

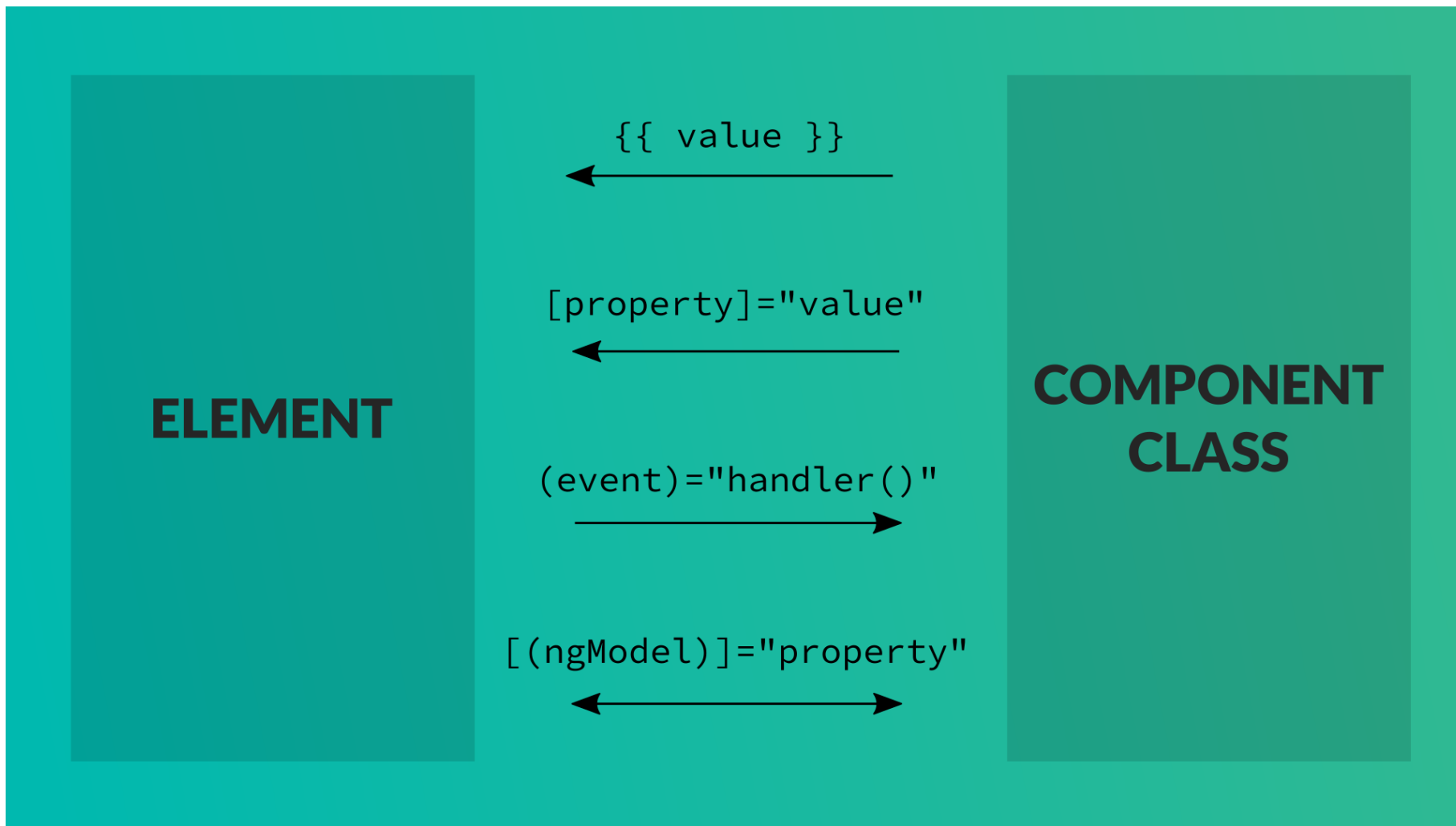


DATA BINDING

LIAISON DE DONNÉES

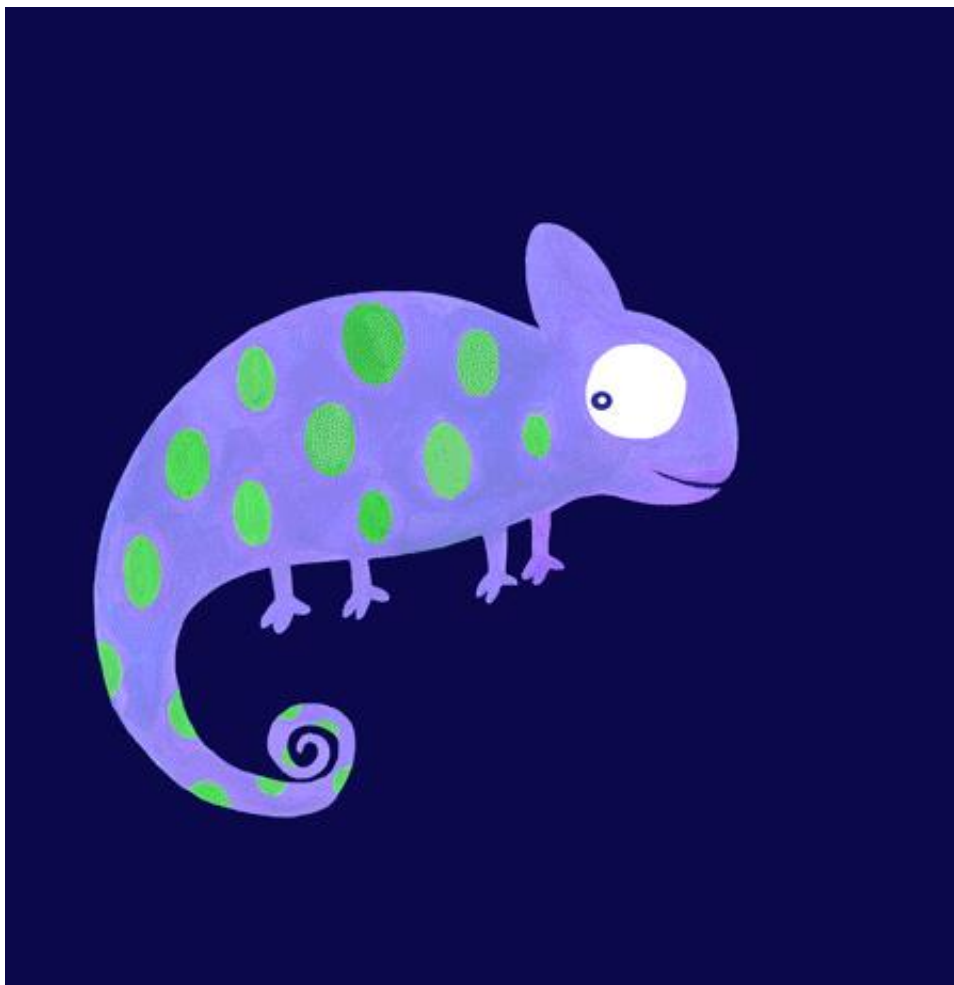


Le principe



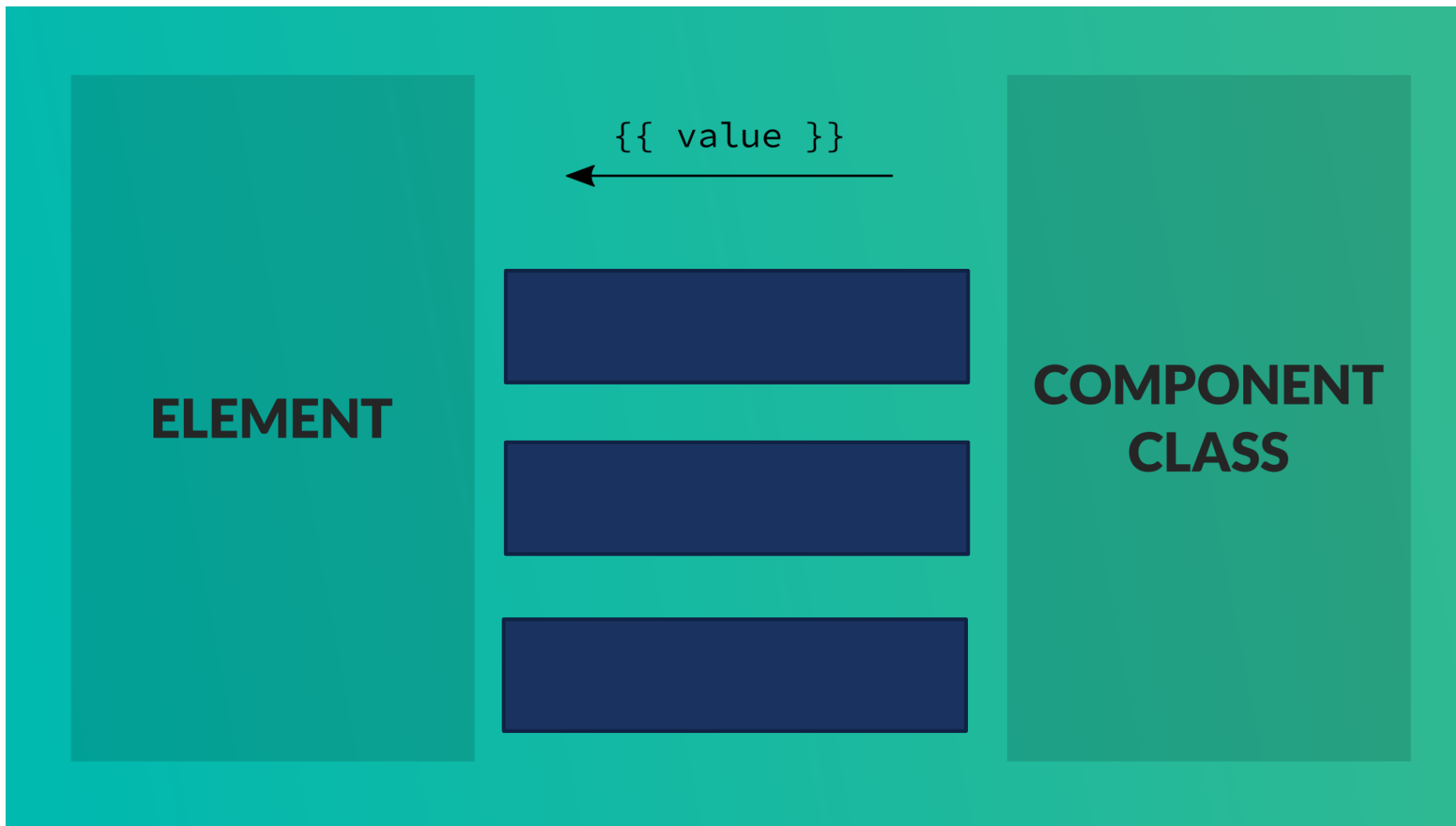


Le but





String Interpolation





String Interpolation

▼ cars-list

cars-list.component.css

U

