**Annexe 1 – React Native : On se lance !**

**Qu'est‑ce que React Native ?**  
React Native est un framework open‑source développé par Facebook qui permet de créer des applications mobiles natives pour iOS et Android en utilisant JavaScript et la bibliothèque React.

**Démarche pour créer un nouveau projet**

**MANIÈRE LONGUE (CLI React Native)**

1. Installer le CLI globalement :

npm install -g react-native-cli

1. Créer un nouveau projet :

react-native init NomDuProjet

1. Se déplacer dans le dossier :

cd NomDuProjet

1. Démarrer Metro (bundler) :

npx react-native start

1. Lancer l'application :
   * Sur Android :

npx react-native run-android

* + Sur iOS (macOS uniquement) :

npx react-native run-ios

**MANIÈRE COURTE (Expo)**

1. Ouvrir le projet **ModeleDepart** fourni sur LEA ou V:/elabonte/ModeleDepart.
2. Dans VS Code, ouvrir un terminal à la racine du projet.
3. Installer Expo :

npm install expo

1. Démarrer Expo :

npx expo start

1. Appuyer sur **a** pour lancer un émulateur Android (un AVD doit être ouvert dans Android Studio).

**Outils clés**

* **Node.js** : exécution JavaScript côté serveur/termina.
* **npm (Node Package Manager)** : gestion des paquets.
* **npx** : exécution de paquets sans installation globale.
* **Metro** : bundler JavaScript.
* **Expo** : plateforme pour développer et tester rapidement des apps RN.

**Structure du projet**

ModeleDepart/

├── node\_modules/ # dépendances (dans package.json)

├── app/

│ └── tabs/

│ └── index.tsx # composant d'exemple en TSX

├── components/ # composants réutilisables

│ └── MonComposant.js

├── App.js # point d'entrée

├── App.json # config de l'application

└── package.json # dépendances et scripts

**Créer un composant sans état**

1. Ouvrir App.js ou index.js.
2. Supprimer le code par défaut.
3. Définir un composant stateless.

**Exercice : Composant Chanson**

**Composant fonctionnel**

Un composant fonctionnel est une fonction JavaScript (ou TypeScript) qui reçoit des propriétés (props) en argument et renvoie du JSX pour décrire l’interface utilisateur. Depuis React 16.8, les composants fonctionnels peuvent gérer l’état et les effets de cycle de vie grâce aux Hooks (useState, useEffect, etc.), offrant une syntaxe plus concise et une meilleure lisibilité par rapport aux composants de classe.

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Composant de classe**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Qu'est‑ce que JSX ?**

JSX (JavaScript XML) est une extension syntaxique qui permet d'écrire des balises ressemblant à du HTML dans le code JavaScript, simplifiant la définition de l'interface utilisateur.

**Stylisation avec StyleSheet**

A screen shot of a computer

AI-generated content may be incorrect.

**Exercice : Playlist**

**A computer screen with many colorful text

AI-generated content may be incorrect.**

**Flexbox: deux chansons par ligne**

const styles = StyleSheet.create({

playlist: {

flexDirection: 'row',

flexWrap: 'wrap',

justifyContent: 'space-between',

},

chanson: {

width: '48%',

marginVertical: 8,

},

});

**Bonnes pratiques**

* **Versionner :** git add ., git commit -m "message", git push.
* **Re-cloner :**
* git clone <url>
* cd <repo>
* npm install

npx expo start

* Vérifier que node\_modules est ignoré (.gitignore).

**Lien utile**

https://reactnative.dev/docs/environment-setup