A template reference variable is often a reference to a DOM element within a template. It can also be a reference to an Angular component or directive or a web component (Read more at Angular.io). That means you can easily access the variable anywhere in the template.

You declare a reference variable by using the hash symbol (#). The **#firstNameInput** declares a **firstNameInput** variable on an `<input>` element.

```
<input type="text" #firstNameInput><input type="text"
#lastNameInput>
```

After that, you can access the variable anywhere inside the template. For example, I pass the variable as a parameter on an event.

```
<button (click)="show(lastNameInput)">Show</button>
```

Remember that the lastNameInput belongs to HTMLInputElement type.

```
show(lastName: HTMLInputElement){
    console.log(lastName.value);
}
```

Usually, the reference variable can only be accessed inside the template. However, you can use **ViewChild** decorator to reference it inside your component.

```
import {ViewChild, ElementRef} from '@angular/core';// Reference
firstNameInput variable inside Component
@ViewChild('firstNameInput') nameInputRef: ElementRef;
```

After that, you can use ***this.nameInputRef*** anywhere inside your Component.

```
show(lastName: HTMLInputElement){
  this.fullName = this.nameInputRef.nativeElement.value + ' ' +
lastName.value;
}
```

Working with <ng-template>

In the case of ng-template, it is a little bit different because each template has its own set of input variables. For example:

```
// app.component.html
<ng-template #testingTemplate let-fullName="fullName">   <div>Full
Name (template): {{fullName}}</div></ng-template><ng-container
*ngTemplateOutlet="testingTemplate;context:ctx"></ng-container>
```

- We use the prefix ***let-*** to declare the input variable ***fullName***
- this variable ***fullName*** is visible inside the ***ng-template***, not the outside
- In order to access the variable inside ***ng-template***, we have the declare the context

```
// app.component.ts
export class AppComponent  {   fullName: string;
   ctx = {fullName: ''}
   ...   show(lastName: HTMLInputElement){
      this.fullName = this.nameInputRef.nativeElement.value + ' ' +
lastName.value;
      this.ctx.fullName = this.fullName;
   }}
```