

## TP Multiplication

Lien : <https://github.com/LaxhP/multiplication>

Pour commencer, j'ai créé un composant ident. Dans la vue du composant j'ai créé un formulaire qui envoie deux paramètres au composant: un nombre qui va correspondre à la multiplication affichée (partie 1) et un nombre de table à afficher (partie 1).

```
<form [formGroup]="identForm" (ngSubmit)="choixNb()">
<div class="field">
<label class="label">Entrez un nombre</label>
<div class="control">
<input type="number" placeholder="chiffre" value="" FormControlName="chiffre">
</div>
```

Lorsqu'on va remplir le formulaire, la fonction choixNb() va s'exécuter. Elle va envoyer les paramètres au composant parent (app).

```
choixNb() {
  let chiffre = this.identForm.value.chiffre;
  console.log("le chiffre choisi est :" + chiffre);
  this.leChiffre.emit(this.identForm.value.chiffre);

  @Output() leChiffre = new EventEmitter<number>();
```

Dans le composant parent, on récupère le paramètre grâce aux événements, puis on le renvoie dans le composant table-multiplication.

```
<app-ident (leChiffre)="onChiffreAdded($event)" (nbTable)="nbTableAdded($event)">
</app-ident>
<app-table-multiplication [chiffre]="chiffre" ></app-table-multiplication>
```

On va le récupérer grâce au input, puis on va l'utiliser dans la fonction multiplication qui va retourner un tableau avec les résultats des produits.

```
@Input() chiffre!: number;

multiplication(): number[] {
  var table: number[] = [];
  if (this.chiffre) {
```

```

    for (let i = 1; i <= 10; i++) {
      let x = this.chiffre * i;
      table.push(x);
    }
    return table;
  }
  return table;
}

```

Puis dans la vue on affichera le tableau.

```

<table>
  <tr *ngFor="let x of multiplication(); let i = index">
    <td>{{chiffre}} x {{i+1}} = {{x}}</td>
  </tr>
</table>

```

Le fonctionnement est similaire pour `tables-multiplication`. On va récupérer `nbTable` depuis le composant `app`. Puis on va appeler le composant `table-composant` directement depuis `tables-composant`.

```

<div *ngFor='let in of counter(nbTable + 1) ;let i = index'>
  <app-table-multiplication [chiffre]="i"></app-table-multiplication>
</div>

```

Pour chaque `i` (de 1 à `nbTable`) on affichera la table de multiplication.

