# Annexe 1 – React Native - On se lance !

## Qu'est-ce que React Native ?

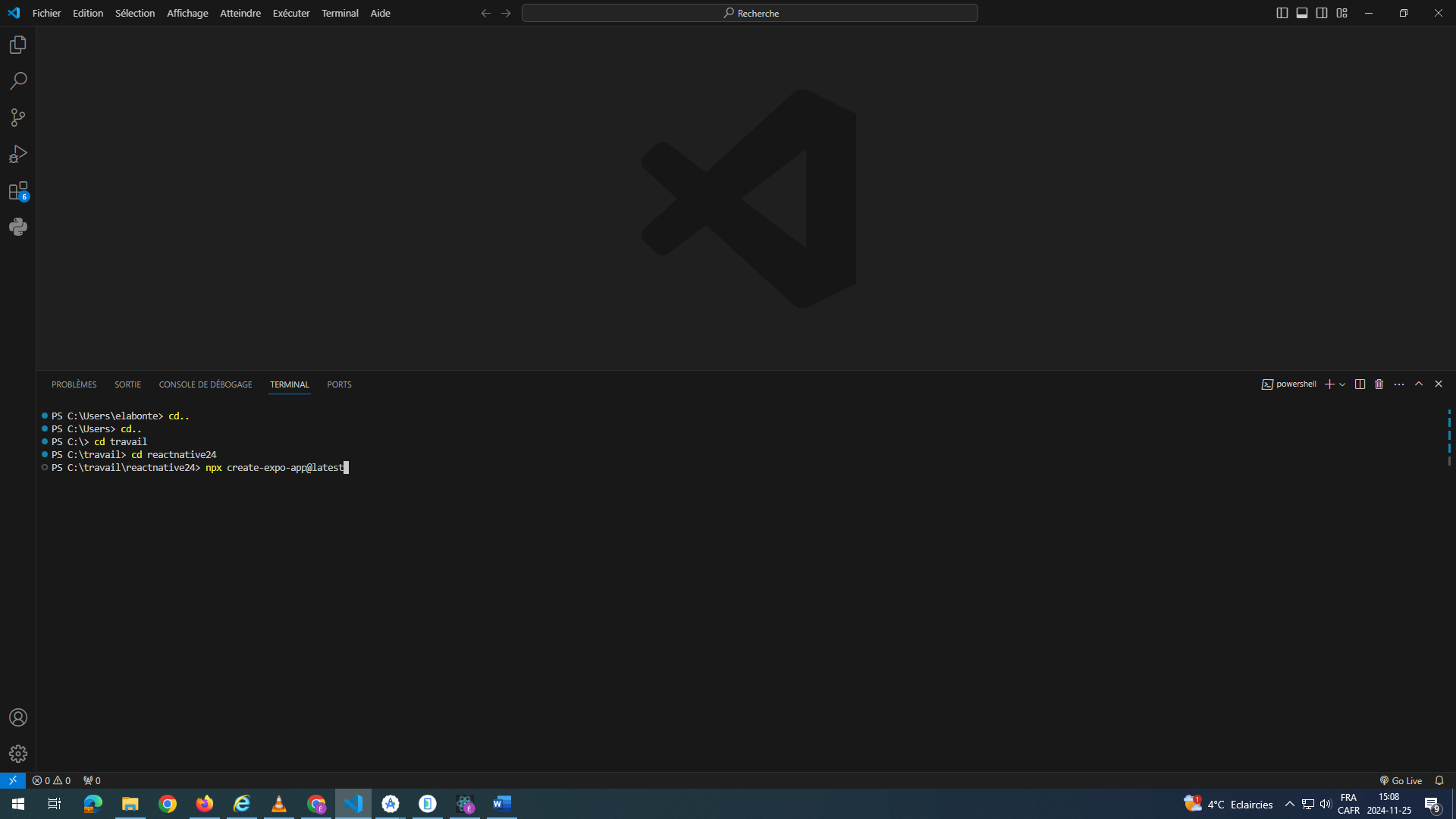
Voir présentation Prezi

## Démarche pour créer un nouveau projet ( AUTOMNE 2024 ) :

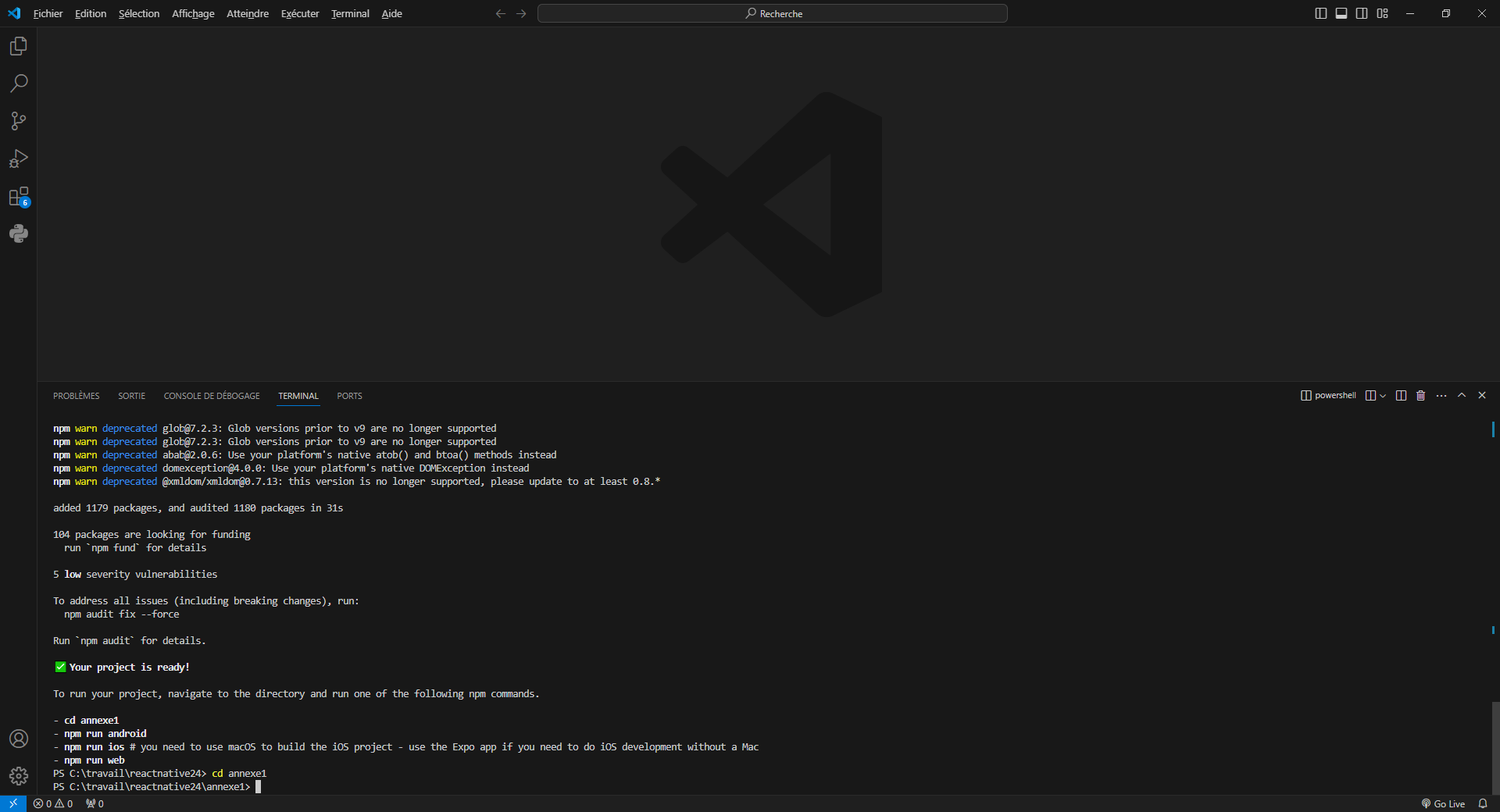
## 1ère fois ( voir l’app exemple de React Native )

1. Démarrez Android Studio ( uniquement pour accéder à un émulateur, au démarrage 🡪 Virtual Device Manager )
2. Dans VS Code, ouvrez un terminal. En pointant à la racine de votre dossier cloné provenant d’un repository GitHub ( dans c:/travail ) , faire :

