**The Basics**

**How Angular App gets Loaded ?**

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Angular remplace la balise <app-root> dans index.html par ce qui lui est demandé via app.component.ts

Une image contenant texte

Description générée automatiquement

Dans **app.module.ts**, on trouve la clé bootstrap avec un tableau contenant AppComponent. This bootstrap array basically lists all the component swhich should be known to Angular at the point of time it analyzes our index.html file.

So ***main.ts*** triggers the ***app.module.ts***, in this ***AppComponent*** in the bootstrap array, angular will reads the setup we pass in the ***app.component.ts*** and therefore knows this selector ‘***app-root’*** and now Angular is able to handle ***app-root*** in the ***index.html*** file and it knows the following informations .

**Creating a new component :**

Une image contenant texte

Description générée automatiquement create the component, and add the @Component decorator. Then add this component in the declarations section in the **app.module.ts**

Une image contenant texte

Description générée automatiquement

- Then in another component you can nest it, by calling <app-server> tag in the ***app.component.html*** file.

Une image contenant texte

Description générée automatiquement

* You can also do it inline, like this :
  + Une image contenant texte

    Description générée automatiquement
* Or you can generate the component autimatically using the CLI wit the following command :
  + **ng g c componentName** (ng generate component servers)
* templateUrl 🡪 point to an external html template
* template 🡪 inline template

