Angular - Expressions and Operatives

29 September 2021 10:39

Trainers: Christophe Ramsamy
Bilkiss Dulloo

[ngClass]

Adds and removes CSS classes on an HTML element.

Exported from: CommonModule

Syntax:

```
<some-element [ngClass]="{'class1 class2
class3' : true}">...</some-element>
```

Usage Example: An input field with validations being processed by adding attributes to the 'is-invalid' CSS property via the [ngClass]

```
<input formControlName="description"
type="text" class="form-control"
[ngClass]="{'is-invalid':
carForm.controls.description.invalid &&
carForm.controls.description.dirty}">
```

*ngIf

A structural directive that conditionally includes a template based on the value of an expression coerced to Boolean.

Exported from: CommonModule

Syntax: *ngIf="condition"

Usage Example: Toggle elements depending on specific conditions

```
<button class="btn"
*ngIf="storage.get(item.name)"
(click)="#">
     <i class="fa fa-heart" ></i>
</button>

<button class="btn" (click)="#"
*ngIf="!storage.get(item.name)">
     <i class="fa fa-heart-o fav" ></i>
</button>
```

<ng-template>

Element that defines a template that is not rendered by default. Can then be instructed and called in HTML tags providing full control over how and when to display its contents.

Imported from: @angular/core

Usage Example: Text in a ng-template to be displayed as a tooltip on a button.

```
  <ng-template #btnDelete>
    <p>
      New User? <span class="text-danger">
Click to Register!</span>
    </p>
  </ng-template>

//using the template id(#btnDelete) in the [tooltip] for
//fetching the text
<button *ngIf="!validLogin" class="btn"
[tooltip]="btnDelete">
</button>
```

*ngFor

A structural directive that renders a template for each item in a collection.

Exported from: CommonModule

Usage Example:

```
<ng-container *ngFor="let msg of messages">
   <div class="col-3 text-end">
{{msg?.username}} : </div>
   <div class="col-9">{{msg.message}}</div>
</ng-container>
```

[ngContainer]

Can be used to hold directives without creating an HTML element. Also, In order to avoid having to create that extra div.

Imported from: @angular/core

Usage Example: avoiding the use of additional divs to display content inside other divs

```
<div class="row row-cols-2">
   <ng-container>
     <div class="col-3 text-end">
{{msg?.username}} :
    </div>
   </ng-container>
</div>
```

[(ngModel)]

Creates a FormControl instance from a domain model and binds it to a form control element.

Exported from: FormsModule

Usage Example:

```
<input type="text" class="form-control"
(keyup)="searchCar()" aria-
label="Amount (to the nearest dollar)"
          placeholder="Search car"
[(ngModel)]="searchText">
```