

Glossaire	Définition
AngularJS	Framework Javascript
Angular	Nom courant de Angular 5.x, géré par Google
Angular-cli	Command line Interface, outil pour exécuter des scripts pour la création d'une application Angular. Fonctionne avec NODE.JS et NPM (installation via : <code>npm install -g @angular/cli</code> ) les commandes commenceront alors par <code>ng</code> , exemples : <code>ng new nom-projet --style=css</code> ou <code>ng serve --open</code> => qui va lancer le serveur et ouvrir un navigateur
Autre framework JS	React, Ember, Vue
TypeScript	Langage qui a un typage strict, l'ajout de fonctions lambda, la création de Classes et Interfaces. C'est un « superset » de Javascript qui est transcompilé pour pouvoir être compris par les navigateurs.
Single page app	Application rapide puisqu'un seul chargement de page au début, et modification des contenus (components) grâce au « routing », fichier 'app.module.ts' qui liste les routes disponibles
Components angular	Comme avec reactNative, tout est component. Chaque « boîte », chaque contenu sera isolé dans un component qui aura comme nom par exemple MenuComponent, ContentComponent, Sidebarcomponent, tous se trouvant dans le dossier app
Service Angular	Codes qui seront utilisés dans plusieurs parties de l'application. Cela évite la répétition puisque tout est centralisé dans le service. En fonction de l'utilisation souhaitée du service, on l'injecte soit dans AppModule, pour être accessible partout (services, components), soit dans AppComponent et là, tous les components y auront accès mais pas les autres services, soit dans un seul component, et là c'est seulement lui et ses « enfants » qui ont y auront accès.
Mobile JS	
Model Angular	Modèle des Objets, cela liste les attributs et les types d'attributs de l'objet en question.
Databinding	Il s'agit des différentes manières de synchroniser le template et le component, dans un sens, dans l'autre ou dans les 2.
Interpollation	Échange des données depuis le code (TypeScript) vers le template (html), grâce à la syntaxe <code>{{ maVar }}</code> (dans l'html) qui prend la valeur définie dans la classe du component (.ts)

Glossaire	Définition
propertyBinding	« Liaison par propriété » : Communication dynamique entre le code TypeScript et le template. Exemple d'un statut « actif » qui doit être true pour faire afficher un élément du template. Ça se passe donc entre l'html où la condition sera vérifiée via le TypeScript. La syntaxe est avec des crochets dans la balise. Ex : <pre>&lt;button class="btn btn-success" [disabled]="!isAuth"&gt;Un bouton magique&lt;/button&gt;</pre>
EventBinding	« Liaison par événement » : Selon un événement sur le DOM, il y aura une réaction dans le code et cela activera certaines méthodes. La syntaxe est avec des parenthèses dans la balise. Ex : <pre>&lt;button class="btn btn-success" (click)="onMixer()"&gt;Mixer&lt;/button&gt;</pre>
Observable	Un objet qui émet des informations dans le temps (exemple : un champs texte où l'utilisateur va saisir des données), soit des envois de données ou bien de types erreurs ou encore des messages.
Observer	Code qui sera exécuté à chaque fois que l'Observable émettra une information.
Subjects	De la même manière qu'une Observable, le Subject est associé à un Observer. Cependant, on peut faire des subscription à un Subject, mais un Subject peut aussi souscrire à d'autres Observables.