<> cars-list.component.html


U

TS cars-list.component.ts


U





Retrouvez les exemples sur Github **Amar2050**


 Amar2050 / Drive-X-Angular


 Code

 Issues

 Pull requests

 Actions

 main

 Commits on Oct 14, 2020

Data Binding - String Interpolation Exemple 4



Amar2050 committed 1 minute ago

Data Binding - String Interpolation Exemple 3



Amar2050 committed 11 minutes ago

Data Binding - String Interpolation Exemple 2



Amar2050 committed 1 hour ago

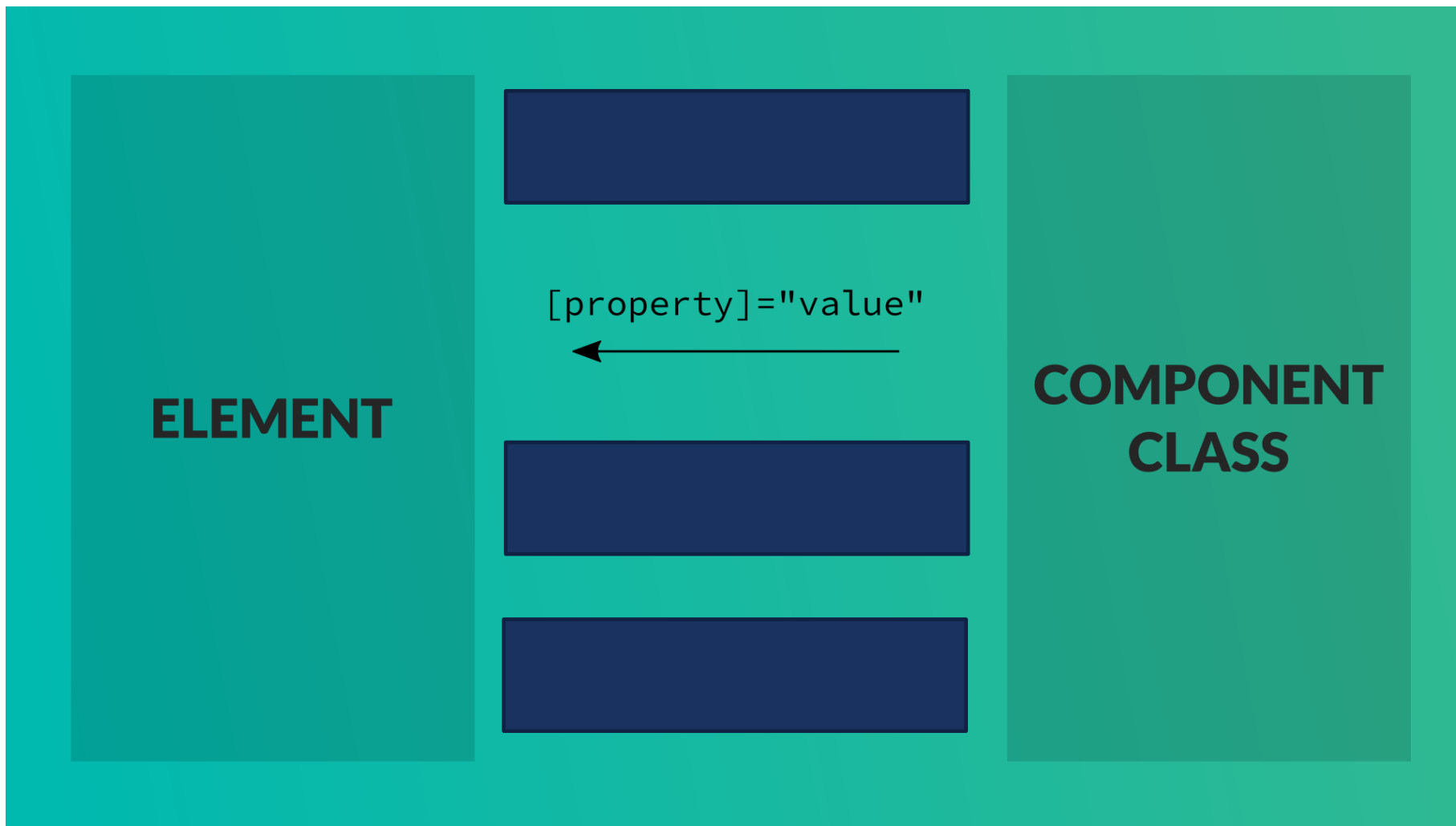
Data Binding - String Interpolation Exemple 1



