

TP - Convertisseur de poids

Objectif :

Créer une application REACT JS avec les notions vues en classe cette semaine. Dans cette application, vous devrez donner la possibilité à l'utilisateur de convertir un poids en kilos en LBS (pounds), comme ci-dessous :



The screenshot shows a dark blue rectangular interface with a white border. At the top, the title 'Kilos to Pounds convertor' is centered in white. Below the title, there are two input fields. The first is labeled 'Enter weight in kilos' and contains the value '1001'. The second is labeled 'Enter weight in pounds' and contains the value '2202'. At the bottom, a message reads 'The weight is higher than one tonne !', where the word 'higher' is highlighted in red.

Les étapes à suivre

Etape 1 - création de l'application

Créer une application de zéro from scratch avec la commande npx.

Etape 2 - Création des composants

- App
- Header
- Input
- Footer

Etape 3 - création de la classe App

Créer un composant classe App en mettant en place deux variables d'état :

- Weight (le poids de départ pour le calcul)
- Unit (en quoi on veut convertir le poids) :
- si `state.unit === 'k'` le calcul sera `state.weight * 0.45`
- si `state.unit === 'l'` le calcul sera `state.weight * 2.2`

TP - Convertisseur de poids

Vous devrez créer deux méthodes :

- **toLbs()** : méthode qui renvoie un poids en LBS en partant d'un poids en kilos
- **toKilos()** : méthode qui renvoie un poids en Kilos en partant d'un poids en LBS

Le calcul est le suivant

- Pour calculer un poids en LBS : $\text{weight (kg)} * 2.2$
- Pour un calculer un poids en Kilos : $\text{weight(lbs)} * 0.45$

Créer une méthode pour faire la conversion **convert()** :

Cette méthode attendra en argument le poids à convertir, et la méthode à appeler pour calculer le poids convertis.

Dans cette méthode :

- Vous vérifierez si le poids à convertir est bien un chiffre (`Number.isNaN()` en JS)
- Vous appellerez la bonne méthode de conversion (`toLbs()` ou `toKilos()`)
- Vous arrondirez le résultat à deux chiffres après la virgule
- La méthode renverra le poids converti

Vous devrez mettre en place deux autres méthodes :

- **handleKilosChange()** : cette méthode mettra les variables d'état à jour (unit ici aura la valeur 'l', et dans Weight on mettra la valeur du champs kilos)
- **handlePoundsChange()** : cette méthode mettra les variables d'état à jour (unit ici aura la valeur 'k', et dans Weight on mettra la valeur du champs pounds (lbs))

Pour terminer au niveau de la méthode **render()**, vous variabiliserez le poids qui a été rentré (`State.weight`) et l'unité de mesure pour le calcul (`State.unit`)

Vous poserez une condition pour calculer à chaque fois le poids en kilos et en pounds.
Vous déclarerez deux variables, kilos et pounds.

`const kilos = si state.unit === 'k' il faudra appeler la méthode convert(weight, this.toKilos) sinon la valeur de la const kilos ce sera weight.`

`const pounds = si state.unit === 'l' il faudra appeler la méthode convert(weight, this.toLbs) sinon la valeur de la const pounds ce sera weight.`

Toujours dans le **render()** de App vous ferez référence à :

<Header />

<Input />

<Input />

<Footer />

Vous devrez passer au premier composant `<Input/>` une props `weight` avec le poids en kilos calculé ainsi qu'une props **onWeightChange** qui appellera la méthode **handleKilosChange()**.

Vous devrez passer au deuxième composant `<Input/>` une props `weight` avec le poids en pounds calculé ainsi qu'une props **onWeightChange** qui appellera la méthode **handlePoundsChange()**.

TP - Convertisseur de poids

Vous devrez passer au composant Footer une props qui contiendra le poids en kilos calculé.

Etape 4 - création du composant Input

Vous allez capter l'événement onChange sur le champ, et appeler la props onWeightChange défini dans App au Niveau du composant <Input/> en lui passant en paramètre la valeur actuelle du champ.

Etape 5 - création du composant Footer

Vous devrez récupérer la valeur de la props passé depuis le composant App (le poids en kilos calculé dans l'application).

Vous ferez une condition dans le code JSX en précisant :

- si ce poids en kilos est supérieur à 1 tonne alors vous affichez : 'Le poids est supérieur à une tonne' avec le mot supérieur en rouge
- si ce poids en kilos est inférieur à 1 tonne alors vous affichez : 'Le poids est inférieur à une tonne' avec le mot inférieur en vert

Conseil pratique :

Commencer par gérer un seul calcul, de Kilos à LBS.