Angular Forms

Angular forms are the most common and important for many web applications and are used for getting data from users. User input data are taken from the form, we may need to validate it and at last, data in the form is saved to database at server.

## Angular form types

There are two ways to build a form in angular, we can either use the template form or reactive form. Both these approaches are part of the core @angular/forms library, but they also have their own separate module  
1. Template drive form (FormsModule)  
2. [Reactive form (ReactiveFormsModule)](https://edupala.com/angular-reactive-form-validation/)

## ****Angular  Template Forms****

Template-driven forms are useful for adding the simple form to an app, such as the email list signup form. They’re easy to add to a web app, but they don’t scale as well as the reactive forms.

If you have the requirement of basic form and logic that can be managed solely in the template, use template forms.

In template forms, most of the work is done in the view template.

## ****Angular Reactive Forms****

Reactive forms or Model-driven forms are more robust, scalable, reusable, and testable. If forms are the key part of your application, or you’re already using reactive patterns for building your web application, use reactive forms.

In Reactive Forms, most of the work is done in the component class.

One of the advantages of the Model-driven approach is that the Validations become very easy.

In Reactive Forms We use Angular FormBuilder to create a FormGroup object (form property) which is then bound to the template <form> element (using the [formGroup] directive later).

Validators provides a set of built-in validators (required, minLength, maxLength…) that can be used by form controls.

A validator is a function that processes a FormControl or collection of controls and returns an error map or null (meaning validation has passed).

In both types of form, we have common building blocks as FormControl, FormGroup, and FormArray. Now we might ask which is better, neither is truly better than the other, both have their own advantage and disadvantage. Based on our requirement we can use it, in the case of simple form, we can use a template, and more organized and complex form we can use the reactive form approach.

## Using angular CSS classes to highlight invalid form fields

If showing a plain error message is not enough, we can also adjust the appearance of each form-control using CSS.

For that, angular automatically assigns some classes to each control, depending of its status.

.ng-valid

.ng-invalid

.ng-pending

.ng-pristine

.ng-dirty

.ng-untouched

.ng-touched