templateName">
    If isValid is true
</div>

<ng-template #templateName>
    If isValid is false
</ng-template>
```

3. Using **If with Then** (Please notice to **templateName**)

```
<div *ngIf="isValid; then templateName">
    Here is never showing
</div>

<ng-template #templateName>
    If isValid is true
</ng-template>
```

4. Using **If with Then and Else**

```
<div *ngIf="isValid; then thenTemplateName else elseTemplateName">
    Here is never showing
</div>

<ng-template #thenTemplateName>
    If isValid is true
</ng-template>

<ng-template #elseTemplateName>
```

> Tip: **ngIf** evaluates the **expression** and then renders the **then** or **else** template in its place when expression is truthy or falsy respectively. Typically the:
>
> - **then** template is the inline template of **ngIf** unless bound to a different value.
> - **else** template is blank unless it is bound.

edited Jul 17 '18 at 17:34
John Henckel
4,099 1 35 50

answered Oct 19 '17 at 15:18
user2577907

It seems like the compiler does not accept `...; else ...` . Probably the `;` should be removed. – slartidan Feb 27 '18 at 15:55

5 in angular-6, I tested with `...; else ...` and it worked – WasiF Sep 7 '18 at 14:32

14

To work with observable, this is what I usually do to display if the observable array consists of data.

```
<div *ngIf="(observable$ | async) as listOfObject else emptyList">
   <div >
        ....
    </div>
</div>
 <ng-template #emptyList>
   <div >
        ...
    </div>
</ng-template>
```

answered Sep 29 '17 at 9:03
Ah Hiang
313 3 11

"bindEmail" it will check email is available or not. if email is exist than Logout will show otherwise Login will show

```
<ng-template #login><li><a routerLink="/login">Login</a></li></ng-template>
```

edited Jul 23 '17 at 13:17

answered Jul 23 '17 at 13:13
Prashant Shrivastava
144 1 7

2 This doesn't work. If it were correct, then it still wouldn't add any value because the accepted answer shows already how to do it. – Günter Zöchbauer Jul 23 '17 at 13:18

4

In Angular 4.0 `if..else` syntax is quite similar to conditional operators in Java.

In Java you use to `"condition?stmnt1:stmnt2"` .

In Angular 4.0 you use `*ngIf="condition;then stmnt1 else stmnt2"` .

edited Jun 2 '17 at 19:25
msanford
7,415 7 47 68

answered Apr 11 '17 at 3:51
Amit Gaikwad
180 1 9

1 looks like Oracle case when expression.. :-) – Peter Jun 30 '17 at 8:09

4 You're referring to the ternary operator which exists in most C-based languages but this is closer to Kotlin's if expressions. – mythz Jun 30 '17 at 11:05

4

ngif expression resulting value won't just be the boolean true or false

if the expression is just a object, it still evaluate it as truthiness.

if the object is undefined, or non-exist, then ngif will evaluate it as falseness.

common use is if an object loaded, exist, then display the content of this object, otherwise display "loading.......".

```
<div *ngIf="!object">
```

```
<div *ngIf="object">
     <!-- the content of this object -->

           object.info, object.id, object.name ... etc.
 </div>
```

another example:

```
  things = {
 car: 'Honda',
 shoes: 'Nike',
 shirt: 'Tom Ford',
 watch: 'Timex'
 };

 <div *ngIf="things.car; else noCar">
  Nice car!
 </div>

<ng-template #noCar>
   Call a Uber.
</ng-template>

 <!-- Nice car ! -->
```

anthoer example:

