Cheatography

Bases Angular Cheat Sheet

by Maxime Frémeaux (maxant15) via cheatography.com/132904/cs/26971/

Gestion du projet

`ng new monProjet [--routing false] [--style css]`	Créer un nouv. projet [Spécifier le routage] [Spéc le style]
`ng version (v)`	Donne la version installée
`ng serve (s) [--host 0.0.0.0] [-port 4205]`	Lancer le serveur [Adresse perso] [Port perso] *(localhost:4200)*
`ng generate (g) <ty pe> [options]`	Générer un décorateur (voir section en-des-sous)

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

Formulaires

```
import { FormsM odule } from '@angu lar /fo rms';
<input [(ngMo del )]= " use rNa me">
```

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

Décorateur de classes

```
@Compo nen t({...})
class MyComp onent() {}
@Direc tiv e({...})
class MyDire ctive() {}
@Pipe( {...})
class MyPipe() {}
```

Déclare 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)

Champs de décorateur de classe

```
import { Input, ... } from '@angu lar /core';
@Input() myProp erty;
```

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

```
@Output() myEvent = new EventE mit ter();
```

Déclare une propriété de sortie qui lance des évènements qu'on peut suivre avec une reliure d'evt

```
@HostB ind ing ('c las s.v alid') isValid;
```

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

```
@HostL ist ene r(' click', ['$eve nt']) onClick(e) {...}
```

Suit un élément évènement avec une méthode de directive/co-mposant

```
@Conte ntC hil d(m yPr edi cate) myChil dCo mpo nen t;
@Conte ntC hil dre n(m yPr edi cate) myChil dCo mpo nents;
```

Cheatography

Bases Angular Cheat Sheet

by Maxime Frémeaux (maxant15) via cheatography.com/132904/cs/26971/

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';
@NgMod ule({
        dec lar ations: [Exemp leC omp onent],
        imp orts: [Exemp leM odule],
        exp orts: [Autre Com pon ent],
        pro viders: [MonSe rvice, { provid -
e:...}],
        ent ryC omp one nts:[],
        boo tstrap: [MyApp Com ponent]
})
class MyModule {}
```

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

Syntaxe de template

`<input [value ]="f irs tNa me">`

Relie la propriété `value` au résultat de l'expression `firstname`

`<button (click )="r ead Rai nbo w($ eve nt) ">`

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

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

Relie le contenu du texte à une chaîne de caractères interpolée

`<my-cmp [(titl e)] ="na me">`

Configure une reliure à deux sens

`<video #movie player ...>`

`<button (click )="m ovi epl aye r.p lay ()">`\`

`</v ide o>`

Créé une variable locale qui fournit l'accès à l'instance d'élément `vi deo` (pour les data-binding et event-binding)

Syntaxe de template (cont)

`<p *myUnl ess ="mo nEx pre ssi on">...< /p>`

Equivalent à `<ng -te mplate [myUnl ess ]="m onE xpr ess ...</ p>< /ng -te mpl ate>` (template intégré)

`<p>Card No.: {{cardNumber | myCard Num ber For mat te`

Transforme la valeur de l'expression `CardNumber` via le pipe `myCard matter`

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

Opérateur de navigation sécurisée. Si `employer` est `undefined`, le re ession est ignoré

Directives built-in

`<se ction *ngIf= " sho wSe cti on">`

Retire ou recrée une portion du DOM basée sur l'expression de `sho wSe ction`

`<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 [ngSty le] ="{' pro perty': 'value '}">`

`<div [ngSty le] ="dy nam icS tyl es( )">`

Utiliser du CSS directement ou appeler une méthode de style

Détection changement composant/directive & cycles

`constr uct or( myS ervice: MyService, ...) { ... }`

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

`ngOnCh ang es( cha nge Record) { ... }`

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 `ngOnCh anges`.

`ngOnDe stroy() { ... }`

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.