Amar2050 committed 2 hours ago



Property Binding





Property Binding

▼ cars-list

cars-list.component.css

U

<> cars-list.component.html

U

TS cars-list.component.ts

U



Event Binding

ELEMENT



`(event)="handler()"`



**COMPONENT
CLASS**

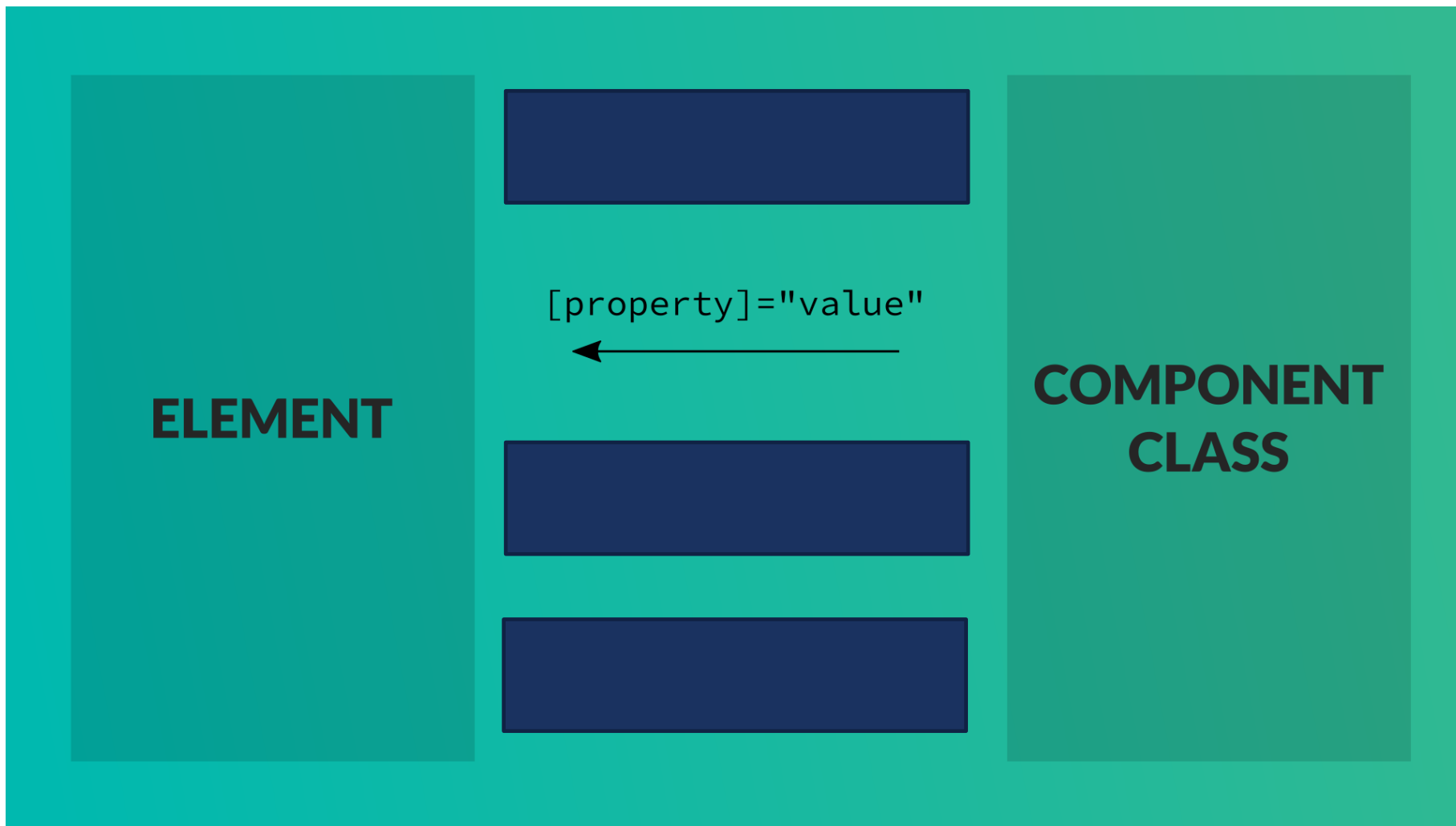


Event Binding **drivers-list**

```
▼ drivers-list
  # drivers-list.component.css      U
  <> drivers-list.component.html    U
  TS drivers-list.component.ts      U
```



Custom Property Binding + @Input()





car-item composant **enfant** de cars-list

```
Shell https://aka.ms/pscore6
```

```
d\HelloWorld> ng g c cars-list/car-item
```

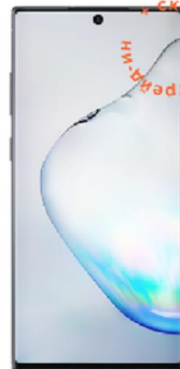


Каталог

Все Новинки Хиты продаж



Samsung Galaxy Note 10
4690 ₽ в мес.



Samsung Galaxy Note 10+
5490 ₽ в мес.

256 Гб

Стоимость смартфона	89990 ₽
Ежемесячный платеж	5490 ₽
Обмен на новый после 12 платежей	бесплатно
Ранний обмен на новый	4900 ₽
Samsung Smart Care+ на 1 год	✓



Samsung Galaxy S10e
2804 ₽ в мес.

SINGLE-ITEM-COMPONENT



ITEM-LIST-COMPONENT



Mini Easy TP de cars-list

Nos Vehicules

Mercedes AMG

Prix: 42000 €

Range Rover Evoque

Prix: 55000 €

Ferrari F50

Prix: 350000 €

Reproduire ça

Aide

vehiculeUn n'est plus de type string...
(vehiculeDeux, Trois non plus...)

Bootstrap : **p-5** | **row** | **card** | **m-2** | **border-warning** (Parent?Enfant? Les deux?)



Correction Mini TP de cars-list

TS cars-list.component.ts

```
vehiculeUn:any =  
{  
  title: "Mercedes AMG",  
  price: 42000  
}
```