npx create-expo-app@latest

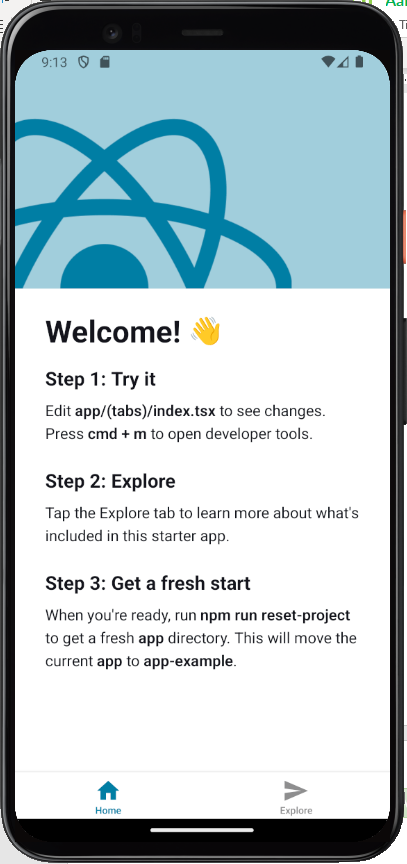


1. Tel qu’indiqué, donner un nom à votre projet
2. Une fois le projet prêt, dans le terminal, déplacez-vous à la racine de votre nouveau dossier projet



1. **Faites npx expo start**. Tapez a pour le démarrer sur un émulateur Android ensuite

Vous aurez le projet de base de ReactNative affiché dans votre émulateur :



## Outils :

npm, npx c’est quoi ?

node.js

npm : node package manager :

-vient avec un cli(command line interface) npm install npm run

- télécharger / installer / gérer des packages / librairies qu’on a bsoin

- grande librairie opensources de composant

Npx exécuter des package directement npx expo start

Metro, c’est quoi ? Java script bundle enveloppe le projet et ses dépendances, permet de recharger l’app rapidement

Expo, c’est quoi ? c’est un framework pour tester/ créer des app en react

## Contenu du dossier-projet actuel :

Dossier node\_modules : en lien avec le fichier package.json

Node modules : l’ensemnle des librairies qu’on peut utiliser pour notre app, fichier package.json repertoire

fichier app/tabs/index.tsx :

* Regarder son contenu, c’est en TSX avec JSX mais on peut décoder / modifier des éléments

Dossier components : comprend des fichier comprenant des composants qui sont exportes

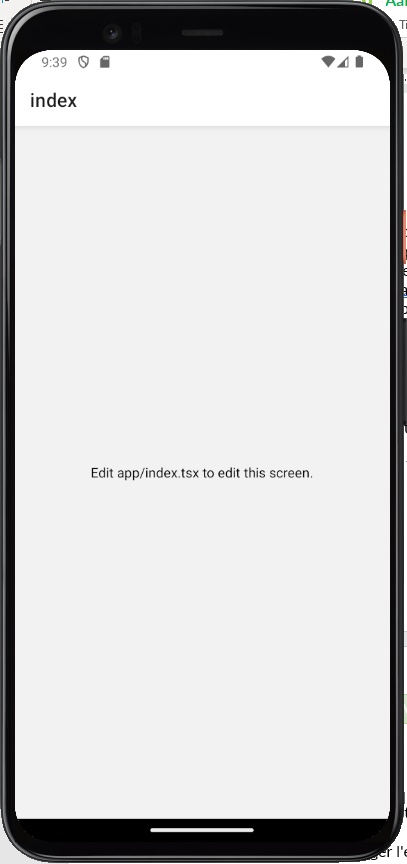
fichier App.json : nom de l’app peut être modifier dans ce fichier lorsqu’on produira

## Un premier composant :

* On peut utiliser des composants React Native pré-établis ou créer les nôtres composés de ces composants pré-établis
* Les composants peuvent être des composants de classe ou des composants fonctionnels ( provenant d’une fonction )
* On va retirer le code par défaut du fichier tsx et créer un composant sans état ( stateless ) simple.

## Pour repartir avec un projet quasi-vide à partir de notre projet initial

* Fermer le dossier sur VSCode
* Dans un nouveau terminal, pointer sur le dossier de notre projet
* **Taper npm run reset-project**
* Ouvrir le projet à nouveau dans VS Code.
* Dans un nouveau terminal, faire npx expo start, puis a ( pour rouler sur android ) , puis r ( pour recharger l’app )
* Vous devriez voir ceci :



Soit un composant simple ne contenant que le titre d'une chanson

Trouvez le fichier index.tsx, changer l'extension pour index.js ( on va travailler en JavaScript plutôt qu'en TypeScript dans ce cours )

## Exercice Créons un composant Chanson, de façon fonctionnelle d'abord, et avec une classe ensuite. Prenez des notes !

Composant fonctionnel :

Composant fonctionnel ( arrow function ) :

Composante de classe :

## JSX

C'est quoi ?

