

Exercice 1:

1-en utilisant les fonction (ou lambda function) créer un composant <Livres id='1' titre='abc' prix='110'/>

Et afficher le dans un exemple sous le format

Livre : id:1, Titre:abc, Prix:110

2-Soit le tableau Livres L déclarer globale dans index.js

let L=[{id :1,titre :'abc',prix :110},{id :2,titre :'Light',prix :110},{id :1,titre :'win',prix :110}]

en Utilisant un tableau livre L, et aussi le composant <Livres .. /> précédent, créer et utiliser le Composant <LivresAll T={L}/> acceptant en propriété le tableau des Livres L , et qui affiche les livres sous forme de liste :

- **Livre 1** :id:1 ,Titre:abc Prix:110
 - **Livre 2** :id:2 ,Titre:abc Prix:110
 - **Livre 3** :id:3 ,Titre:abc Prix:110
 - ...

Exercice 2 :

1-Soit un composant class qui a pour rendu :

bienvenu visiteur

Inscription

En utilisant le state dans un composant class on veut changer le rendu au clic sur le button :

Inscription reussi

merci

Code Solution Exercice 2 :

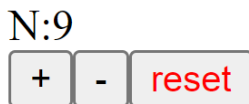
```
export default class ComposantState extends React.Component{
  constructor(props){
    super(props)
    this.state={x: 'bienvenu visiteur',y: 'Inscription'}//declaration state
  }
  changer=(event)=>{
    this.setState({x: 'Inscription reussi',y: "merci"})//modifier state et update du rendu
  }
  render(){
    return <>
      <h1>{this.state.x}</h1>
      <button value="salam" onClick={this.changer}>
        {this.state.y}</button></>
    </>
  }
}
```

2-Repetier question 1 mais maintenant composant function ComposantState et aussi le Hook

useState()

Exercice 3 :

On veut réaliser le compteur de l'interface suivant avec les button +, - et reset



Le composant a les propriétés start valeur de début et step valeur d'incrémentations appelées du composant sera dans index.js comme suit:

```
r.render(<><Compteur start={1}; step={2}/> </>)
```

Exemple de code Solution avec class:

```
import React from "react";
class Compteur extends React.Component{
  constructor(props){
    super(props)
    this.state={cmp: this.props.start,color: 'red' }
  }
  plus=(e)=>{this.setState({...this.state,cmp:this.state.cmp+this.props.step})}
  moins=(e)=>{this.setState({...this.state,cmp:this.state.cmp-this.props.step})}
  reset=(e)=>{this.setState({...this.state,cmp:0})}
  render(){
    return<div><div>N:{this.state.cmp}</div>
      <button onClick={this.plus}>+</button>
      <button onClick={this.moins}>-</button>
      <button onClick={this.reset} style={{color:this.state.color}}>reset </button>
    </div>
  }
}
export default Compteur;
```

-Refait le compteur avec composant fonction et les hook useState appeler le **CompteurFun**

- <CompteurFun start={1} step={2} />