CLASS

TS car-item.component.ts X

```
@Input()vehicule:any;
```

<> cars-list.component.html X

TEMPLATE

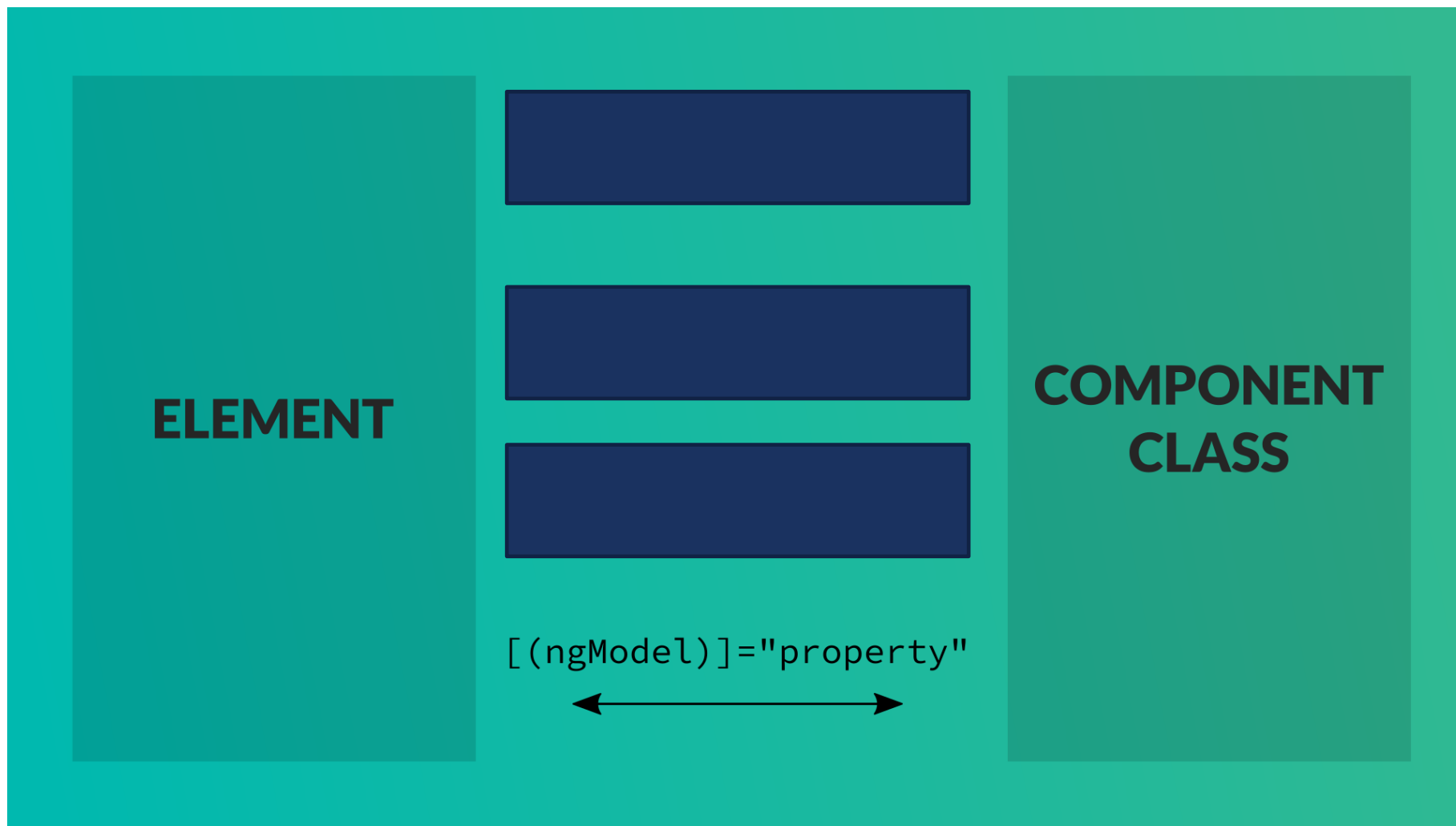
```
<h1 class="my-5">  
  Nos Vehicules  
</h1>  
  
<div class="container">  
  
  <div class="row">  
    <app-car-item [vehicule]="vehiculeUn"></app-car-item>  
    <app-car-item [vehicule]="vehiculeDeux"></app-car-item>  
    <app-car-item [vehicule]="vehiculeTrois"></app-car-item>  
  </div>  
  
</div>
```

<> car-item.component.html X

```
<div class="card border-warning p-5 m-2">  
  <h3>{{vehicule.title}}</h3>  
  <p>Prix: [{{vehicule.price}}] &euro;</p>  
</div>
```



Two way binding





Two way data binding [(ngModel)] **home-page**





```
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { FormsModule } from '@angular/forms';
6
7 import { AppComponent } from './app.component';
8 import { HeaderComponent } from './header/header.component';
9 import { CarsListComponent } from './cars-list/cars-list.component';
10 import { DriversListComponent } from './drivers-list/drivers-list.component';
11 import { HomePageComponent } from './home-page/home-page.component';
12 import { FourOhfourComponent } from './four-ohfour/four-ohfour.component';
13 import { CarItemComponent } from './cars-list/car-item/car-item.component';
14
```

You, a few seconds ago | 1 author (You)

```
5 @NgModule({
6   declarations: [
7     AppComponent,
8     HeaderComponent,
9     CarsListComponent,
10    DriversListComponent,
11    HomePageComponent,
12    FourOhfourComponent,
13    CarItemComponent,
14  ],
15   imports: [
16     BrowserModule,
17     AppRoutingModule,
18     FormsModule
19  ]
20 })
21 export class AppModule {}
```

