

# Component Communication

## 4. Template Reference Variable

# Template Reference Variable

- In the component template you can place # to access Dom elements, components, and directives
- TRV is used to access public properties and methods of an instance

```
@Component({
  selector: 'app-root',
  template: `
    <h1 #hello>hello: HtmlHeadingElement</h1>

    <!-- inputModel: NgModel -->
    <input type="text" ngModel #inputModel="ngModel" />

    <!-- childComponent: ChildComponent -->
    <academeez-child #childComponent></academeez-child>
  `,
})
```

# Default behaviour

- If Template reference variable (TRV) is placed on a DOM element we will get that DOM element (even if there is a directive on that element)
- If TRV is placed on a component, we will get the component instance even if there is a directive on the component as well.

# TRV - Directive

- In order to get a directive, you will have to specify the **exportAs**

## Parent

```
@Component({
  selector: 'app-root',
  template: `
    <!-- fooDirective: FooDirective -->
    <academeez-child #fooDirective="academeezFoo" foo>
    </academeez-child>
  `,
})
```

## Directive

```
@Directive({
  selector: '[foo]',
  exportAs: 'academeezFoo'
})
```

# Summary

- With @Input, @Output we cannot access the inner state and methods of a component or directive, with TRV we can do that
- A parent component can place TRV # on a Dom element a component or a directive with
- To get a directive we will have to place **exportAs** in the directive metadata

# Thank You

**Next Lesson: 5. @ViewChild**