

Gestion du projet

```
ng new monProjet [--routing false] [--style css]
```

```
ng version (v)
```

```
ng serve (s) [--host 0.0.0.0] [-port 4205]
```

```
ng generate (g) <type> [options]
```

Formulaires

```
import { FormsModule } from '@angular/forms';
```

```
<input [(ngModel)]="userName">
```

Fournit l'accès aux données dans les 2 sens, l'analyse et la

[Spécifier la validation

le routage]

[Spécifier le style]

```
@Component({...})
```

```
class MyComponent() {}
```

```
@Directive({...})
```

```
class MyDirective() {}
```

```
@Pipe({...})
```

```
class MyPipe() {}
```

Declare qu'une classe est un composant (resp. une directive ou un

pipe) et fournit de la métadonnée à propos du composant (resp.

directive ou pipe)

[Port

perso]

(local

t:4205)

Générer

```
@Input() myProperty;
```

Déclare une propriété d'entrée qui peut être mise à jour

```
@Output() myEvent = new EventEmitter();
```

Déclare une propriété de sortie qui lance des événements qu'on peut

suivre avec une reliure d'evt

```
@HostBinding('class.valid') isValid;
```

Relie une propriété d'élément à une propriété de directive/composant

```
@HostListener('click', ['$event']) onClick(e)
```

```
{...}
```

Suit un élément événement avec une méthode de directive/co-

```
@ContentChild(myPredicate) myChildComponent;
```

```
@ContentChild(referencem yPredicate) myChildComponents;
```

Génération de décorateur

Type

component (c) nomC Composant

directive (d) nom Direct. personnalisée

pipe (p) nom Pipe

route (r) comp Route

service (s) nom Service

Options

--skip -tests Pas de fic. test

--flat Pas de répertoire

-s Style en ligne

-t Template en ligne



By Maxime Frémeaux

(maxant15)

cheatography.com/maxant15/

maxime-fremeaux.fr

Not published yet.

Last updated 10th March, 2021.

Page 1 of 2.

Sponsored by [ApolloPad.com](https://apollopod.com)

Everyone has a novel in them. Finish

Yours!

<https://apollopod.com>

Champs de décorateur de classe (cont)

Fournit le(s) [premier] résultat(s) de la requête du contenu du composant à une propriété de la classe

NgModules

```
import { NgModule } from '@angular/core';
@NgModule({
  declarations: [ExempleComponent],
  imports: [ExempleModule],
  exports: [AutreComposant],
  providers: [MonService, { provide:
e:...}],
  entryComponents: [],
  bootstrap: [MyAppComponent]
})
class MyModule {}
```

entryComponent : déclarer des composants non référencés dans aucun template atteignable (par exemple créé dynamiquement par le code)

Syntaxe de template

```
<input [value]="firstName">
```

Relie la propriété value au résultat de l'expression firstName

```
<button (click)="readRainbow($event)">
```

Appelle la méthode readRainbow quand un click est détecté sur cet élément (ou ses enfants) et passe l'objet event

```
<p> Hello {{myVariable}}</p>
```

Relie le contenu du texte à une chaîne de caractères interpolée

```
<my-cmp [(title)]="name">
```

Configure une reliure à deux sens

```
<video #movie player ...>
```

```
<button (click)="moviePlayer.play()">
```

```
</video>
```

Crée une variable locale qui fournit l'accès à l'instance d'élément video (pour les data-binding et event-binding)

Syntaxe de template (cont)

```
<p *myUnless="moonExpressions">...</p>
```

Equivalent à <ng-template [myUnless]="moonExpressions">...</ng-template> (template intégré)

```
<p>Card No.: {{cardNumber | myCardNumberFormatter}}</p>
```

Transforme la valeur de l'expression CardNumber via le pipe myCardNumberFormatter

```
<p> Employer: {{employer?.companyName}}</p>
```

Opérateur de navigation sécurisée. Si employer est undefined, le résultat est ignoré

Directives built-in

```
<section *ngIf="showSection">
```

Retire ou recrée une portion du DOM basée sur l'expression de showSection

```
<li *ngFor="let item of list">
```

Transforme l'élément courant et son contenu dans un template, et instancie une vue pour chaque élément de la liste

```
<div [ngStyle]="{'property': 'value'}">
```

```
<div [ngStyle]="dynamicStyles()">
```

Utiliser du CSS directement ou appeler une méthode de style

Détection changement composant/directive & cycles

```
constructor(myService: MyService, ...) { ... }
```

Appelé avant n'importe quel autre cycle (injection de dépendances).

```
ngOnChanges(changesRecord) { ... }
```

Appelé après changement à la propriété d'entrée et avant de procéder au contenu ou aux vues des fils.

```
ngOnInit() { ... }
```

Appelée après le constructeur, initialise les propriétés d'entrée et le premier appel à ngOnChanges.

```
ngOnDestroy() { ... }
```

Appelé une fois, avant que l'instance ne soit détruite.



By Maxime Frémeaux

(maxant15)

cheatography.com/maxant15/

maxime-fremeaux.fr

Not published yet.

Last updated 10th March, 2021.

Page 2 of 2.

Sponsored by [ApolloPad.com](https://apollopad.com)

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>