1️⃣ Qu'est-ce qu'un état en React ?

Un **état** (state) est une **valeur qui change au cours du temps** et qui déclenche une mise à jour de l'interface.

💡 **Exemple simple :**
Si tu as un compteur et que tu veux l'augmenter à chaque clic, tu as besoin d'un état pour **mémoriser la valeur actuelle du compteur**.

2️⃣ C'est quoi un composant fonctionnel ?

En React, tout tourne autour des **composants**.

Un **composant fonctionnel** est une fonction **JavaScript** qui **retourne du code HTML**.

📌 **Exemple d'un composant fonctionnel :**

```
function Bonjour() {
    return <h1>Bonjour !</h1>;
}
```

➡️ Ce composant affiche juste du texte et **ne change jamais**.

Mais si on veut un bouton qui change de texte lorsqu'on clique, on a besoin d'un état !

3️⃣ Qu'est-ce qu'un Hook ?

Un **Hook** est une **fonction spéciale** de React qui permet aux **composants fonctionnels** d'avoir un état et d'autres fonctionnalités.

Le **Hook** `useState` est l'un des plus utilisés car il permet de **gérer un état** dans un composant.

4️⃣ Comment fonctionne `useState` ?

`useState` permet à un composant fonctionnel d'avoir un état.

Regarde cet exemple :

```
import { useState } from "react";

function Compteur() {
    const [count, setCount] = useState(0);

    return (
        <div>
```

```
            <p>Compteur : {count}</p>
            <button onClick={() => setCount(count + 1)}>+1</button>
        </div>
    );
}
```

📌 **Explication simple :**

1. `useState(0)` définit un **état initial** (`count = 0`).
2. `setCount` est la **fonction qui met à jour** `count`.
3. Quand on clique sur le bouton, `setCount(count + 1)` augmente `count`.
4. React **redessine** (`re-render`) le composant avec la nouvelle valeur.

Résultat : Le nombre affiché augmente à chaque clic.

5️⃣ Pourquoi `useState` est utile ?

- Sans état (`useState`), un composant **ne peut pas changer** après son affichage.
- Avec `useState`, on peut **mettre à jour l'interface dynamiquement**.

🚀 **En gros,** `useState` **permet de stocker une valeur et de la changer quand on veut !**

📌 En résumé :

Concept	Explication
État	Une valeur qui change (ex : compteur, texte d'un champ, etc.).
Hook	Une fonction spéciale qui ajoute des fonctionnalités à un composant.
`useState`	Un Hook qui permet d'ajouter un état à un composant.
Composant fonctionnel	Une fonction qui affiche du HTML.

💡 Exemple interactif : Formulaire avec `useState`

```
import { useState } from "react";

function Formulaire() {
    const [nom, setNom] = useState("");

    return (
        <div>
            <input
                type="text"
                value={nom}
                onChange={(e) => setNom(e.target.value)}
            />
            <p>Vous avez tapé : {nom}</p>
```

```
            </div>
        );
    }
```

📌 Explication :

- `nom` commence vide (`""`).
- Quand on tape, `setNom(e.target.value)` met à jour `nom`.
- L'écran affiche toujours la valeur actuelle de `nom`.

🔥 Tu veux tester du code React en live ?

1. Va sur CodeSandbox.
2. Sélectionne **"React"** et colle un des codes ci-dessus.
3. Modifie et joue avec ! 🎉