

# CODING CLUB

EPITECH NANTES



## React-Tasking

---

VERSION 1.0

# React Tasking

## Setup ton espace de travail

### Ce dont on a besoin

Tu auras besoin de l'application Expo Go pour réaliser ce workshop

### Lancer ton application

Avant tout, connectes ton ordinateur et ton téléphone sur le même réseau WiFi

Pour lancer l'application, tapes la commande `npx expo start`. Un QR Code est généré pour l'utiliser:

- Sur iOS, scannes le directement depuis la caméra du téléphone
- Sur Android, scannes le depuis l'application Expo Go

# Le React-Native, qu'est ce que c'est ?

## Comment ça fonctionne ?

Le React-Native est un langage de programmation open-source créé par Facebook afin de créer des applications à la fois sur iOS et Android en utilisant une seule base de code. Pour faire simple, quand tu utiliseras une fonction propre à React-Native, ton téléphone utilisera automatiquement la fonction propre à son système pour réaliser ce que tu lui demandes. Ton code sera donc en quelque sorte "traduit" afin d'être compris au mieux par ton téléphone (en Kotlin sur Android et en Swift sur iOS).

En ce qui concerne le code en lui même, c'est un framework JavaScript, ce qui veut dire qu'il y a beaucoup de similitudes avec le JavaScript que tu connais peut-être.

## Qu'est ce que tu vas faire ?

Pour te présenter ce langage, tu vas réaliser une petite application de ToDo List qui te permettra de découvrir comment créer une application pour t'aider à gérer tes tâches personnelles dans ta journée !

## Par où commencer ?

L'architecture front-end du react-native est très similaire à l'HTML que tu connais peut-être sauf que les balises n'ont pas les mêmes noms. Voici quelques exemples:

- `<View/>`, c'est l'équivalent d'une `<div/>`
- `<Text/>`, l'équivalent de `<span/>`

Il existe aussi des balises liées au développement mobile. Certaines qui sont utilisées sont décrites dans le fichier `App.tsx`.

# L'aventure commence

## Ajouter un fond de couleur dégradé

Dans `App.tsx`, tu trouveras le code principal de l'application.

Pour commencer, `ligne 101` : Changes la balise et rajoute une variable qui permettra d'avoir un fond dégradé dans ton application, pour faire ça regarde comment utiliser les [LinearGradient](#)

## Ajoute une image

Toujours dans `App.tsx`, `ligne 110`.

Ajoute une image de fond que tu vas stocker dans le dossier `assets`. Si tu veux une image qui convient déjà bien, prends celle qui s'appelle `illustration.png`.

Trouves un moyen de rajouter cette image (tu peux chercher sur la documentation officielle de `react-native` [ici](#)), et donnes lui le style `styles.image` que tu peux retrouver en bas du fichier `App.tsx`

## Créées tes propres tâches

Si tu as remarqué, l'application contient déjà une zone pour pouvoir entrer du texte.

Cependant, si tu appuies sur le + après avoir tapé du texte, il ne se passe rien.

Essaies de trouver comment faire dans `App.tsx`, `ligne 71`

Pour ça, essayes de faire ces différentes étapes:

- Récupères tous les éléments du tableau, cherches avec le `spread operator` de JavaScript, tu peux trouver la doc [ici](#)
- Changes la valeur du tableau en utilisant la fonction `setAllTasks` définie `ligne 59` (la nouvelle valeur étant ce que tu viens de récupérer et le contenu de la zone de texte).

Si tu vois ta nouvelle tâche dans le terminal de ton ordinateur, ça veut dire que tu as réussi ! 🎉

**Bonus** Si une nouvelle tâche est vide ou remplie d'espaces, ne le rajoute pas !

## Affiches tes tâches

Actuellement, l'affichage de tes tâches n'est pas très beau, désolé. Essayes d'améliorer ça en changeant les styles présents dans le fichier `Components/Task.tsx`. Les styles react-native sont globalement comme le CSS, sauf que l'on utilise le camelCase. Par exemple, la propriété `background-color` sera remplacée par `backgroundColor` tu peux trouver l'explication officielle [ici](#).

## Supprimes des tâches

Pour t'y retrouver, ce serait bien de pouvoir supprimer tes tâches une fois faites. Pour ça, implémentes la suppression d'une tâche dans `App.tsx` ligne 90.

Pour ça, on va encore découper 3 étapes claires:

- Récupères tous les éléments du tableau dans une variable, cherches avec le `spread operator` de JavaScript, tu peux trouver la doc [ici](#)
- Retires la tâche située à l'`index` voulu du tableau, cherches comment faire ça avec les [différentes fonctionnalités](#) qu'offre les tableaux en JavaScript
- Changes la valeur du tableau en utilisant la fonction `setAllTasks` définie ligne 59.

# Et maintenant ?

## Personnalises encore ton app

Si tu as finis, il te reste encore pleins de choses que tu peux faire !

Tu peux par exemple:

- Ajouter une description a chaque tâche
- Changer les icônes utilisées
- Ajouter un retour haptique de ton téléphone quand tu ajoutes/supprimes une tâche (regardes la doc [ici](#))

## Pour aller plus loin

React-Native n'est qu'un des nombreux langages qui existent pour créer des applications mobiles. On peut citer Flutter qui est développé par Google et qui a la côte en ce moment (qui utilise la base du langage Dart), .NET-MAUI qui est le langage venant de Microsoft (qui utilise le C#) ou encore AngularJS qui est aussi un framework JavaScript.

Tu peux essayer de refaire l'application avec ces différents langages pendant ton temps libre, ce qui te permettra de voir les différences, les forces et les faiblesses de chacun d'entre eux !