You, a few seconds ago • Uncommitted changes



Two way data binding [(ngModel)] home-page

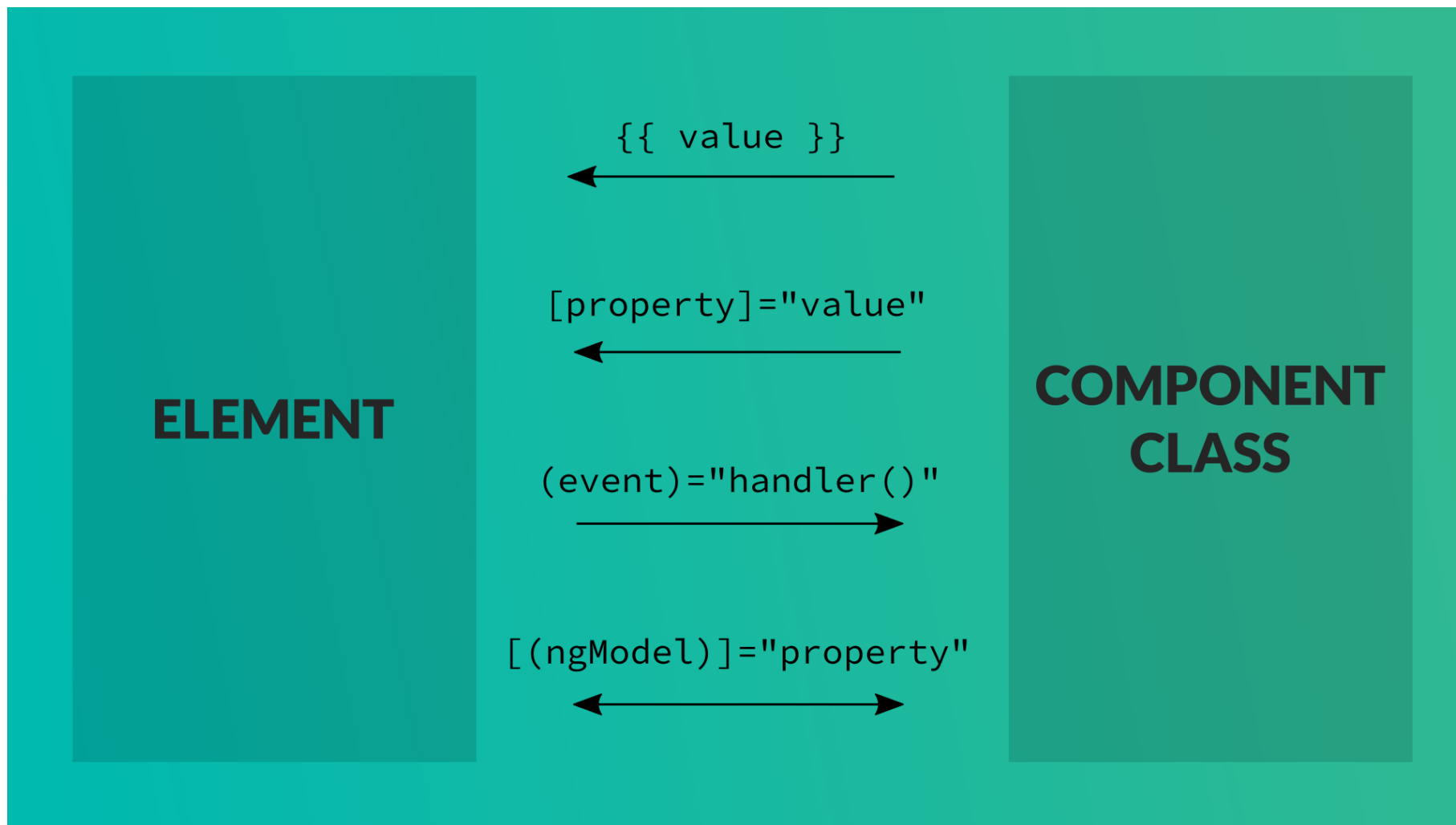
13

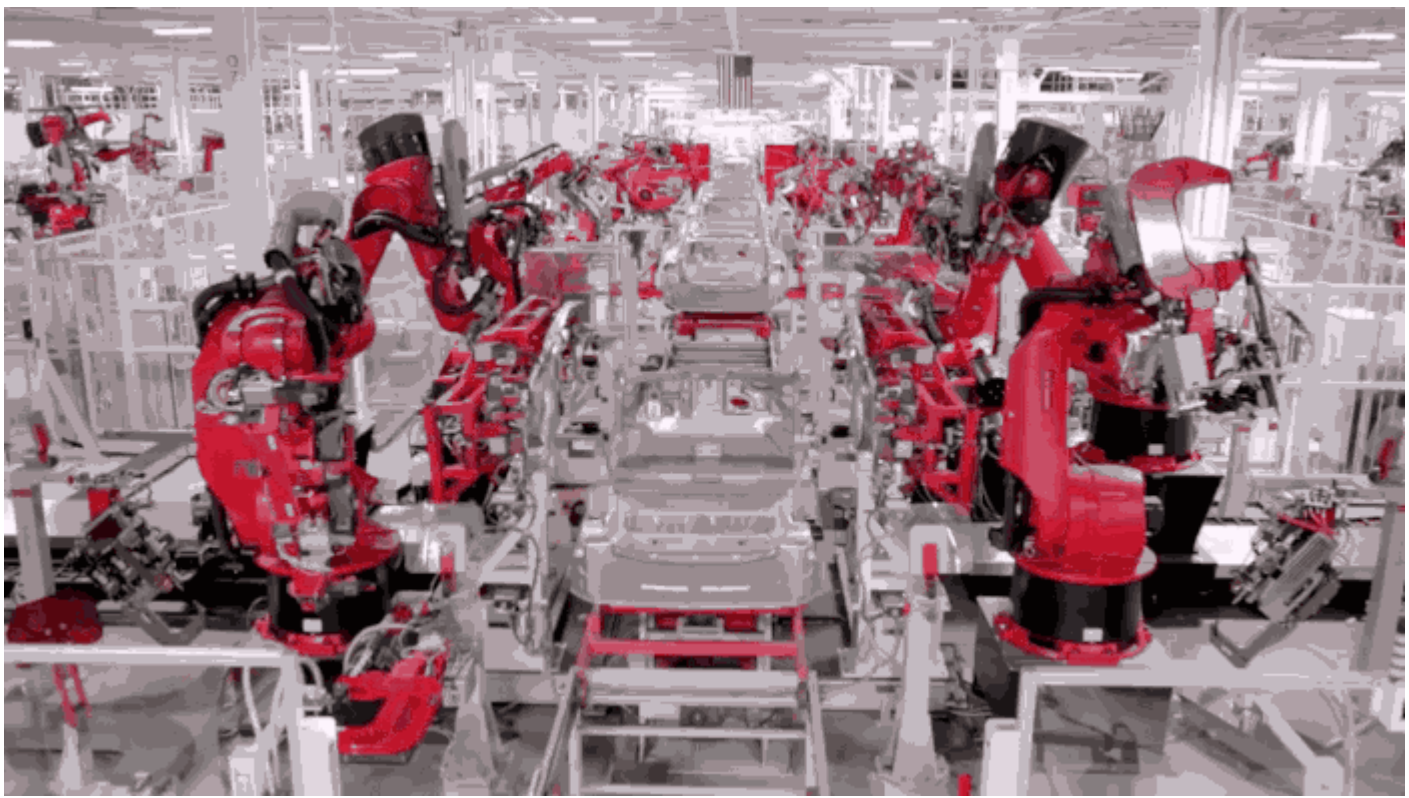
14 `<input type="text" [(ngModel)]="username">`

15 `<h2>Nom: {{username}}</h2>`



Data Binding





GO BACK TO DRIVE-X



Focus sur **car-list**

Sur la page Nos Véhicules rajouter :

données mise à jour : 'afficher la date d'aujourd'hui'

Comment ?