**Working with component styles :**

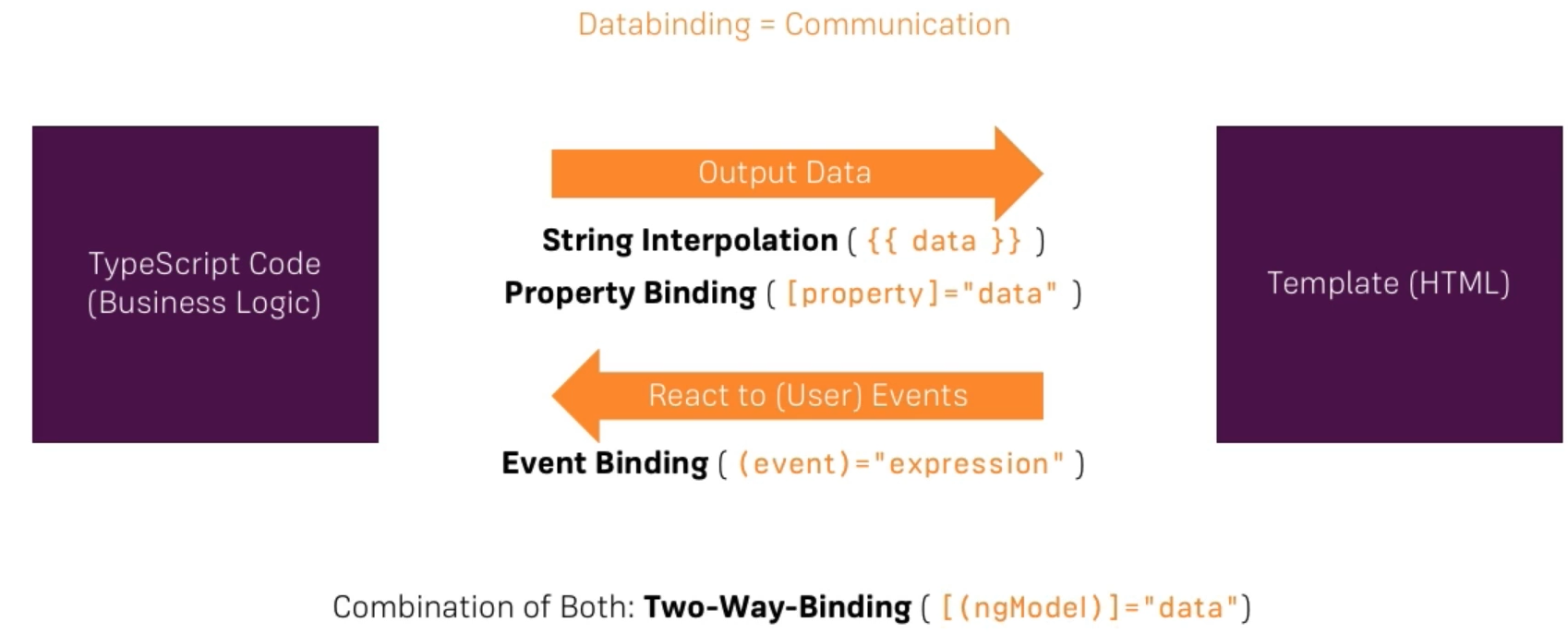
* you can choose to do it inline

Une image contenant texte

Description générée automatiquement

* Or external file in the app.component.css file

**Data Binding :**



**Data Binding – String interpolation**

Une image contenant texte

Description générée automatiquement



**Data Binding – Property Binding**

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

* Each HTML element you use is parsed by the browser and kind of translated into an element on the document object model. And therefore, we have an element in this DOM and this element has a couple of properties. A lot of these can’t even be set through attributes on the HTML element.

**Event Binding :**

* For events, like onClick, … . First, we create the onCreateServer method in the component. Then, in the html we call it via event Binding using parenthesis. Look example below ;

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Passing and using data with event Binding :

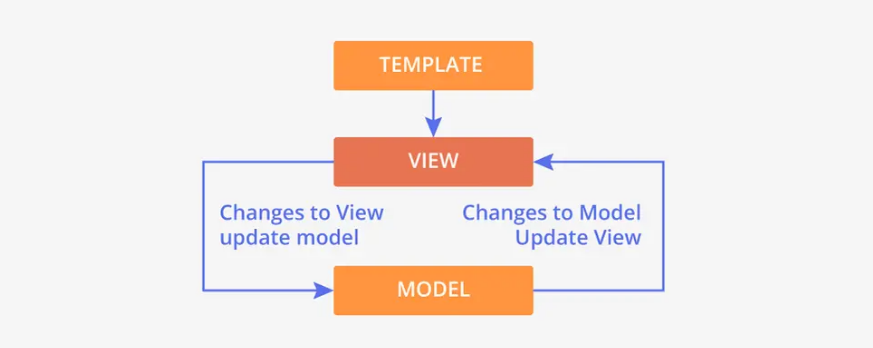
Une image contenant texte

Description générée automatiquement



* $event is a reserved variable name, you can use in the template when using event binding. Here $event will simply be the data emitted with that event. Input and Click are default events provided by the DOM, and they ship some data when they are fired.

**Two-Way Data Binding :**



* Two-way binding using [(ngModel)] attribute.

Une image contenant texte

Description générée automatiquement

**Directives :**

Are instructions in the DOM, e.g. are components. Angular please add our component in this place.

* Below example with ngIf, it is like an if else statement, so when the serverCreated variable will be true, then it will show the p tag ; and when it is false, then it will not be created.

Une image contenant texte

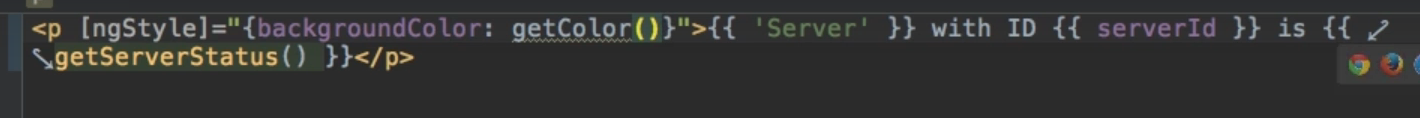
Description générée automatiquement

* You can add an else statement, like in the following example.

Une image contenant texte

Description générée automatiquement

**Styling elements dynamically with ngStyle :**



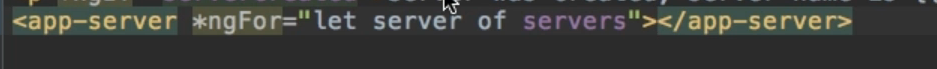
**Apply CSS class Dynamically with ngClass :**

Une image contenant texte

Description générée automatiquement

**Outputting Lists with ngFor :**

* Here we use ngFor built-in directive : here servers is the list



**Getting index using ngFor :**

Une image contenant texte

Description générée automatiquement