

**Atelier n°1 :**

## **Manipulation de composant (partie1)**

### **Pré requis :**

- Environnement mis en place
- Un projet Angular qui tourne

### **Objectifs :**

- Identifier les différents éléments d'une application Angular
- Créer et manipuler un composant
- Appliquer l'interpolation, eventBinding et property binding
- Définir un modèle de données avec TypeScript
- Manipuler les directives structurales ngFor\* et ngIf\*
- Employer des expressions Angular

### **Exercice 1:**

- 1- Créez un composant **FournisseurComponent**.
- 2- Définissez les propriétés **idFournisseur**, **code** et **libelle** dans la classe du composant crée.
- 3- Initialisez les propriétés de ce composant par :

```
idFournisseur = 105;  
code = 'A104B89';  
libelle = 'MyTeck';
```

- 4- Affichez les informations de ce Fournisseur dans le template du composant comme suit, voir Figure 1.

fournisseur works!

- IdFournisseur: 105
- Code: A104B89
- Libelle: MyTeck

Figure 1

NB : toutes les informations sont écrites à l'intérieur d'une division « div ».

- 5- Ajoutez une propriété appelé **hide** de type booléen pour ce composant et initialisez sa valeur par « false ».
- 6- Ajoutez un bouton « Cacher » au-dessous des informations du fournisseur et en dehors de la division. En cliquant sur ce bouton la valeur de la propriété hide change de false à true et s'affiche dans la console du navigateur.
- 7- Appliquez la valeur de la propriété hide sur l'attribut « hidden » de la « div » contenant les informations du fournisseur de sorte que si hide=false la division est affichée sinon la division est cachée.

Résultat attendu : au début les informations sont affichées, en cliquant sur le bouton toute la division contenant les informations est cachée.

## Exercice 2 :

- 1- Créez un composant appelé « **ListFournisseurComponent** ».
- 2- Définissez un modèle de données appelé « **Fournisseur** » dans un fichier models/fournisseur.ts comme suit :

```
{
  idFournisseur : number,
  code : string,
  libelle : string,
}
```

- 3- Ajoutez une propriété appelée **list** de type **tableau de fournisseur** « **Fournisseur[]** » et initialisez la comme suit :

```
[
  {idFournisseur:1,code:"1ABC4522",libelle:"hp"},
  {idFournisseur:2,code:"2ABC4522",libelle:"dell"},
  {idFournisseur:3,code:"3ABC4522",libelle:"lenovo"},
  {idFournisseur:400,code:"40ABC452",libelle:"asus"}
]
```

- 4- Au niveau de la vue du composant **ListFournisseurComponent**, affichez la liste des fournisseurs récupérés depuis la propriété **list** comme indiqué dans la Figure 2.

IdFournisseur : 1
Code: 1ABC4522
Libelle : hp
IdFournisseur : 2
Code: 2ABC4522
Libelle : dell
IdFournisseur : 3
Code: 3ABC4522
Libelle : lenovo
IdFournisseur : 400
Code: 40ABC452
Libelle : asus

Figure 2

- 5- Utilisez la directive \*ngIf pour afficher les fournisseurs dont le idFournisseur est inférieur ou égale à 100.
- 6- Pour les fournisseurs dont le libelle est égal à 'hp', l'arrière-plan du fieldset correspondant est vert, pour le reste il est jaune. Manipulez pour cela la propriété background-color du fieldset.
- 7- Si le code du fournisseur commence par '2A' alors la valeur du code s'affiche en gras. Manipulez pour cela la propriété font-weight et utilisez une méthode getColor(), que vous définissez vous-même, pour faire ce traitement.
- 8- En **double cliquant** sur le fieldset d'un fournisseur, ce dernier est supprimé et la nouvelle liste est réaffichée.