

LA MANU

ÉCOLE SUPÉRIEURE
DES MÉTIERS DU NUMÉRIQUE



States – Hooks - Cycle de vie

States (états)

Baseline optionnelle

Qu'est-ce qu'un state

Un state est une variable locale d'un composant dont la modification provoque une nouveau rendu de ce dernier.

Il existe deux type de composants :

- Stateless component (composant sans state)
- Statefull component (composant avec state(s))

Comment ça marche ?

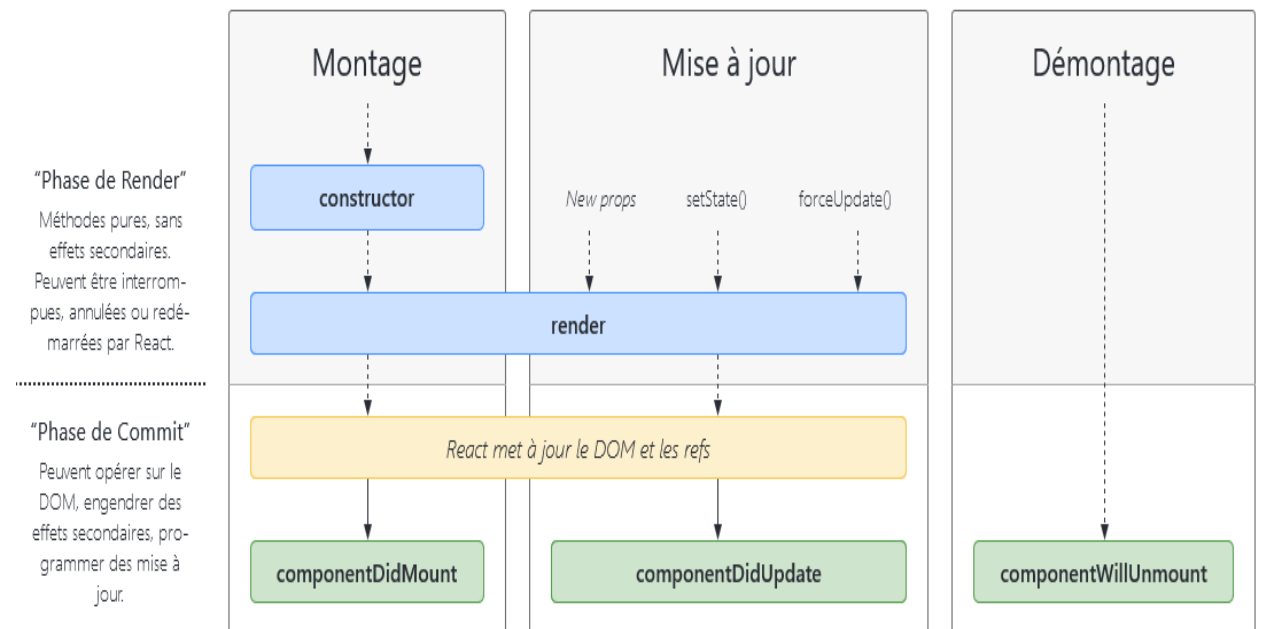
```
const HomeScreen = () => {  
  //Création d'un state.  
  //title est le nom de la constante et setTitle est la fonction //permettant de changer cette  
  constante.  
  const [title, setTitle] = useState('Bonjour')  
  return (  
    //Affichage du state dans une balise Text  
    <Text>{title}</Text>  
  );  
};
```

Cycle de vie

Baseline optionnelle

Cycle de vie

- Un composants peut avoir 3 fonctions différentes pour gérer son cycle de vie.
- `componentDidMount()` : Permet de faire une ou plusieurs action au chargement du composant.
- `componentDidUpdate()` : Permet de faire une ou plusieurs action à la mise à jour du composant.
- `componentWillUnmount()` : Permet de faire une ou plusieurs action au démontage du composant.
- Pour simplifier tout cela, il à été inclus à React et React-Native les hooks.