Simple string interpolation en class et template

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-cars-list',
5    templateUrl: './cars-list.component.html',
6    styleUrls: ['./cars-list.component.css']
7  })
8  export class CarsListComponent implements OnInit {
9
10
11    carUpdate:any;
12
13    constructor() { }
14
15    ngOnInit(): void {
16
17      this.carUpdate = new Date();
18
19    }
20
21  }
```




Simple string interpolation en class et template

```
1  <h1 class="my-5">
2  |    Nos Véhicules
3  </h1>
4
5  <div class="badge badge-dark my-5">Données mise à jour: {{ carUpdate }}</div>
6
7
```




car-list & car-item

Sur la page Nos Véhicules afficher 3 fiches de voitures :

Quelle component va contenir les voitures ?

Quelle component va afficher les voitures ?

Comment ?



Class car-list – Vrai données – Nom de variable en anglais

```
export class CarsListComponent implements OnInit {  
  
  carUpdate:any ;  
  
  carOne: any = {  
    name: "pagani huayra",  
    pays: "italie",  
    power: 765,  
    perf: 3.2  
  };  
  
  carTwo: any = {  
    name: "koenigsegg agera rs",  
    pays: "suède",  
    power: 1383,  
    perf: 2.6  
  };  
  
  carThree: any = {  
    name: "zenvo tsr s",  
    pays: "danemark",  
    power: 1200,  
    perf: 2.8  
  };  
}
```



Template car-list

HelloWorld > src > app > cars-list > <> cars-list.component.html > ...

```
1  <h1 class="my-5">
2    Nos Véhicules
3  </h1>
4
5  <div class="badge badge-dark my-5">Données mise à jour: {{ carUpdate }}</div>
6
7  <app-car-item [car]="carOne" ></app-car-item>
8  <app-car-item [car]="carTwo" ></app-car-item>
9  <app-car-item [car]="carThree" ></app-car-item>
10 |
```



Class car-item

```
1  import { Component, OnInit, Input } from '@angular/core';
2
3  @Component({
4      selector: 'app-car-item',
5      templateUrl: './car-item.component.html',
6      styleUrls: ['./car-item.component.css']
7  })
8  export class CarItemComponent implements OnInit {
9
10
11      @Input() car:any;
12
```



Template car-item

```
HelloWorld > src > app > cars-list > car-item > <> car-item.component.html
```

```
1  <p>{{car.name}}</p>
2  <p>Origine: {{car.pays}}</p>
3  <p>{{car.power }} CH</p>
4  <p>0 à 100 km/h: {{car.perf}} sec</p>
5  <hr>
```



Réservation d'un véhicule

Niveau **UI** ça ressemble à quoi ?

On as vu **quoi** qui s'approche de ce qu'on veut ?

Sur **quelle component** on va bosser selon vous ?



Mini TP Medium

Car-item & Event binding + Property binding + Interpolation


On me propose de réserver la voiture
(un message, un bouton...)

Quand je clique sur le bouton il se désactive
(un bouton... 🤪)

On me dit j'ai réserver cette voiture
(un autre message...)



Template Car-item

```
HelloWorld > src > app > cars-list > car-item > <> car-item.component.html >  hr
1   <p>{{car.name}}</p>
2   <p>Origine: {{car.pays}}</p>
3   <p>{{car.power }} CH</p>
4   <p>0 à 100 km/h: {{car.perf}} sec</p>
5   <p>{{bookingMessage}}</p>
6   <button class="btn btn-primary" (click)="onBooking()" [disabled]="isBooked">Réserver </button>
7   <hr>
```




Class Car-item

```
8 export class CarItemComponent implements OnInit {  
9  
10   @Input() car:any;  
11   isBooked:boolean = false;  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24 }  
25
```



Class Car-item

```
8  export class CarItemComponent implements OnInit {  
9  
10     @Input() car:any;  
11     isBooked:boolean = false;  
12     bookingMessage: string = "Réserver maintenant !";  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24 }  
25
```



Class Car-item

```
8  export class CarItemComponent implements OnInit {  
9  
10     @Input() car:any;  
11     isBooked:boolean = false;  
12     bookingMessage: string = "Réserver maintenant !";  
13  
14     constructor() { }  
15  
16     ngOnInit() {  
17     }  
18  
19     onBooking(){  
20         this.isBooked= true;  
21         this.bookingMessage= "Cette voiture est réservée";  
22     }  
23  
24 }  
25
```



Car-item & Event Binding + Property binding

pagani huayra

Origine: italie

765 CH

0 à 100 km/h: 3.2 sec

Cette voiture est réservée

Réserver

koenigsegg agera rs

Origine: suède

1383 CH

0 à 100 km/h: 2.6 sec

Réserver maintenant !

Réserver



PIPES

FILTRES DE TRANSFORMATION DE DONNÉES



PipeTransform

- INTRODUCTION
- GETTING STARTED >
- SETUP
- FUNDAMENTALS >
- TECHNIQUES >
- DEV WORKFLOW >
- CONFIGURATION >
- RELEASE INFORMATION >
- QUICK REFERENCE >
- CLI COMMANDS >

API

stable (v8.2.14)

API > @angular/core

PipeTransform INTERFACE

An interface that is implemented by pipes in order to perform a transform on a value with a set of parameters as the second argument in list form.