Une syntaxe melange le javaScript et xml

Utiliser par react et react Native

Permet d’exrire des composant dans du javascript

<Text> bonjour </Text>

* { }; pour passer une expression java script

## Donnez un style à notre composant

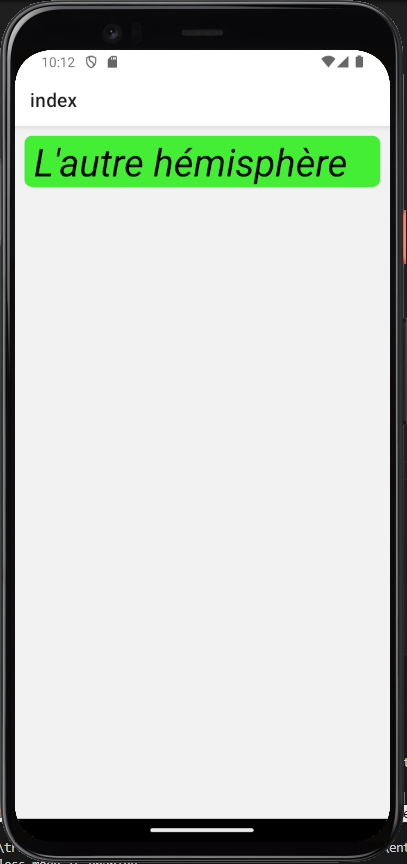
On va créer un objet styles à l'aide de la fonction create de la classe StyleSheet de ReactNative !

* importer StyleSheet de react-native
* faire des règles similaires à des règles CSS

## Exercice :

## Donnez une couleur de fond, une marge et un *borderRadius* à notre objet Chanson

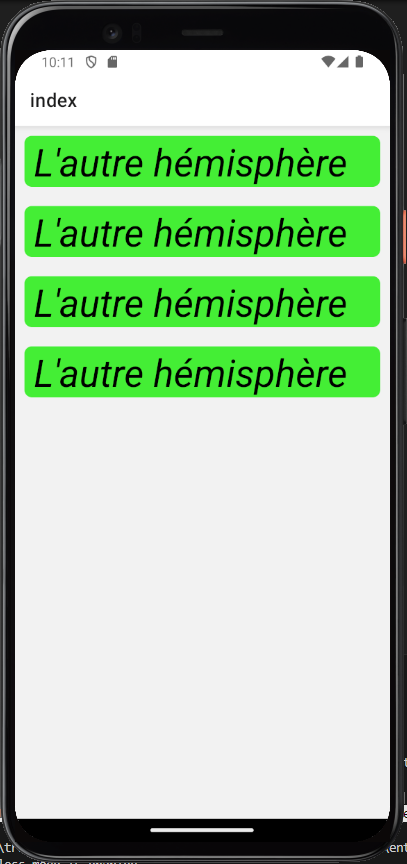
## Donnez une plus grande taille et un style italique au texte à l'intérieur



## 

## Créer un composant à partir d'autres composants

* Exercice Créons, dans le même fichier App.js, un autre composant de classe appelé Playlist qui comprendra 4 Composants Chanson