```
<div *ngIf="things.car; let car">
   Nice {{ car }}!
 </div>
<!-- Nice Honda! -->
```

[ngif template](https://stackoverflow.com/questions/43006550/how-to-use-ngif-else)

[ngif angular 4](https://stackoverflow.com/questions/43006550/how-to-use-ngif-else)

answered Jan 11 '18 at 18:00

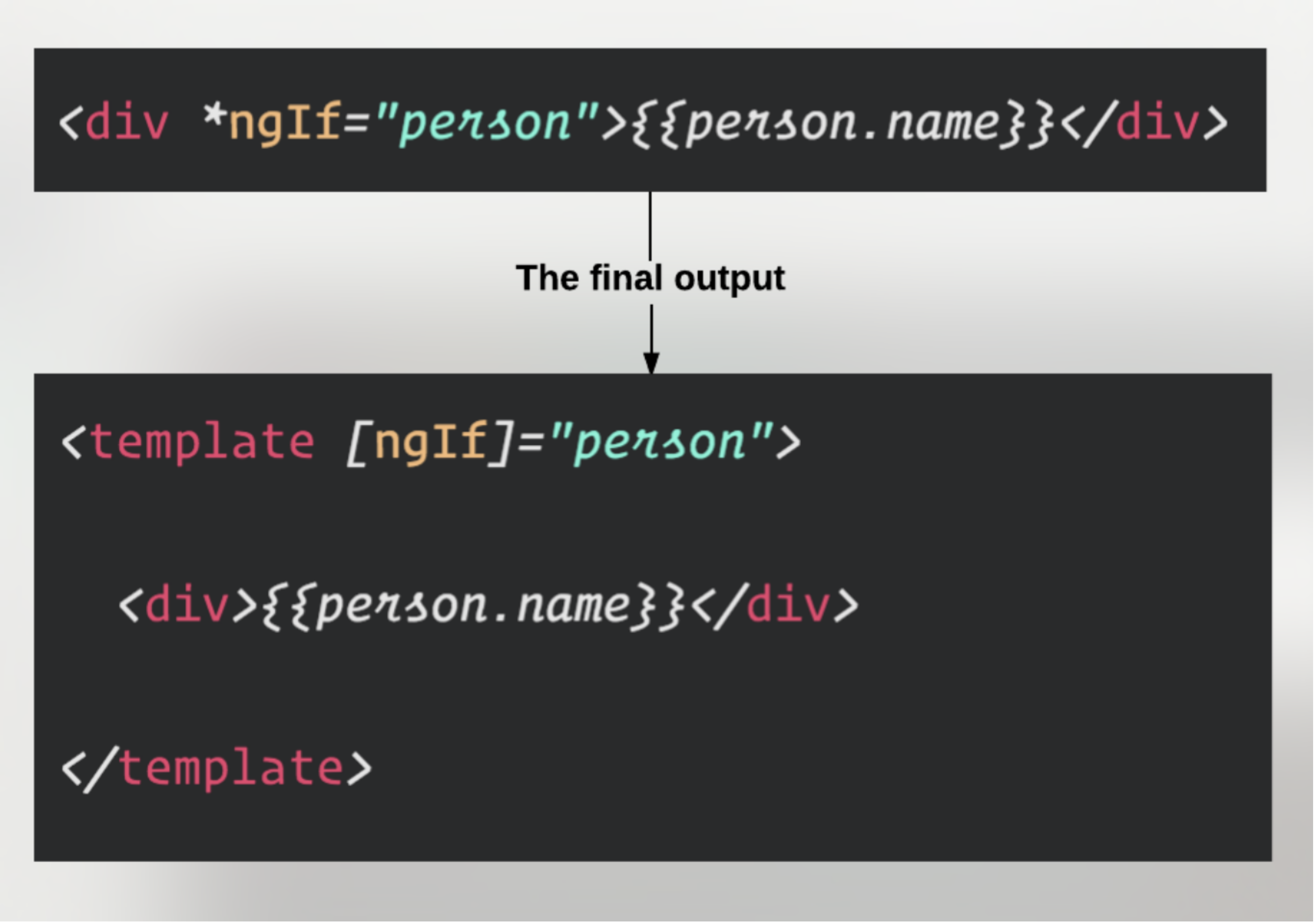
hoogw

1,875 19 18

4

Syntax for ngIf/Else

```
<div *ngIf=”condition; else elseBlock”>Truthy condition</div>
<ng-template #elseBlock>Falsy condition</ng-template>
```



```
<div *ngIf="person">{{person.name}}</div>
```

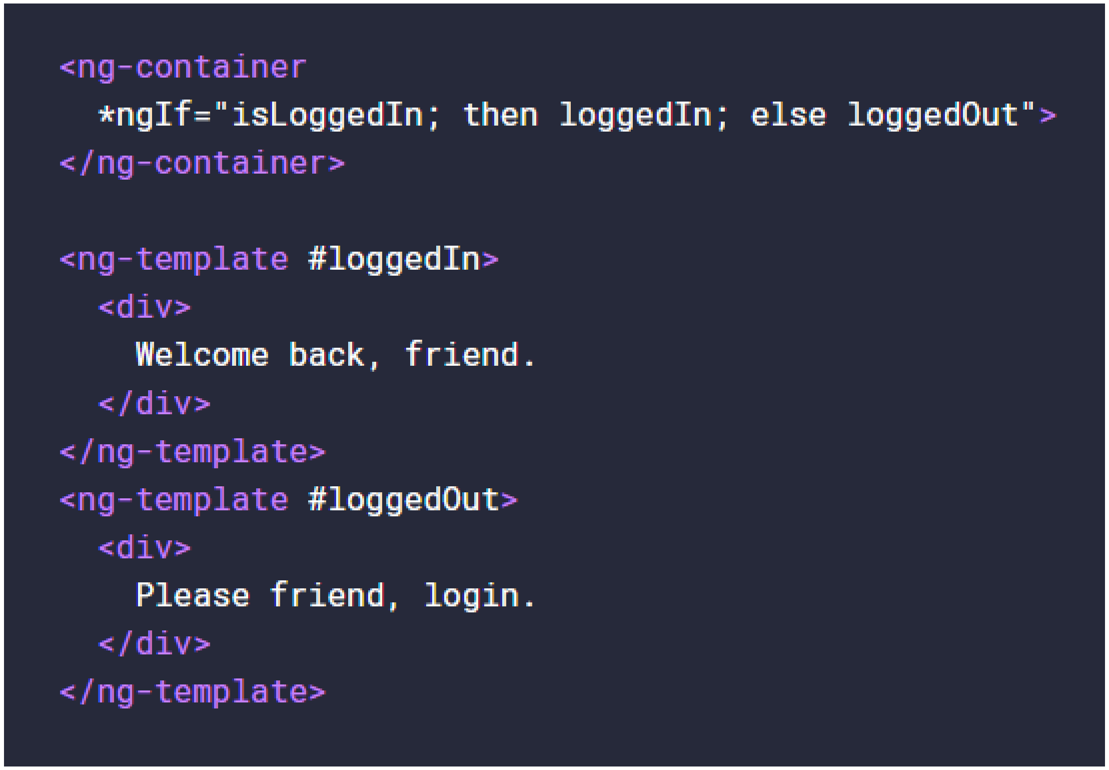
```
<template [ngIf]="person">

