

Template reference variables (#var)



Contents

Alternative syntax

A **template reference variable** is often a reference to a DOM element within a template. It can also refer to a directive (which contains a component), an element, [TemplateRef](#), or a [web component](#) [↗](#).

See the [live example](#) / [download example](#) for a working example containing the code snippets in this guide.

Use the hash symbol (#) to declare a reference variable. The following reference variable, #phone, declares a phone variable on an <input> element.

src/app/app.component.html

```
<input #phone placeholder="phone number" />
```



You can refer to a template reference variable anywhere in the component's template. Here, a <button> further down the template refers to the phone variable.

src/app/app.component.html

```
<input #phone placeholder="phone number" />

<!-- lots of other elements -->

<!-- phone refers to the input element; pass its `value` to an event handler -->
<button (click)="callPhone(phone.value)">Call</button>
```



Angular assigns each template reference variable a value based on where you declare the variable:

- If you declare the variable on a component, the variable refers to the component instance.
- If you declare the variable on a standard HTML tag, the variable refers to the element.
- If you declare the variable on an <ng-template> element, the variable refers to a `TemplateRef` instance, which represents the template.
- If the variable specifies a name on the right-hand side, such as `#var="ngModel"`, the variable refers to the directive or component on the element with a matching `exportAs` name.

How a reference variable gets its value

In most cases, Angular sets the reference variable's value to the element on which it is declared. In the previous example, `phone` refers to the phone number `<input>`. The button's click handler passes the `<input>` value to the component's `callPhone()` method.

The `NgForm` directive can change that behavior and set the value to something else. In the following example, the template reference variable, `itemForm`, appears three times separated by HTML.

src/app/hero-form.component.html

```
<form #itemForm="ngForm" (ngSubmit)="onSubmit(itemForm)">
  <label for="name"
    >Name <input class="form-control" name="name" ngModel required />
  </label>
  <button type="submit">Submit</button>
</form>

<div [hidden]="!itemForm.form.valid">
  <p>{{ submitMessage }}</p>
</div>
```

The reference value of `itemForm`, without the `ngForm` attribute value, would be the [HTMLFormElement](#). There is, however, a difference between a Component and a Directive in that a Component will be referenced without specifying the attribute value, and a Directive will not change the implicit reference (that is, the element).

However, with `NgForm`, `itemForm` is a reference to the [NgForm](#) directive with the ability to track the value and validity of every control in the form.

The native `<form>` element doesn't have a `form` property, but the `NgForm` directive does, which allows disabling the submit button if the `itemForm.form.valid` is invalid and passing the entire form control tree to the parent component's `onSubmit()` method.

Template reference variable considerations

A template *reference* variable (`#phone`) is not the same as a template *input* variable (`let phone`) such as in an `*ngFor`. See [Structural directives](#) for more information.

The scope of a reference variable is the entire template. So, don't define the same variable name more than once in the same template as the runtime value will be unpredictable.

Alternative syntax

You can use the `ref-` prefix alternative to `#`. This example declares the `fax` variable as `ref-fax` instead of `#fax`.

src/app/app.component.html

```
<input ref-fax placeholder="fax number" />
<button (click)="callFax(fax.value)">Fax</button>
```

RESOURCES

[About](#)

[Resource Listing](#)

[Press Kit](#)

[Blog](#)

[Usage Analytics](#)

HELP

[Stack Overflow](#)

[Gitter](#)

[Report Issues](#)

[Code of Conduct](#)

COMMUNITY

[Events](#)

[Meetups](#)

[Twitter](#)

[GitHub](#)

[Contribute](#)

LANGUAGES

[简体中文版](#)

[正體中文版](#)

[日本語版](#)

[한국어](#)

Super-powered by Google ©2010-2020. Code licensed under an MIT-style License. Documentation licensed under CC BY 4.0.

Version 10.0.10-local+sha.b32126c335.