

# Template expression operators

Marked for archiving

To ensure that you have the best experience possible, this topic is marked for archiving until we determine that it clearly conveys the most accurate information possible.

In the meantime, this topic might be helpful: [Hierarchical injectors](#).

If you think this content should not be archived, please file a [GitHub issue](#).

The Angular template expression language employs a subset of JavaScript syntax supplemented with a few special operators for specific scenarios.

See the for a working example containing the code snippets in this guide.

{@a non-null-assertion-operator}

## The non-null assertion operator ( `!` )

When you use TypeScript's `--strictNullChecks` flag, you can prevent the type checker from throwing an error with Angular's non-null assertion operator, `!`.

The Angular non-null assertion operator causes the TypeScript type checker to suspend strict `null` and `undefined` checks for a specific property expression.

For example, you can assert that `item` properties are also defined.

Often, you want to make sure that any property bindings aren't `null` or `undefined`. However, there are situations in which such states are acceptable. For those situations, you can use Angular's non-null assertion operator to prevent TypeScript from reporting that a property is `null` or `undefined`.

The non-null assertion operator, `!`, is optional unless you turn on strict null checks.

For more information, see TypeScript's [strict null checking](#).

{@a any-type-cast-function}

## The `$any()` type cast function

Sometimes a binding expression triggers a type error during [AOT compilation](#) and it is not possible or difficult to fully specify the type. To silence the error, you can use the `$any()` cast function to cast the expression to the [any type](#) as in the following example:

Using `$any()` prevents TypeScript from reporting that `bestByDate` is not a member of the `item` object.

The `$any()` cast function also works with `this` to allow access to undeclared members of the component.

The `$any()` cast function works anywhere in a binding expression where a method call is valid.