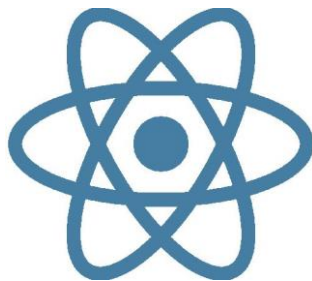


React

Développer des applications Web

Intervenant : Jean-Frédéric VINCENT



The background of the slide is a photograph of a dense forest of evergreen trees, likely spruce or fir, under a hazy, overcast sky. A semi-transparent green rectangular overlay is positioned horizontally across the middle of the image. On the left side of this overlay, the text 'Module 6' is written in a white, sans-serif font. Below it, the title 'Création d'un composant' is written in a larger, bold, white, sans-serif font.

Module 6

Création d'un composant

Création d'un nouveau composant

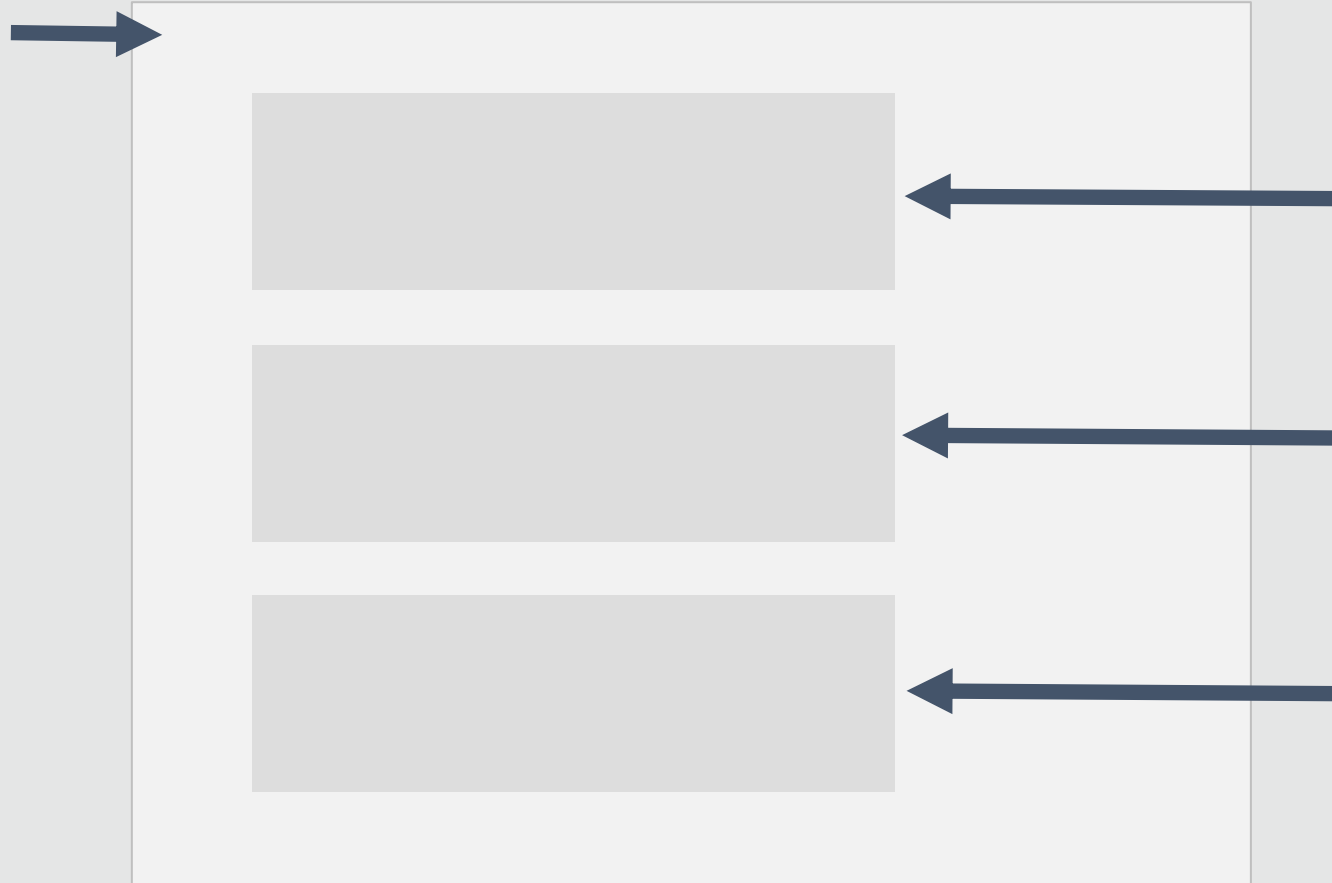
les Props

Les Props et les fonctions

Mise en place de use Effect

Création d'un composant

Imbrication des composants




Création d'un composant

Création d'un composant

Création du répertoire **components**

Création du fichier **Personne.tsx**

```
function Personne() {  
  return (  
