Template: uses HTML (template) with two way data binding

Reactive: manages the form and its data in the component code

A screenshot of a web page

Description automatically generated

A diagram of state

Description automatically generated

Pristine: unchanged (all must be pristine to be considered pristine)

Dirty: changed (if one input element is dirty, then the whole form is dirty)

A black text on a white background

Description automatically generated

Form Model:

-Data structure that represents the HTML form

-don’t confuse the form module with the data model that we use with data binding

A diagram of a form

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated

A diagram of a group

Description automatically generated

A close-up of a chart

Description automatically generated

A close-up of a diagram

Description automatically generatedA diagram of a complex scenario

Description automatically generated

Template:

-Angular generates the form model by creating FormGroup and FormControl instances

Reactive

-We create the form model by creating FormGroup and FormControl ourselves in our component class

A screenshot of a computer code

Description automatically generated-Then bind the template to the form model, this means that our form is not directly modifying our data model

Form Model

-requires one FormGroup (root) that represents the entire form

A screenshot of a phone

Description automatically generated

A screenshot of a computer program

Description automatically generated

-Could have put it in the constructor, but selected to use ngOnInit to ensure the component and template are initialized before building the form model A screen shot of a computer program

Description automatically generated

-Inside FormGroup,, we pass in an object that contains FormControls and nested FormGroups



-Each control is defined with a key (control name), value (AbstractControl) pair.

-AbstractControl is the base class for FormControl and FormGroup

-This structure (orange box) is the form model and tracks the form value and state

-Don’t confuse this form model (defines the set of FormGroups/Controls that match up with the HTML form and input elements) with our data model (customers = new Customer()) which defines the data passed to and from a back-end server

A screenshot of a computer program

Description automatically generated

formGroup

-use this directive to bind the form element in the template to the root FormGroup of our form model (customerForm from our component class)

- The form then knows not to build its own form model

formControlName

-to bind each input elements to its associated FormControl

A screenshot of a computer program

Description automatically generated

A diagram of a form builder

Description automatically generatedA screenshot of a computer code

Description automatically generated

Patch values: if we only want to subset of the values

A close-up of a logo

Description automatically generated

FormBuilder

-is a class we can use to build reactive forms

-think of it as a factory that creates FormGroups/Controls

A screenshot of a computer code

Description automatically generated

-The group method returns an initialized root FormGroup instance with all associated FormControls and nested FormGroups

-First element of the array is the default value expression or the object with the value and disabled state

A computer code with text

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generatedA close-up of a computer screen

Description automatically generated

Validation

A screenshot of a computer code

Description automatically generated

-Validators class and specify the name of the desired validation rule

-use an array to specify multiple validation rules

-3rd element of the array is asynchronous validation and is used most for calling server-side validation method

-To minimize asynchronous calls, asynchronous validators are not executed until all synchronous validators pass validation

A screenshot of a computer

Description automatically generated

-Updating the validators doesn’t cause the validation status of the control to be reevaluated. If we want the control validation to be evaluated, call updateValueAndValidity

A screenshot of a computer code

Description automatically generated

-Validator functions takes one parameter,, the FormControl or FormGroup being validated

-We type the parameter using the abstract class so we can use control or group

A screen shot of a computer

Description automatically generated

-returns null if valid

-The key is a string and defines the name of the broken validation rule

-If the validator will only be used by this component, then add it above the @Component otherwise put it inside its own file

A screenshot of a computer program

Description automatically generated

Because a validator functions can only take one parameter (the AbstractControl), so we need to build a factory function that returns the validator function that returns the validator function

A screen shot of a computer code

Description automatically generated

-Common use case is start, end date or confirming passwords

A close-up of a sign

Description automatically generated

-since it is nested, must reference like this

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generatedA screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

Name=”firstName”

So the formcontrol instance is associated correctly in the form model

#firstNameVar=”ngModel”

Template reference variable to use for validation styling and messages

When to create a instance or a Class?

A blue and white box with black text

Description automatically generated