



TP DRIVERS

CORRECTION



Créer le component driver-item

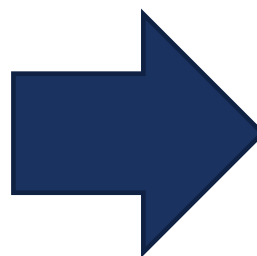
```
▼ drivers-list  
  > driver-item  
  # drivers-list.component.css  
  <> drivers-list.component.html  
  TS drivers-list.component.ts
```

```
ng g c drivers-list/driver-item
```



Dans – **driver-list** les données du fichier texte déjà

```
drivers: any = [  
  {  
    fullName: "ken block",  
    pays: "usa",  
    coverImage: "./assets/img/drivers/kenblock.jpg",  
    category: "gymkhana",  
    likeIts: -1  
  },  
  {  
    fullName: "danica patrick",  
    pays: "usa",  
    coverImage: "./assets/img/drivers/danicapatrick.jpg",  
    category: "nascar",  
    likeIts: 0  
  },  
  {  
    fullName: "sebastien loeb",  
    pays: "france",  
    coverImage: "./assets/img/drivers/Loeb.jpg",  
    category: "rallye",  
    likeIts: 1  
  },  
]
```



```
export class DriversListComponent implements OnInit {  
  
  drivers: any = [  
    {  
      fullName: "ken block",  
      pays: "usa",  
      coverImage: "./assets/img/drivers/kenblock.jpg",  
      category: "gymkhana",  
      likeIts: -2  
    },  
    {  
      fullName: "danica patrick",  
      pays: "usa",  
      coverImage: "./assets/img/drivers/danicapatrick.jpg",  
      category: "nascar",  
      likeIts: 0  
    },  
    {  
      fullName: "sebastien loeb",  
      pays: "france",  
      coverImage: "./assets/img/drivers/Loeb.jpg",  
      category: "rallye",  
      likeIts: 1  
    },  
  ],  
}
```



Dans – **assets** les photos



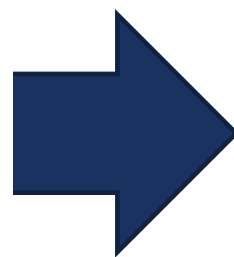
cars



drivers



404.gif



✓ assets

✓ img

> cars

> drivers

🖼️ 404.gif

📁 .gitkeep



Même card – **donc** même code – même principe

Nos Véhicules

Données mise à jour: 18 OCTOBRE 2020 11:09

Pagani Huayra



Origine: ITALIE

Puissance: **765 CH**

0 à 100 km/h: 3.2 sec

Réserver maintenant !

Réserver

Koenigsegg Agera Rs



Origine: SUÈDE

Puissance: **1383 CH**

0 à 100 km/h: 2.6 sec

Réserver maintenant !

Réserver

Nos Pilotes

Ken Block



Origine: USA

Discipline: **Gymkhana**



Danica Patrick



Origine: USA

Discipline: **Nascar**





Forcement – **drivers-list** comme **cars-list**

```
<div class="card-deck justify-content-center">  
  <app-car-item *ngFor="let oneCar of cars" [car]="oneCar" ></app-car-item>  
</div>
```

Pas encore les données donc erreurs...

```
<div class="card-deck justify-content-center">  
  <app-driver-item *ngFor="let oneDriver of drivers" [driver]="oneDriver"></app-driver-item>  
</div>
```

Nos Pilotes

driver-item works!driver-item works!driver-item works!driver-item works!driver-item works!driver-item

Mais la boucle
fonctionne déjà



Import – **driver-item** Custom Property Binding et @Input

```
import { Component, OnInit, Input } from '@angular/core';
```

```
export class DriverItemComponent implements OnInit {  
  
  @Input() driver:any;
```

```
p > drivers-list > driver-item > <> driver-item.component.html >  
<p>{{driver.fullName}}</p>  
|
```

On transfère et on teste en simple

```
driver-item works!driver-item works!driver-item works!driver-item
```



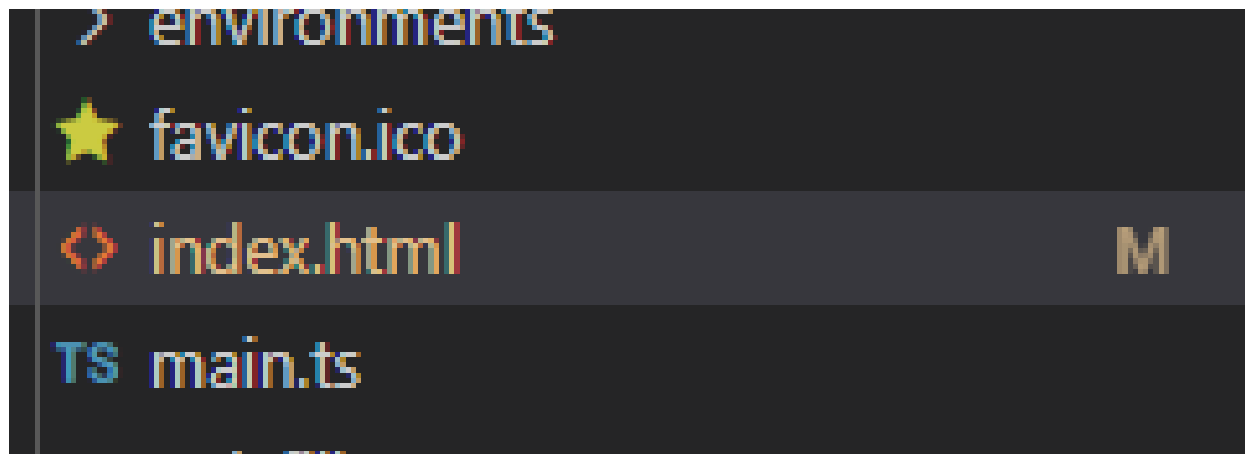
```
ken blockdanica patricksebastien loebmolly taylorcollin mcraeshirley
```

On récupère bien nos données !