    <h1>Je suis le component Personne !</h1>  
  )  
}  
export default Personne
```



- > public
- ▼ src
 - > assets
 - ▼ components
 - ⚛ Personne.jsx
 - # App.css
 - ⚛ App.jsx
 - # index.css
 - ⚛ main.jsx
- ⚙ .eslintrc.cjs
- 📄 .gitignore
- <> index.html
- { } package-lock.json
- { } package.json
- 📖 README.md
- JS vite.config.js

Création d'un composant

Appel du composant enfant créé

Depuis le component mère **App.tsx** ,On appelle le component **Personne.tsx**
Les **import**

1. Importer le component
2. Appeler en le component

```
import Personne from './components/Personne.jsx'

function App() {
  return (
    <>
      <Personne />
      <Personne />
      <Personne />
    </>
  )
}
export default App
```

Je suis le component Personne !
Je suis le component Personne !
Je suis le component Personne !

Création d'un composant

Démonstration

Les Props

Création d'un composant

Les props

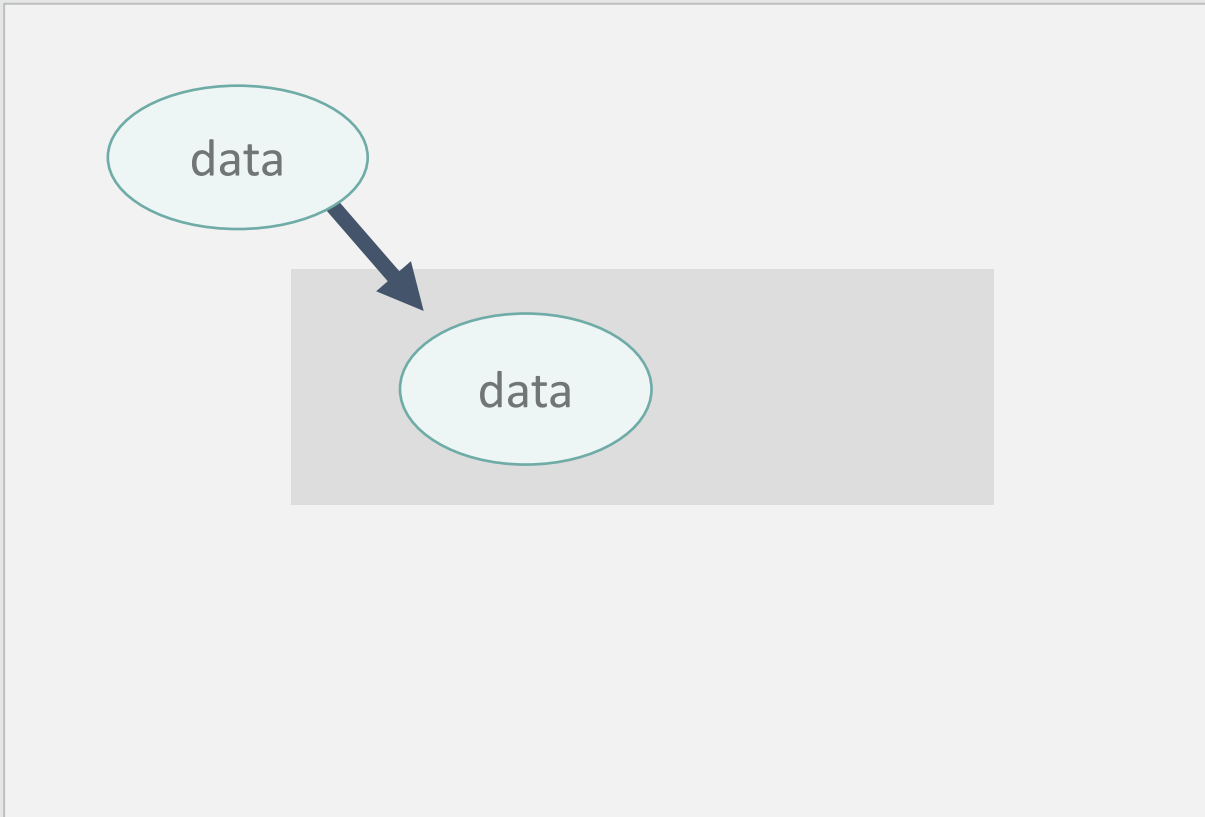
Les **props** permettent :

