

## Structural directives

---

Structural directives are responsible for HTML layout. They shape or reshape the DOM's *structure*, typically by adding, removing, or manipulating elements.

Structural directives are easy to recognize. An asterisk (\*) precedes the directive attribute name.

The built-in structural directives are

- `NgIf`—conditionally creates or destroys subviews from the template.
- `NgFor`—repeat a node for each item in a list.
- `NgSwitch`—a set of directives that switch among alternative views.

### NgIf directive

---

A structural directive that conditionally includes a template based on the value of an expression coerced to Boolean. When the expression evaluates to true, Angular renders the template provided in a then clause, and when false or null, Angular renders the template provided in an optional else clause. The default template for the else clause is blank.

`NgIf` refers to the directive *class*; `ngIf` refers to the directive's *attribute name*.

A [shorthand form](#) of the directive, `*ngIf="condition"`.

Syntax:

```
<div *ngIf="condition">
    Content to render when condition is true.
</div>
```

Angular translates the `*ngIf` *attribute* into a `<ng-template>` *element*, wrapped around the host element, like below.

```
<ng-template [ngIf]="condition">
    <div>Content to render when condition is true.</div>
</ng-template>
```

Form with else block is

```
<div *ngIf="condition; else elseBlock">
    Content to render when condition is true.
</div>
<ng-template #elseBlock>
    Content to render when condition is false.
</ng-template>
```

## NgFor directive

---

NgFor is a repeater directive—a way to present a list of items.

Syntax:

```
<element *ngFor="let element of array">{{element}}</element>
```

\*ngFor with index:

```
<div *ngFor="let item of items; let i=index">{{i + 1}} - {{item.name}}</div>
```

## NgSwitch directive

---

NgSwitch is like the JavaScript switch statement. It displays one element from among several possible elements, based on a switch condition. Angular puts only the selected element into the DOM.

[NgSwitch](#) is actually a set of three, cooperating directives: [NgSwitch](#), [NgSwitchCase](#), and [NgSwitchDefault](#).

## Template reference variable

---

A template reference variable is often a reference to a DOM element within a template.

Use the hash symbol (#) to declare a reference variable.

Ex:

```
<input name="studentname" #ref>
```

Angular sets the reference variable's value to the element on which it is declared.

Ex: here, "ref.value" returns the value entered into the input field.