  <div>{{person.name}}</div>

</template>
```

Using NgIf / Else/ Then explicit syntax

```
<div *ngIf=”condition; then thenBlock else elseBlock”> ... </div>
<ng-template #thenBlock>Then template</ng-template>
<ng-template #elseBlock>Else template</ng-template>
```

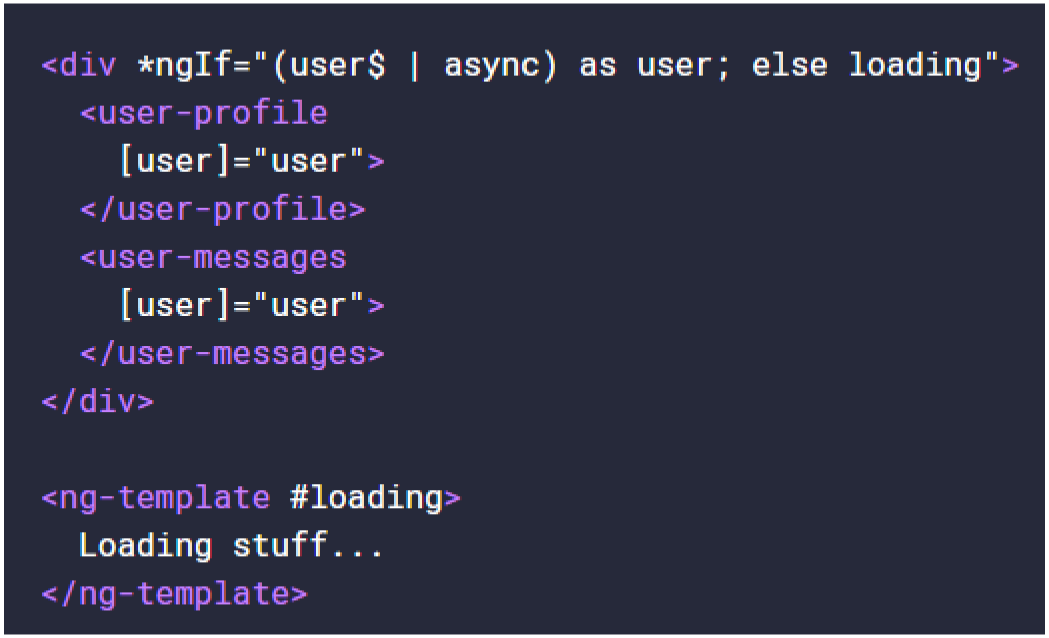


```
<ng-container
  *ngIf="isLoggedIn; then loggedIn; else loggedOut">
</ng-container>

<ng-template #loggedIn>
  <div>
    Welcome back, friend.
  </div>
</ng-template>
<ng-template #loggedOut>
  <div>
    Please friend, login.
  </div>
</ng-template>
```

Observables with NgIf and Async Pipe

For more details



```
<div *ngIf="(user$ | async) as user; else loading">
  <user-profile
    [user]="user">
  </user-profile>
  <user-messages
    [user]="user">
  </user-messages>
</div>