Design – driver-item - icones

Rappel – index représente toute l'app !



Font Awesome



Icones dispo partout
Bootstrap en cdn c'est ici aussi

```
src > <> index.html > ...
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>DriveX</title>
6      <base href="/">
7      <meta name="viewport" content="width=device-width, in
8
9      <link rel="stylesheet" href="https://use.fontawesome.
10      integrity="sha384-50oBUHEmvpQ+1lW4y57PTFmhCaXp0ML5d
11      <link rel="icon" type="image/x-icon" href="favicon.ic
12
13  </head>
14  <body>
15      <app-root></app-root>
16  </body>
17  </html>
18
```



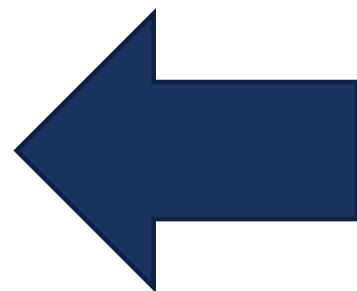

Design – **driver-item** – card et boutons sans directive mais avec data binding obligé

Ken Block



Origine: USA

Discipline: Gymkhana



Le plus gros
du design
est fait

```
<div class="card text-center shadow">

  <div class="card-header bg-dark text-white ">
    <h2 class="card-title">{{driver.fullName | titlecase }}</h2>
  </div>

  <div class="card-body">

    <img [src]="driver.coverImage" width="300px" height="150px"
      class="img-fluid" alt="photo {{driver.fullName}}">
    <hr>

    <p>Origine: {{driver.pays | uppercase }}</p>

    <p>Discipline:{{driver.category | titlecase }}</p>

    <button class="btn btn-success mx-2 border" >
      <i class="fas fa-thumbs-up"></i>
    </button>

    <button class="btn btn-danger border">
      <i class="fas fa-thumbs-down"></i>
    </button>

  </div>
</div>
<hr>
```



Différences – objectifs

Ken Block



Origine: USA

Discipline: Gymkhana



Nos Pilotes

Ken Block



Origine: USA

Discipline: **Gymkhana**



Danica Patrick



Origine: USA

Discipline: **Nascar**



Cadre qui change
de couleur selon
la note

Badge selon la
discipline

Function + et –



ngClass – **easy**

Ken Block



Origine: USA

Discipline: **Gymkhana**



Cadre qui change de couleur selon la note

```
<div class="card text-center shadow"  
  [ngClass]="{'border-success': driver.likeIts > 0,  
             'border-danger': driver.likeIts < 0 }">
```

On pouvait tester en direct en changeant la note



ngClass & ngIf – easy toujours

Ken Block



Origine: USA

Discipline: **Gymkhana**



```
<p>Discipline:
  <span *ngIf="driver.category" [ngClass]="{'badge':true,
    'badge-success': driver.category == 'gymkhana',
    'badge-warning': driver.category == 'formule 1',
    'badge-info': driver.category == 'rallye',
    'badge-primary': driver.category == 'nascar',
    'badge-danger': driver.category == 'drag'}"
    >{{driver.category | titlecase }}</span>
  <span *ngIf="driver.category == null">NC</span>
</p>
```

Badge réactif et nc pour disciple et pays

```
<p>Origine: {{driver.pays | uppercase }}
  <span *ngIf="driver.pays == null">NC</span>
</p>
```



Event Binding— **easy** **encore**

Ken Block



Origine: USA

Discipline: **Gymkhana**



```
<button class="btn btn-success mx-2 border" (click)="onLike()">
  <i class="fas fa-thumbs-up"></i>
</button>

<button class="btn btn-danger border" (click)="onUnlike()">
  <i class="fas fa-thumbs-down"></i>
</button>
```

Pour faire simple deux méthodes

Pour la partie Template c'était finis
Partie logique maintenant



Event Binding— **easy un peu**

Ken Block



Origine: USA

Discipline: **Gymkhana**



But modifier cette note

```
export class DriverItemComponent {  
  @Input() driver:any;
```

```
drivers: any = [  
  {  
    fullName: "ken block",  
    pays: "usa",  
    coverImage: "./assets/...",  
    category: "gymkhana",  
    likeIts: -2
```

Mais comme la réservation ça concerne
que un pilote donc doit être dans driver-
item et affecté driver pas drivers



Event Binding— **easy moyen**

Ken Block



Origine: USA

Discipline: **Gymkhana**



Donc tout simplement

On incrémente
ou l'inverse

```
[ngClass]="{ 'border-success':  
            'border-danger': c
```

```
: driver.likeIts > 0,  
driver.likeIts < 0 }"]>
```

```
export class DriverItemComponent  
  
  @Input() driver:any;  
  
  constructor() { }  
  
  ngOnInit(): void {  
  }  
  
  onLike() {  
    this.driver.likeIts++;  
  }  
  onUnlike() {  
    this.driver.likeIts--;  
  }  
}
```




Bravo



Bonnes bases en Angular