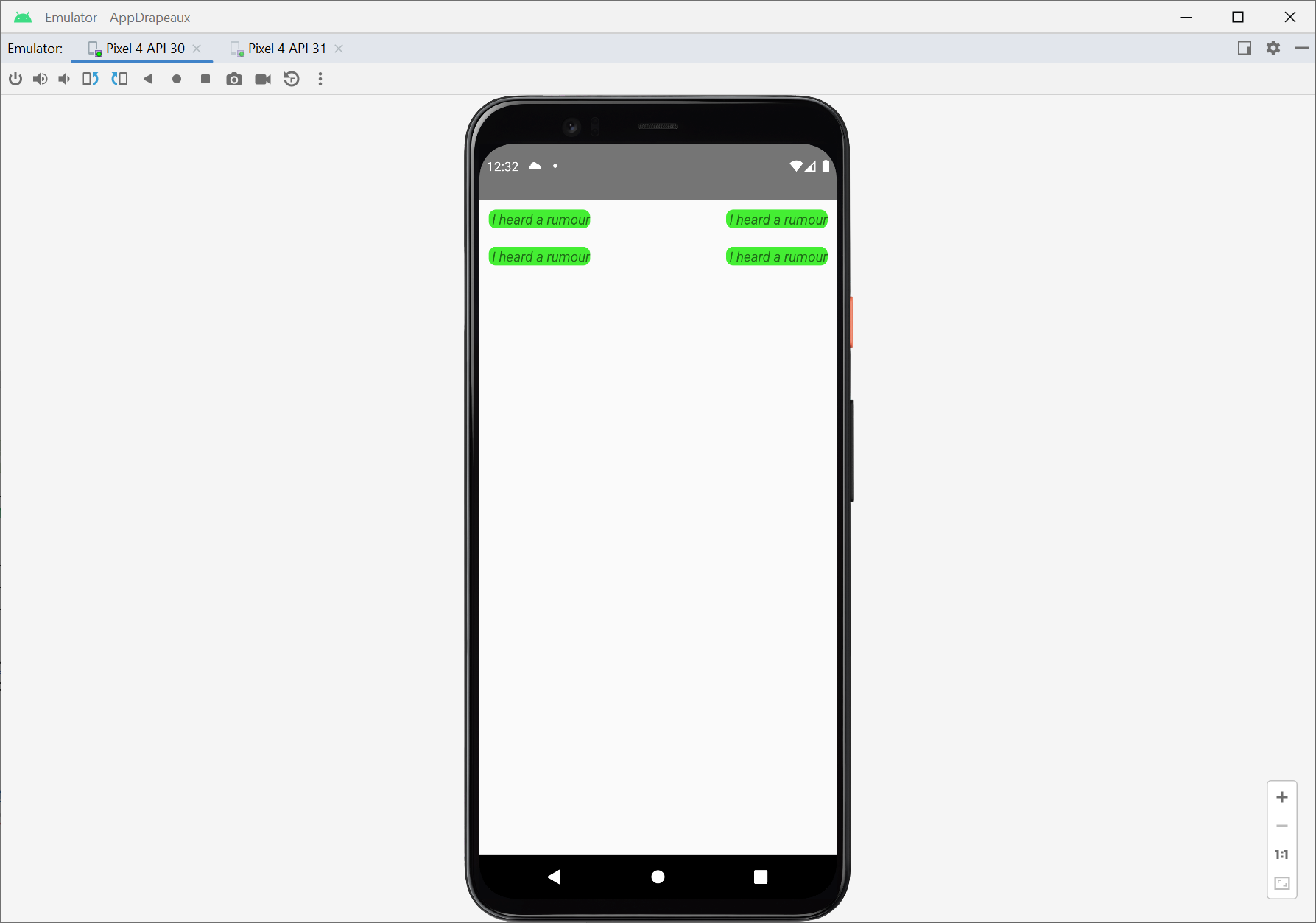
* Le faire en ajoutant des composants
* Le faire en ajoutant des composants avec une boucle

## Organisation des composants sur l'activité

On n'utilise pas de ConstraintLayout ou de LinearLayout avec ReactNative, on utilise plutôt un flexbox comme en CSS.

|  |  |
| --- | --- |
| Différences entre web et React Native pour le flexbox | |
| Web | React Native |
|  |  |
|  |  |
|  |  |

## Exercice : Avec votre connaissance du flexbox, placez les composants Chanson deux par deux sur une ligne comme ceci :



## IMPORTANT : pour conserver votre projet

* + Commit / push de votre repository GitHub
  + Il vient avec son fichier .gitignore qui permettra de ne pas transférer dans votre repository le dossier node\_modules très volumineux

## IMPORTANT : pour retravailler sur un projet déjà amorcé et présent dans votre répertoire git

1. Clonez votre repository dans c:/travail ou c:/travail/ReactNative
2. Dans VS Code, ouvrez le projet
3. Démarrer l’émulateur dans Android Studio
4. Dans un nouveau terminal, en se plaçant sur le répertoire de votre projet, faire npm install expo pour regénérer les fichiers qui n’ont pas été conservé dans GitHub
5. Ensuite, Comme d’habitude, faire npx expo start pour démarrer metro et puis « a » pour démarrer sur android.

## IMPORTANT : pour ne pas avoir à se « débarrasser » du projet par défaut de ReactNative

* Utiliser le projet ModeleDepart présent sur LEA et sur le lecteur V:/elabonte/ModeleDepart
* L’ouvrir dans VS Code
* Dans le terminal, faire npm install expo
* Ensuite comme d’habitude faire npx expo start puis « a » pour démarrer sur Android

## Lien utile ( pour installer à la maison ) :

* Site de React Native : <https://reactnative.dev/docs/environment-setup>