<ng-template #loading>
  Loading stuff...
</ng-template>
```

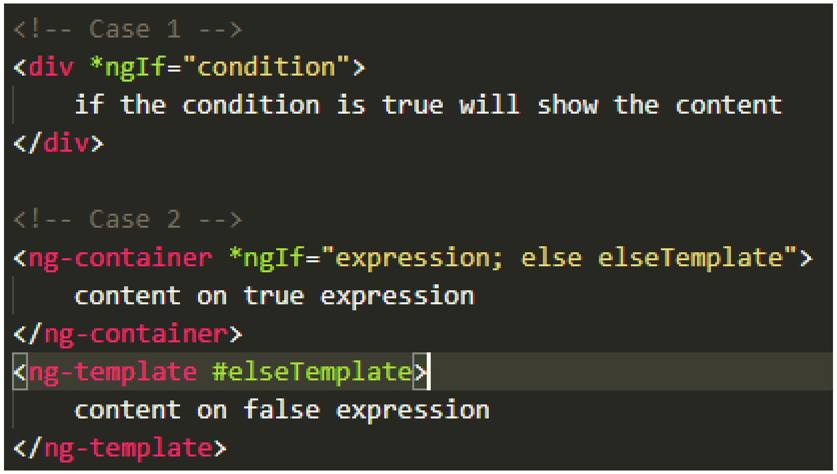
answered Oct 11 '18 at 8:10

Lyes CHIOUKH
2,638 6 22 46

2

There are two possibilities to use if condition on HTML tag or templates:

1. *ngIf directive from CommonModule, on HTML tag;
2. if-else



```
<!-- Case 1 -->
<div *ngIf="condition">
    if the condition is true will show the content
</div>

<!-- Case 2 -->
<ng-container *ngIf="expression; else elseTemplate">
    content on true expression
</ng-container>
<ng-template #elseTemplate>
    content on false expression
</ng-template>
```

answered Nov 13 '18 at 16:30

Rzv Razvan

1,003 1 6 11

2

`ng-template`

```
<ng-template [ngIf]="condition1" [ngIfElse]="template2">
        ...
</ng-template>


<ng-template #template2>
        ...
</ng-template>
```

answered May 18 at 9:03

Aniket

254 1 12

For **Angular 8**

```
export class AppComponent {
  isDone = true;
}
```

1) *ngIf

```
<div *ngIf="isDone">
  It's Done!
</div>

<!-- Negation operator-->
<div *ngIf="!isDone">
  It's Not Done!
</div>
```

2) *ngIf and Else

```
<ng-container *ngIf="isDone; else elseNotDone">
  It's Done!
</ng-container>

<ng-template #elseNotDone>
  It's Not Done!
</ng-template>
```

3) *ngIf, Then and Else

```
<ng-container *ngIf="isDone;  then iAmDone; else iAmNotDone">
</ng-container>

<ng-template #iAmDone>
  It's Done!
</ng-template>

<ng-template #iAmNotDone>
  It's Not Done!
</ng-template>
```

edited Aug 20 at 5:13 answered Aug 7 at 15:08

1

You can also use Javascript short ternary conditional operator ? in angular like this:

```
{{doThis() ? 'foo' : 'bar'}}
```

or

```
<div [ngClass]="doThis() ? 'foo' : 'bar'">
```

answered May 13 at 7:58

Sebastian Viereck

2,775 27 34

0

```
<div *ngIf="this.model.SerialNumber != '';then ConnectedContent else
DisconnectedContent" class="data-font">    </div>

<ng-template #ConnectedContent class="data-font">Connected</ng-template>
<ng-template #DisconnectedContent class="data-font">Disconnected</ng-template>
```

answered Apr 17 '18 at 15:44

Amir Twito

961 1 17 18

0

In Angular 4, 5 and 6

We can simply create a *template reference variable* [2] and link that to the else condition inside an *ngIf directive

The possible *Syntaxes* [1] are:

```
<!-- Only If condition -->
```

```
<!-- If and else conditions -->
<div *ngIf="condition; else elseBlock">...</div>
<!-- or -->
<ng-template #elseBlock>...</ng-template>

<!-- If-then-else -->
<div *ngIf="condition; then thenBlock else elseBlock"></div>
<ng-template #thenBlock>...</ng-template>
<ng-template #elseBlock>...</ng-template>


<!-- If and else conditions (storing condition value locally) -->
<div *ngIf="condition as value; else elseBlock">{{value}}</div>
<ng-template #elseBlock>...</ng-template>
```

DEMO: https://stackblitz.com/edit/angular-feumnt?embed=1&file=src/app/app.component.html

Sources:

1. https://angular.io/api/common/NgIf#syntax
2. https://angular.io/guide/template-syntax#template-reference-variables--var-

answered Aug 8 '18 at 20:24
Kalpesh Panchal
598 5 16

0

I know this has been a while, but want to add it if it helps. The way I went about with is to have two flags in the component and two ngIfs for the corresponding two flags.

It was simple and worked well with material as ng-template and material were not working well together.

answered Sep 20 '18 at 7:29
user1707141
813 3 19 37

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?