Voir le projet sur [GitHub](#)

A Savoir

Un state ne doit jamais être muté directement, cela risque de ne pas déclencher, comme prévu, un nouveau rendu.

Pour des raisons de performances, les mises à jour des states peuvent être asynchrones.

Lorsque l'état d'un composant évolue, les anciens et nouveaux états sont fusionnés, ce qui permet à React de revenir sur un état précédent.

Hooks

Baseline optionnelle

Qu'est-ce qu'un hook

Les hooks sont des fonctions permettant de manipuler:

- les états,
- le cycle de vie
- une partie de la logique des composants.



Pourquoi les hooks ?

Réduire l'utilisation des classes ES5 (complexes pour les humains comme les machines)

Simplifier la compréhension de composants complexes.

Faciliter la réutilisation de la logique à état entre les composants.

Faciliter l'apprentissage de React et React-Native pour les débutants

useEffect / useLayoutEffect

```
const HomeScreen = () => {  
  //Modification du titre à chaque update  
  useEffect(() =>{  
    // Afficher 'Hello' au chargement de la page  
    console.log('Hello')  
  
    // Afficher 'Bye' au démontage de la page  
    return function(){  
      console.log('Bye')  
    }  
    // Le tableau de dépendances peut recevoir une variable, si  
    // Cette variable est modifiée le contenu de useEffect  
    // S'effectuera à nouveau.  
  }, [])  
  //Le contenu de la page  
  return (  
    <Text>Hello World !</Text>  
  );  
};
```

- **useEffect** est **asynchrone**
- **useLayoutEffect** est **synchrone**.

- Effectué au chargement de la page = **componentDidMount**
- Effectué au démontage de la page = **componentWillUnmount**
- Effectué à la mise à jour de la page = **componentDidUpdate**

useCallback / useMemo

```
const handleClick = useCallback(() =>{  
    setCount((c)=> c++)  
}, [count]);
```

useCallback permet de mémoriser une fonction callback.

La fonction mémorisée ne sera appelée que lorsque count sera modifié.

Si un autre state indépendant est modifié, la fonction ne sera pas redéclarée.

useMemo fait de même mais pour une valeur calculée.

Ce sont deux hooks fait pour optimiser les performances, ils doivent être le plus souvent possible pour des callback transmis aux enfant.

A vous de jouer !

Objectif: découvrir les states et certaines fonctions avancées de react native

- ① Créez un composant **Clock** qui retourne une vue avec un texte centré horizontalement et verticalement.
- ② Ajoutez un state **date** à votre composant avec une valeur initiale à **null**.
- ③ Ajoutez le **hook** « `useEffect` » à votre composant et à l'intérieur de celui-ci lancer un timer (**`setInterval`**) qui permet de donner l'heure courante au state toute les secondes.
- ④ Faites en sorte que le timer s'arrête lorsque votre composant disparaît de l'écran.
- ⑤ Affichez la date au format local « **hh:mm:ss** »