- D'envoyer des variables
- Des références de fonctions

Création d'un composant

Envoyer des données au composant

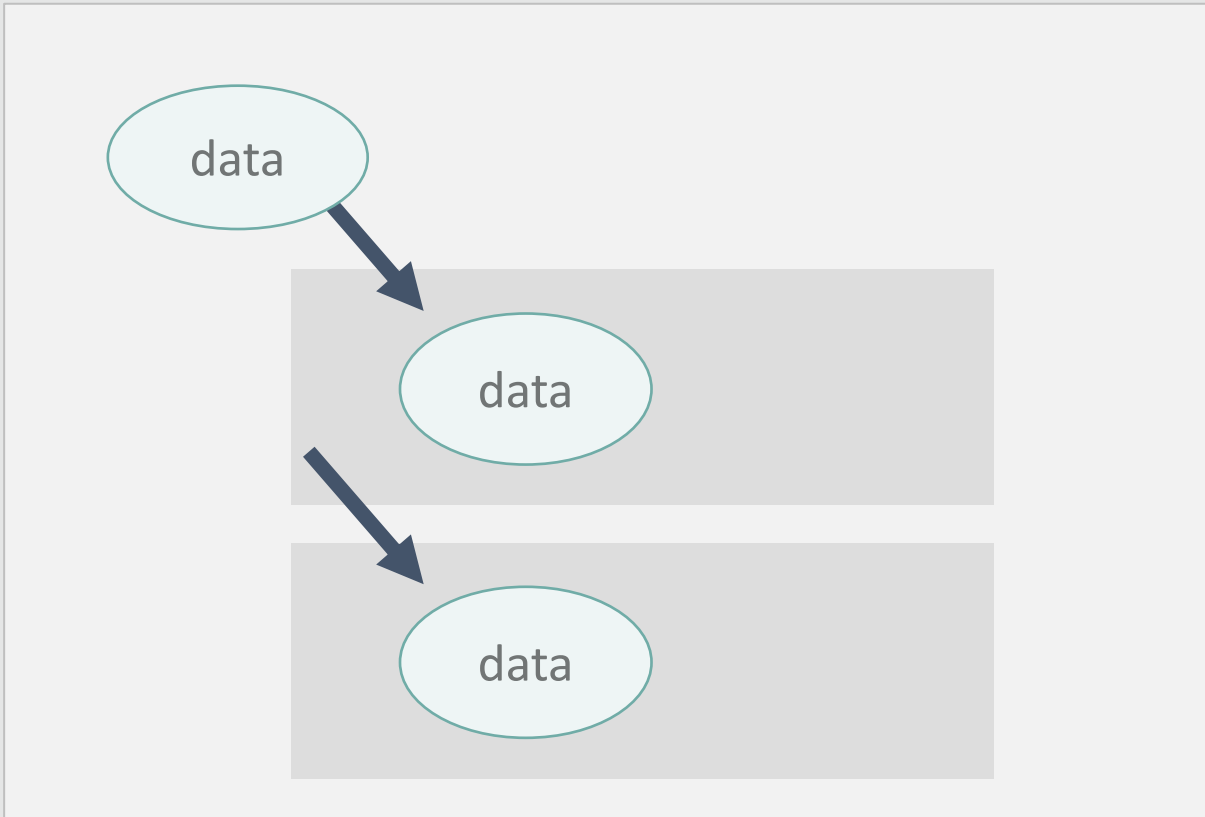
Les **props** permettent d'envoyer des données : des variables



