Cheat Sheet – Forms

Template-driven vs Data-driven

Angular 2 forms can be created via two ways: By using the template-driven approach or the data-driven approach.
In the first approach (template-driven), Angular 2 will automatically create a form with which it works by identifying the <form> tag.
You will have to assign controls to this form by adding the `ngControl` directive to the HTML elements which should be form inputs/ controls.

In the data-driven approach, you create the form on your own (in the component body) and then assign it to the HTML code by using `ngFormModel` for the form and `ngFormControl` for the individual controls.

Template-driven Approach

Angular 2 derives the form for you, whenever it encounters a <form> tag in your template. You still have to set up the controls by assigning `ngControl` directives.

Data-driven Approach

You create the form, using the FormBuilder, which has to be injected into your component.
Forms are then created like this:

```
this.myForm = this.formBuilder.group({
     control1: [‘Default value’, VALIDATORS, ASYNC_VALIDATORS],
     control2: [‘’]
});
```

Custom Validators

You may not only use built-in validators (overview: https://angular.io/docs/ts/latest/api/common/index/Validators-class.html) but also build your own ones.
You may create functions which can be passed as validators inside your component body, in static classes or as global functions.
Make sure to only pass a reference to the function and not execute it instead.

Useful Resources

Basic Tutorial: https://angular.io/docs/ts/latest/guide/forms.html
Template-driven approach:
http://blog.thoughtram.io/angular/2016/03/21/template-driven-forms-in-angular-2.html
The Ultimate Guide to Angular 2 Forms: http://blog.ng-book.com/the-ultimate-guide-to-forms-in-angular-2/