# Angular中的内置指令

Angular提供了若干内置指令。因为内置指令是已经导入过的，故可以直接在组件中直接使用它们。

## ngIf

如果你希望根据一个条件来决定显示或隐藏一个元素，可以使用ngIf指令。这个条件是由你传给指令的表达式的结果决定的。

如果表达式的结果返回的是一个假值，那么元素会从DOM上被移除。

下面是一些例子：

<div \*ngIf=”false”></div> <!--nerver displayed-->

<div \*ngIf=”a > b”></div> <!--displayed if a is more than b-->

<div \*ngIf=”str == ‘yes’”></div> <!--displayed if str holds the string “yes” -->

<div \*ngIf=”myFunc()”></div> <!--displayed if myFunc returns a true value -->

## ngSwitch

有时候你需要根据一个给定的条件来渲染不同的元素。

遇到这种情况时，你可能会像下面这样多次使用ngIf：

## ngStyle

## ngClass

## ngFor

## ngNonBindable

当我们想告诉Angular不要编译或者绑定页面中的摸个特殊部分时，要使用ngNonBindable指令。

假如我们想在模板中渲染纯文本{{ content }}，通常情况下，这段文本会被绑定到变量content的值，因为我们使用了{{}}模板语法。

那该如何渲染出纯文本{{ content }}呢？可以使用ngNonBindable指令。

假如我们想要用一个div来与渲染变量content的内容，紧接着输出文本<- this is what {{ content }} rendered来指向变量实际的值。

为了做到这一点，要使用下面的模板：

template:`

<div class='ngNonBindableDemo'>

<span class="bordered">{{ content }}</span>

<span class="pre" ngNonBindable>

&larr; This is what {{ content }} rendered

</span>

</div>

`

有了ngNonBindable属性，Angular不会编译第二个span里的内容，而是原封不动地将其显示出来。

Some text <- This is what {{ content }} rendered