Création d'un composant

Envoyer des données au composant

Les **props** permettent d'envoyer des données : des variables




Création d'un composant

Les props et fonctions

Envoyer des données en "**dure**"

```
import Personne from './components/Personne.jsx'

function App() {
  return (
    <>
      <Personne prenom="Brad" nom="PITT" />
      <Personne prenom="Nicolas" nom="CAGE" />
    </>
  )
}
export default App
```

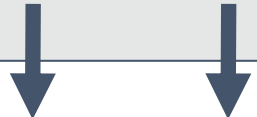


Création d'un composant

Réception des données "props"

Réception des props dans le composant enfant **Personne.tsx**

Personne.tsx



```
function Personne({prenom,nom}) {  
  return (  
    <h1>Je suis {prenom} {nom} !</h1>  
  )  
}  
export default Personne
```

Création d'un composant

Réception des données "props"

Réception des props dans le composant enfant **Personne.tsx**

Personne.tsx



```
function Personne(props) {  
  return (  
    <h1>Je suis {props.prenom} {props.nom} ! </h1>  
  )  
}  
export default Personne
```

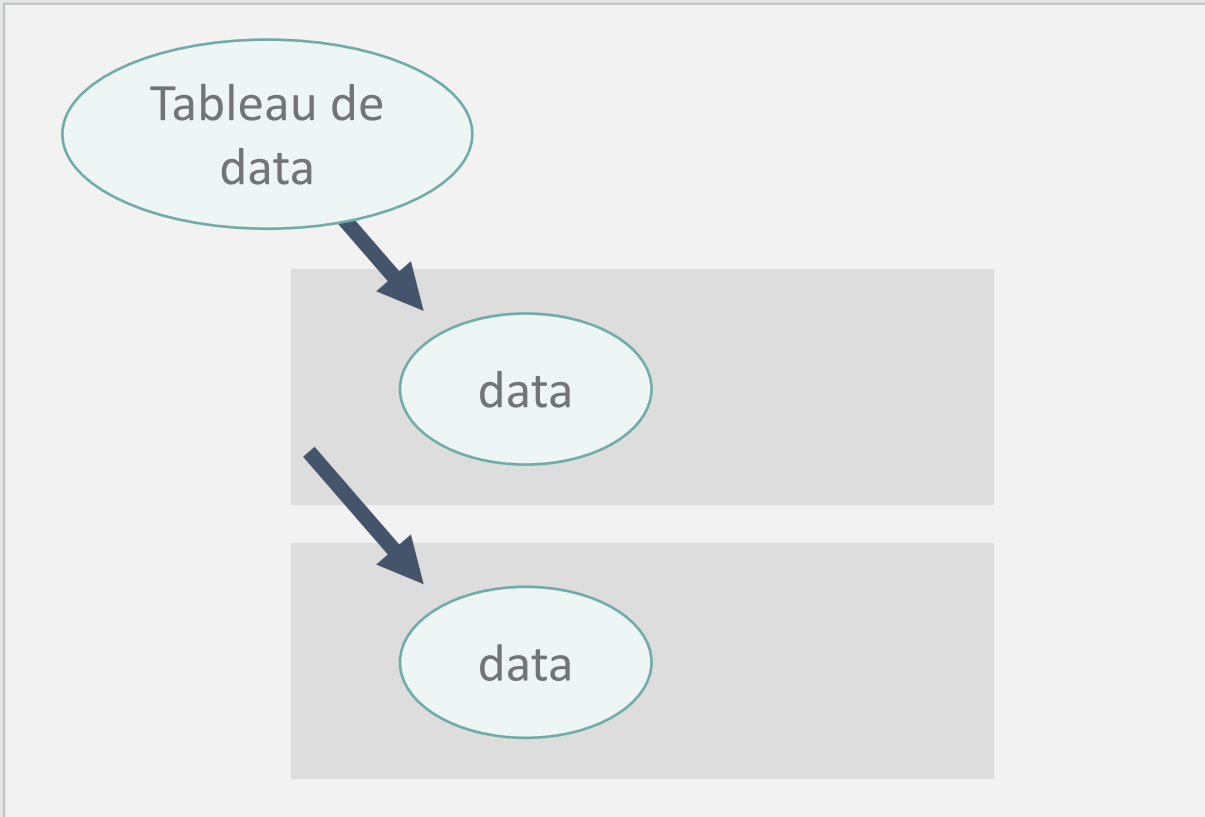
Création d'un composant

Démonstration

Création d'un composant

Envoyer des données au composant

Envoyer chaque élément d'un tableau

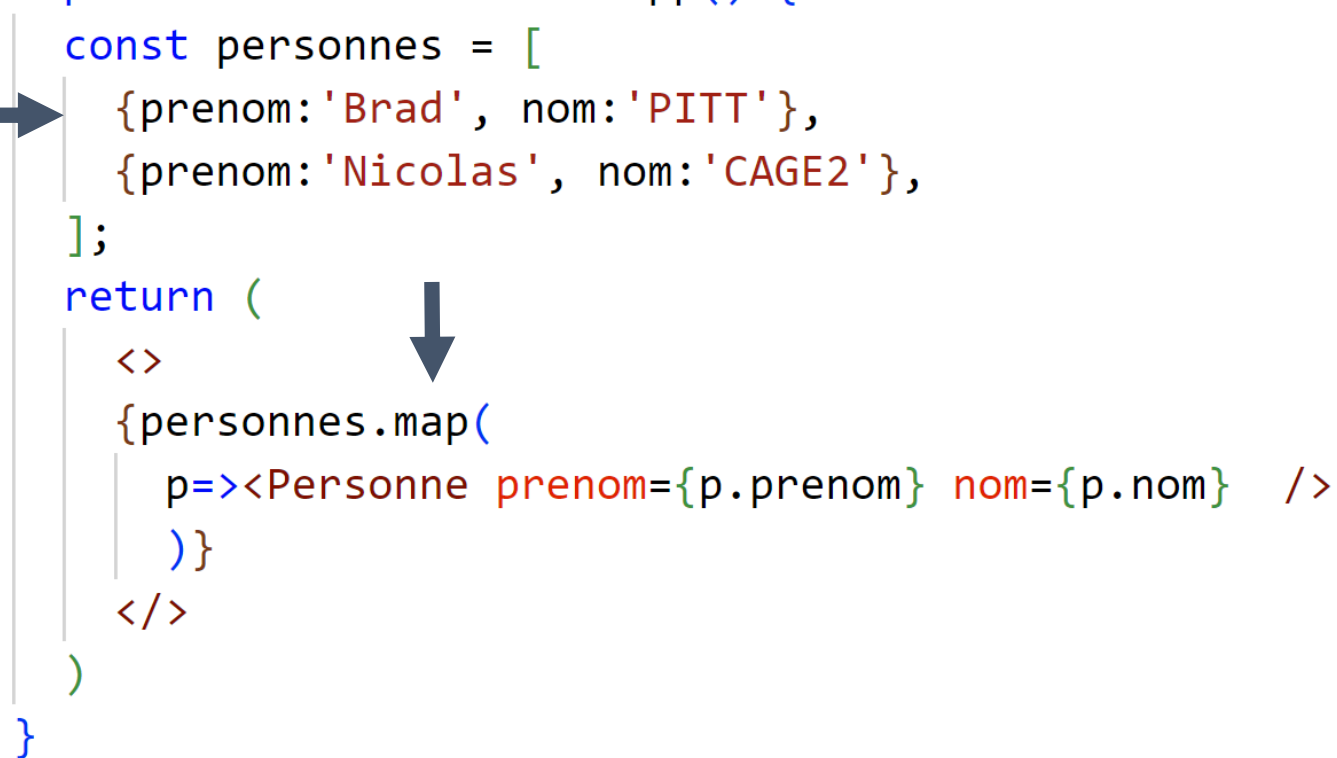


Création d'un composant

Envoyer un tableau dans les props

Main.tsx

```
import Personne from './components/Personne.jsx'
export default function App() {
  const personnes = [
    {prenom: 'Brad', nom: 'PITT'},
    {prenom: 'Nicolas', nom: 'CAGE2'},
  ];
  return (
    <>
      {personnes.map(
        p=><Personne prenom={p.prenom} nom={p.nom} />
      )}
    </>
  )
}
```