```
interface PipeTransform {  
  transform(value: any, ...args: any[]): any  
}
```

Class implementations

- AsyncPipe
- DatePipe
- I18nPluralPipe
- I18nSelectPipe
- JsonPipe
- LowerCasePipe
- CurrencyPipe
- DecimalPipe
- PercentPipe
- SlicePipe
- UpperCasePipe
- TitleCasePipe



Les Pipes titlecase & uppercase

```
HelloWorld > src > app > cars-list > car-item > <> car-item.component.html >  hr  
1    <p>{{car.name | titlecase}}</p>  
2    <p>Origine: {{car.pays | uppercase}}</p>  
3    <p>{{car.power }} CH</p>
```



Pipe et date

Drive-X Voitures Pilotes

Nos Véhicules

Données mise à jour: 27 DÉCEMBRE 2019 16:37



Pipe et date française = app.module.ts

```
4 import { registerLocaleData } from '@angular/common';  
5 import localeFr from '@angular/common/locales/fr';  
6 registerLocaleData(localeFr, 'fr');  
7  
8 import { AppRoutingModuleModule } from '../app-routing.module';  
9 import { FormsModule } from '@angular/forms';  
9
```

```
providers: [{ provide: LOCALE_ID, useValue: "fr-FR" }],  
bootstrap: [AppComponent]  
})  
export class AppModule { }
```



Pipe et date

Dans le template

```
{{ carUpdate | date:"dd MMMM yyyy H:mm" | uppercase }}
```

```
<div class="badge badge-dark my-5">Données mise à jour:  
| | {{ carUpdate | date:"EEEE dd MMMM yyyy à H:mm" }}  
</div>
```

Le but data binding

Lié des données utilisateurs a la classe du module de la directive de l'événement d'une propriété

• A

D'établir une liaison avec une base de donnée

• B

Avoir une structure qui s'adapte aux données ou réactions utilisateurs

• C

Permet de rendre notre application réactive et dynamique

• D

Le but data binding

Lié des données utilisateurs a la classe du module de la directive de l'évent d'une propriété

• A

D'établir une liaison avec une base de donnée

• B

Avoir une structure qui s'adapte aux données ou réactions utilisateurs

• C

Permet de rendre notre application réactive et dynamique

• D

En interpolation

La liaison se fait du Template à la Classe

• A

La liaison se fait du Template au Template

• B

La liaison se fait de la Classe au Template

• C

La liaison se fait dans les deux directions

• D

En interpolation

La liaison se fait du Template à la Classe

• A

La liaison se fait du Template au Template

• B

La liaison se fait de la Classe au Template

• C

La liaison se fait dans les deux directions

• D

QUIZ 3/8

En two way data binding

La liaison se fait du Template à la Classe

• A

La liaison se fait du Template au Template

• B

La liaison se fait de la Classe au Template

• C

La liaison se fait dans les deux directions

• D

En two way data binding

La liaison se fait du Template à la Classe

• A

La liaison se fait du Template au Template

• B

La liaison se fait de la Classe au Template

• C

La liaison se fait dans les deux directions

• D

QUIZ 4/8

En interpolation on peut...

Lié une propriété html a une variable ou une méthode

• A

Afficher une variable ou une méthode

• B

Afficher une variable ou autres et lui appliquer des traitements

• C

Réagir a un évènement du DOM grâce a une variable

• D

QUIZ 4/8

En interpolation on peut...

Lié une propriété html a une variable ou une méthode

• A

Afficher une variable, le retour d'une méthode, un objet...

• B

Afficher une variable ou autres et lui appliquer des traitements

• C

Réagir a un évènement du DOM grâce a une variable

• D

Property Binding

Va de la Classe au Template et lie un évènement a une méthode

• A

C'est lier les propriétés d'une Classe aux évènements du Template

• B

C'est la capacité de menotter des gens dans leur propriété

• C

Va de la Classe au Template et lie une propriété a une méthode ou une variable

• D

Property Binding

Va de la Classe au Template et lie un évènement a une méthode

• A

C'est lier les propriétés d'une Classe aux évènements du Template

• B

C'est la capacité de menotter des gens dans leur propriété

• C

Va de la Classe au Template et lie une propriété a une méthode ou une variable

• D

Event Binding

Va du Template a la Classe et lie un évènement a une méthode

• A

C'est moins chiant a faire que en Js natif...

• B

Cela permet de récupérer les interactions et d'y réagir

• C

Va de la Classe au Template et lie un évènement a une méthode

• D

Event Binding

Va du Template a la Classe et lie un évènement a une méthode

• A

C'est moins chiant a faire que en Js natif...

• B

Cela permet de récupérer les interactions et d'y réagir

• C

Va de la Classe au Template et lie un évènement a une méthode

• D

Les Pipes

C'est des mot-clés de Angular qui aurait pu donner lieu a une blague... mais non...

• A

Pipe ou tuyau , ce sont des filtres de transformations de données

• B

C'est bien et on peut même les faire soit même !

• C

Ils permettent de modifier l'affichage d'une donnée, certains modifient le type de la donnée

• D

Les Pipes

C'est des mot-clés de Angular qui aurait pu donner lieu a une blague... mais non...

• A

Pipe ou tuyau , ce sont des filtres de transformations de données

• B

C'est bien et on peut même les faire soit même !

• C

Ils permettent de modifier l'affichage d'une donnée, certains modifient le type de la donnée

• D

QUIZ 8/8

@Input()

Permet de récupérer les informations des utilisateurs, utiliser dans les formulaires notamment

• A

C'est un décorateur, comme tout décorateur il permet d'apporter des informations supplémentaires a une classe

• B

Lorsqu'un ordinateur ou un appareil reçoit une commande ou un signal de sources externes

• C

Permet de récupérer des informations depuis un autre component pour les utiliser

• D

QUIZ 8/8

@Input()

Permet de récupérer les informations des utilisateurs, utiliser dans les formulaires notamment

• A

C'est un décorateur, comme tout décorateur il permet d'apporter des informations supplémentaires à une classe

• B

Lorsqu'un ordinateur ou un appareil reçoit une commande ou un signal de sources externes

• C

Permet de récupérer des informations depuis un autre composant pour les utiliser

• D