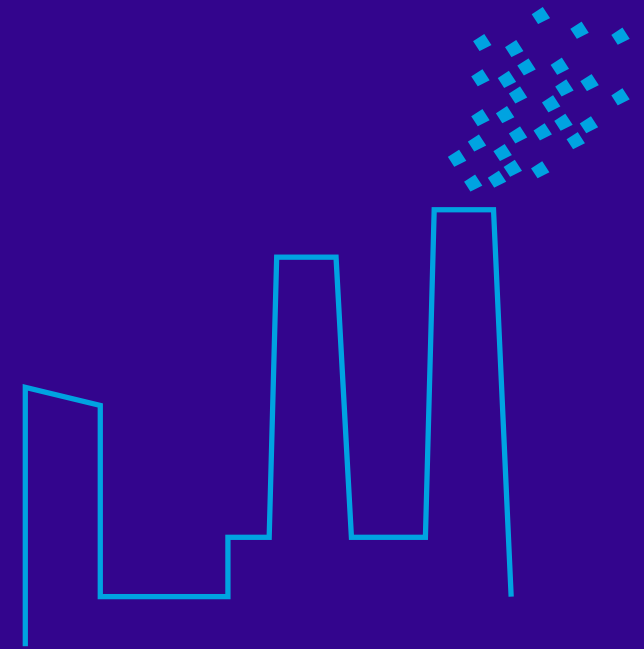
Campus Amiens

contact-amiens@lamanu.fr

<https://lamanu.fr>

70 rue des Jacobin, 80090 Amiens

09 86 27 17 04



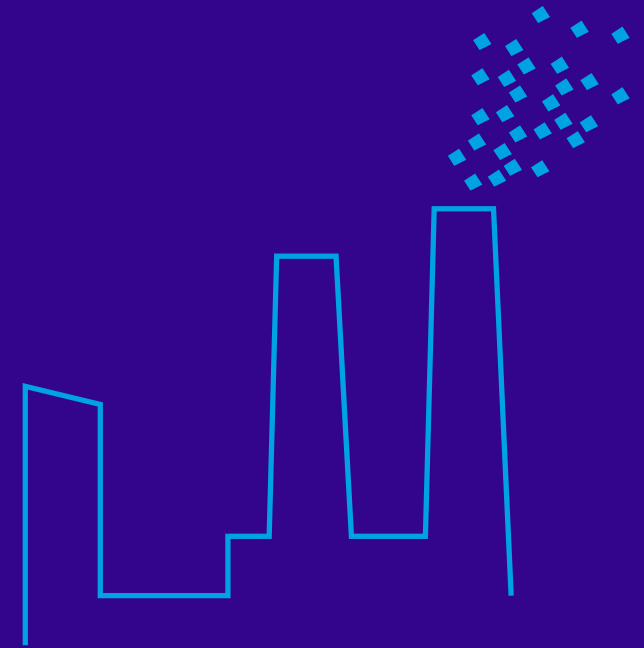
Campus Versailles

contact-versailles@lamanu.fr

<https://lamanu.fr>

143 rue Yves le Coz, 78000 Versailles

09 86 27 17 04



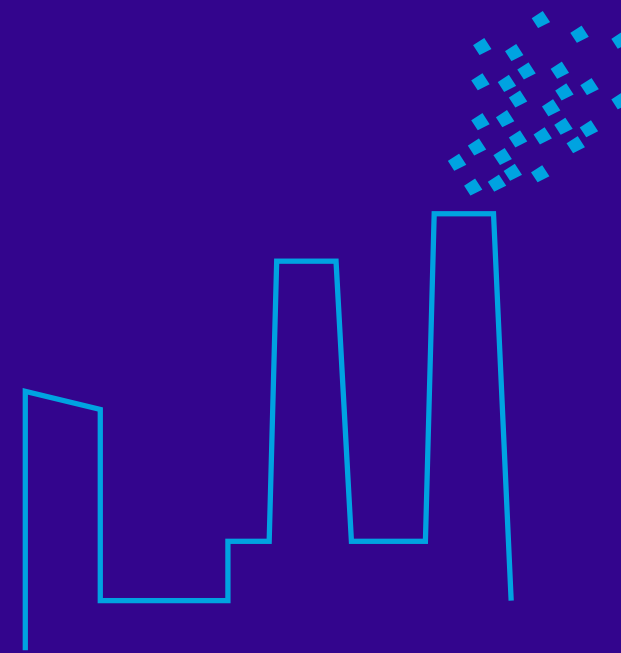
Campus Le Havre

contact-lehavre@lamanu.fr

<https://lamanu.fr>

10 place Léon Meyer, 76600 Le Havre

09 86 27 17 04





ÉCOLE SUPÉRIEURE
DES MÉTIERS DU NUMÉRIQUE



Campus Noyon-Compiègne

contact-noyon@lamanu.fr

<https://lamanu.fr>

1435 Boulevard Cambronne, 60400 Noyon

09 86 27 17 04

