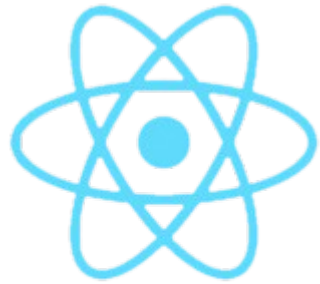


Présentation IHM



React

Une bibliothèque Javascript pour le front-end

Les Composants

- Les composants sont la base de React : ce sont des parties de l'interface utilisateur
- Ils permettent de factoriser le code car il sont réutilisables
- Et de l'organiser facilement en plusieurs fichiers (avec des *imports/exports*)
- On utilise un composant ainsi : `<ComposantName/>`
- Les composants sont des fonctions Javascript (exemple dans les slides suivantes)

JSX

- JSX est une extension de syntaxe pour Javascript qui permet d'écrire du HTML* au sein d'un fichier JavaScript.
- *Quelques différences avec HTML :
 - Toutes les balises doivent être fermées explicitement (devient)
 - Certains attributs ont un nom légèrement différent (par exemple 'class' devient 'className')

Exemple de composants

// Square.jsx

```
1 export default function Square() {  
2   return <button className="square">X</button>;  
3 }  
4 |
```

// App.jsx

```
1 export default function App() {  
2   return (  
3     <>  
4       <Square />  
5       <Square />  
6     </>  
7   )  
8 }
```



Les Fragments

- Un composant ne peut renvoyer qu'une balise JSX
=> Les fragments servent à regrouper des balises JSX dans une même balise parente.
- Ils ressemblent à : `<> //votre code ici </>`
- Le fragment ne sera pas ajouté au DOM

Les *Props*

- Elles permettent de modifier un composant
- Ce sont les paramètres que l'on passe à la fonction qui constitue le composant
- Elles permettent aussi aux composants de communiquer entre eux

Exemple de *props*

- Le composant enfant reçoit les *props* en paramètres :

```
1 export default function Square({symbol}) {  
2   return (  
3     <>  
4       <button className="square">{symbol}</button>  
5     </>  
6   );  
7 }
```



- On passe les *props* de Square depuis le composant parent App :

```
<Square symbol="X" />
```

```
<Square symbol="O" />
```

Exemple de *props*

- Le composant enfant reçoit les *props* en paramètres :

```
1 export default function Square({symbol, color}) {  
2   return (  
3     <>  
4       <button style={{backgroundColor: color}}>{symbol}</butt  
5     </>  
6   );  
7 }
```



- On passe les *props* de Square depuis le composant parent App :

```
<Square symbol="X" color="yellow"/>
```

```
<Square symbol="O" color="red"/>
```

λ Les *props* ne peuvent être passées que de composant parent à composant enfant

Les « useState »

- Ce sont des variables d'état d'un composant
- Quand elles sont modifiées, React déclenche un nouveau rendu du composant
- Cela permet de le rendre réactif à des événements

```
1  import { useState } from "react";
2
3  export default function Square({ color }) {
4    const [symbol, setSymbol] = useState("X")
5
6    function handleClick() {
7      setSymbol("O");
8    }
9    return <button style={{ backgroundColor: color }} onClick={handleClick}> {symbol} </button>
10 }
```

C'est l'heure du TP !

- Consignes du TP et code de départ :
<https://github.com/Bloubing/trois-petits-cochons>
- Des branches pour chaque étape du TP