Difference between Reactive Forms and Template-driven Forms

|  |  |  |
| --- | --- | --- |
| **Comparison Index** | **Reactive Forms** | **Template-driven Forms** |
| Setup (form model) | Reactive forms are more explicit. They are created in component class. | Template-driven forms are less explicit. They are created by directives. |
| Data model | Structured | Unstructured |
| Predictability | Synchronous | Asynchronous |
| Form validation | Functions | Directives, ngForm,ngModel,ngsubmit |
| Mutability | Immutable | Mutable |
| Scalability | Low-level API access | Abstraction on top of APIs |

Similarity between Reactive Forms and Template-driven Forms

There are some building blocks which are shared by both reactive and template-driven forms:

* **FormControl:** It is used to track the value and validation status of an individual form control.
* **FormGroup:** It is used to track the same values and status for a collection of form controls.
* **FormArray:** It is used to track the same values and status for an array of form controls.
* **ControlValueAccessor:** It is used to create a bridge between Angular FormControl instances and native DOM elements.

**Reactive Forms:**

1. **import** { Component } from '@angular/core';
2. **import** { FormControl } from '@angular/forms';
3. @Component({
4. selector: 'app-reactive-favorite-color',
5. template: `
6. Favorite Color: <input type="text" [formControl]="favoriteColorControl"> `
7. })
8. export **class** FavoriteColorComponent {
9. favoriteColorControl = **new** FormControl('');
10. }

TemplateDrivenForms:

1. **import** { Component } from '@angular/core';
2. @Component({
3. selector: 'app-template-favorite-color',
4. template: `
5. Favorite Color: <input type="text" [(ngModel)]="favoriteColor"> `
6. })
7. export **class** FavoriteColorComponent {
8. favoriteColor = '';
9. }