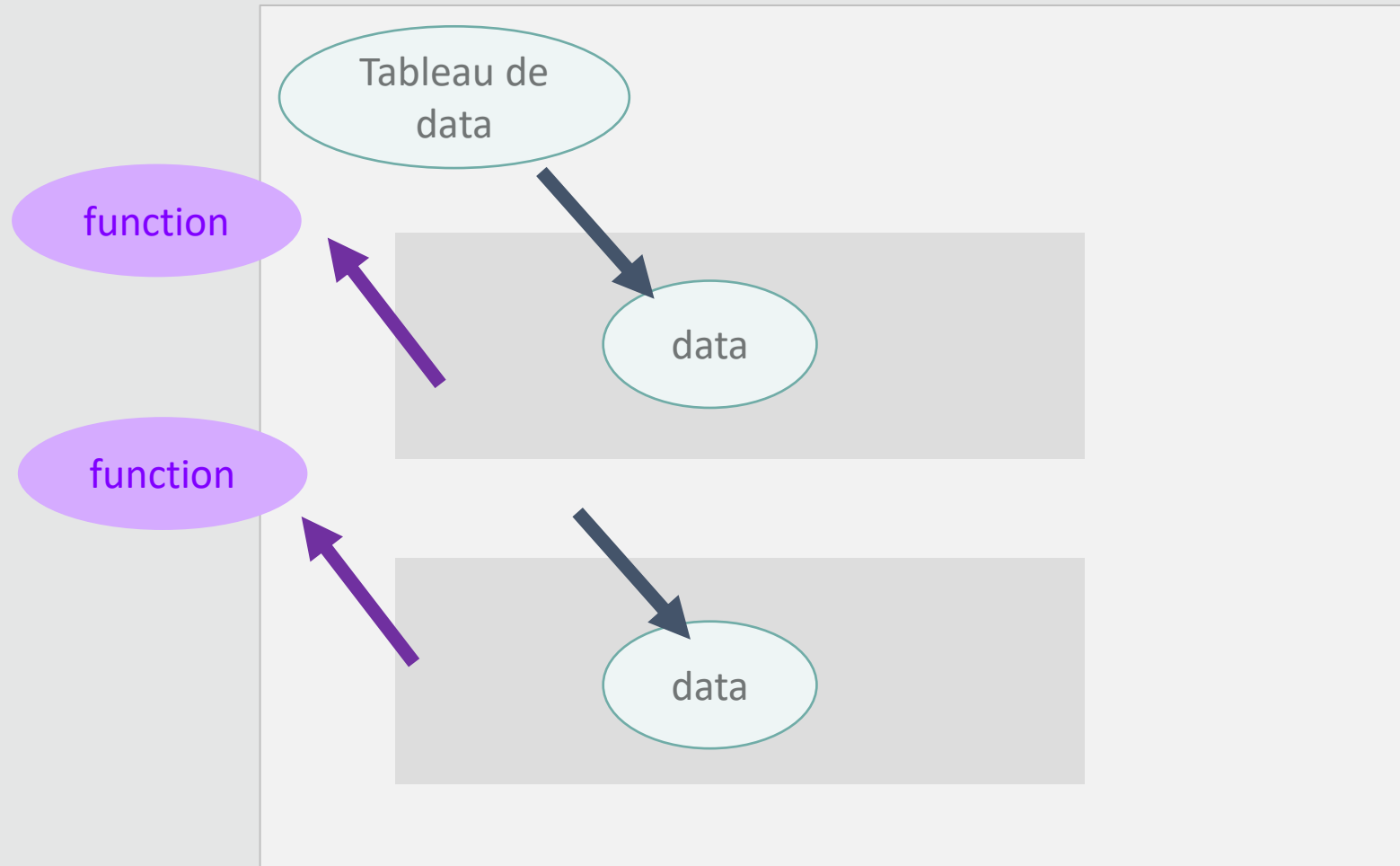
Création d'un composant

Démonstration

Les Props et les fonctions

Création d'un composant



Appeler une fonction du composant mère



Création d'un composant

Les props et fonctions

Appeler une fonction du composant mère *depuis le composant (enfant)*

```
const qui=(indice)=>{  Main.tsx
  console.log(indice);
  console.log(personnes[i]);
}
return (
  <>
    {personnes.map(
      (p,index) =>
        <Personne
          key={index}
          prenom={p.prenom} nom={p.nom} indice={index}
 qui={qui} />
        )}
    </>
  )
}
```

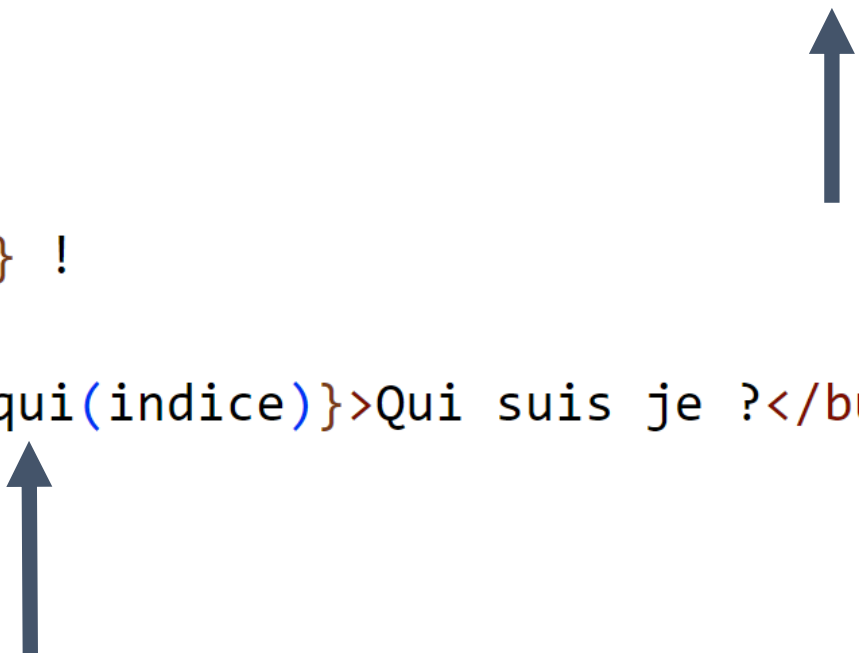
Création d'un composant

Les props et fonctions

Appeler une fonction du composant mère *depuis le composant (enfant)*

Main.tsx

```
export default function Personne({ prenom, nom, indice, qui }) {  
  return (  
    <>  
      <h1>  
        Je suis {prenom} {nom} !  
      </h1>  
      <button onClick={() => qui(indice)}>Qui suis je ?</button>  
    </>  
  );  
}
```



Création d'un composant

Démonstration

Rendre le tableau modifiable

Utiliser useEffect

Création d'un composant

Les props et fonctions



Main.tsx

```
import {useState, useEffect} from 'react'
import Personne from '../components/Personne.jsx'
```

```
export default function App() {
  const [personnes,setPersonnes] = useState([]);
  useEffect(()=>{ ←
    let personnes = [
      {prenom:'Brad', nom:'PITT'},
      {prenom:'Nicolas', nom:'CAGE2'},
    ];
    setPersonnes(personnes)
  },[])
  ↑
  const qui=(indice)=>{
    personnes.splice(indice,1);
    setPersonnes([...personnes])
  }
}
```

→ **UseEffect** permet de venir modifier au lancement du rendu le tableau

→ **[]** , précise de le faire uniquement au premier rendu

Création d'un composant

Démonstration