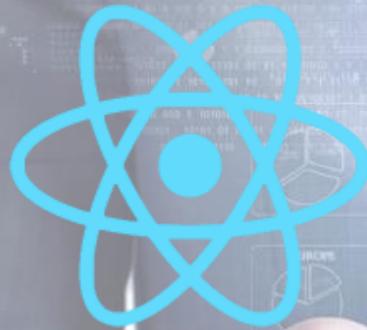


Formation



React

Séance 2

Petit récapitulatif

- create-react-app <nom du projet>
- npm start

Import

Constructor

State

Méthode

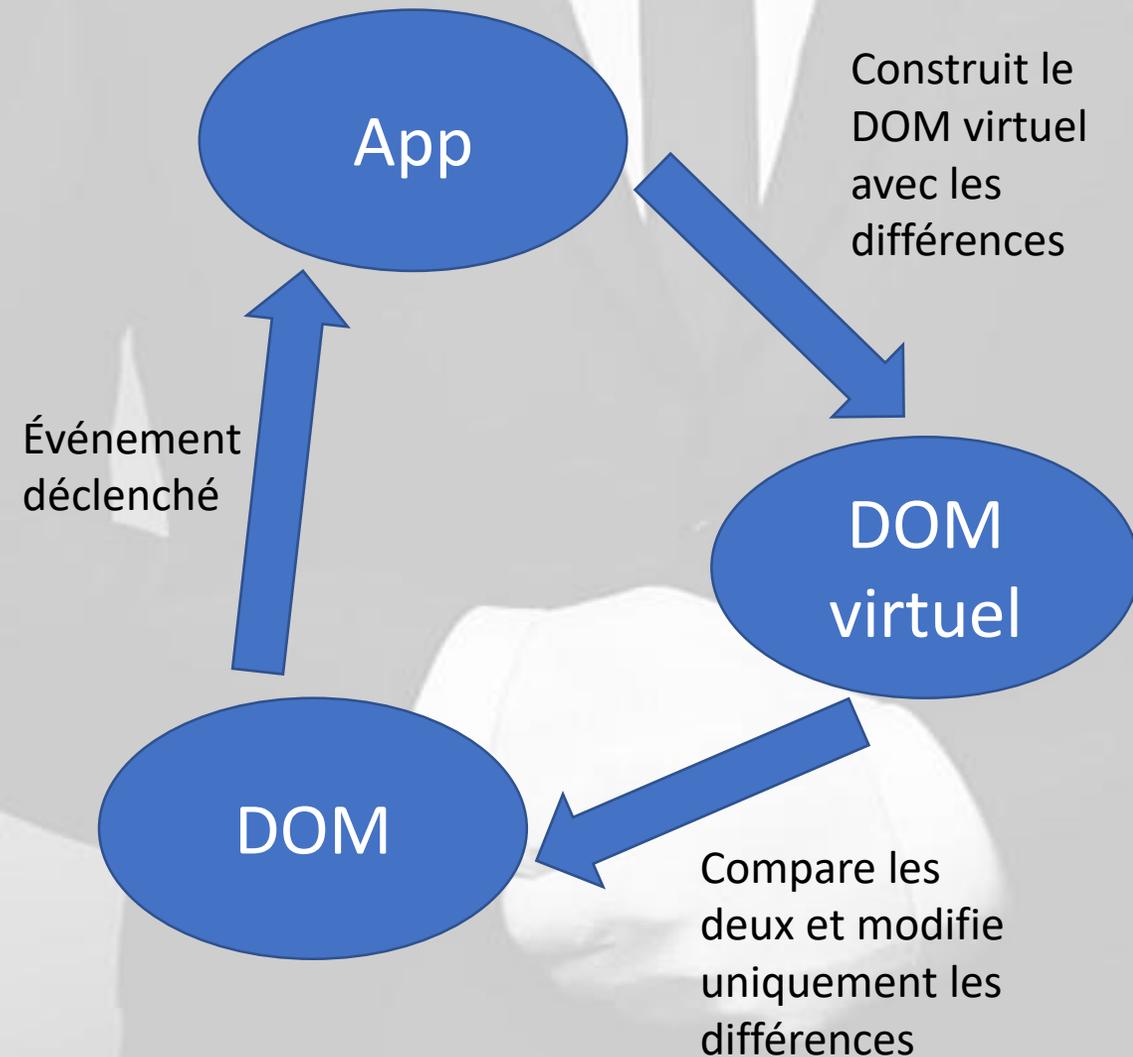
Render

Export

```
import React, { Component } from 'react';

class App extends Component {
  constructor(props) {
    super(props)
    this.state = {
      contenu: 'coucou!'
    }
    this.handleChange = this.handleChange.bind(this)
  }
  handleChange(e) {
    this.setState({
      contenu: e.target.value
    })
  }
  render() {
    return (
      <div>
        <p>Contenu : </p>
        <p>{this.state.contenu}</p>
        <input
          type="text"
          onChange={this.handleChange}
        />
      </div>
    );
  }
}

export default App;
```





TP - Calculatrice

```
class Scientifique extends Component{
  constructor(props){
    super(props)
    this.state={
      ecran: ''
    }
    this.handleResultat=this.handleResultat.bind(this)
    this.handleClick=this.handleClick.bind(this)
    this.handleSuppr=this.handleSuppr.bind(this)
    this.handleReset=this.handleReset.bind(this)
  }
  handleResultat(e){
    this.setState({
      ecran:eval(this.state.ecran).toString()
    })
  }
  handleClick(e){
    this.setState({
      ecran: this.state.ecran+e.target.value,
    })
  }
  handleSuppr(){
    this.setState({
      ecran: this.state.ecran.substr(0,this.state.ecran.length-1)
    })
  }
  handleReset(){
    this.setState({
      ecran: '',
    })
  }
}
```

```
render(){
  return(
    <div>
      <Ecran afficher={this.state.ecran}/>
      <div>
        <Bouton contenu="(" click={this.handleClick}/>
        <Bouton contenu=")" click={this.handleClick}/>
        <Bouton contenu="/" click={this.handleClick}/>
        <Bouton contenu="C" click={this.handleSuppr}/>
      </div>
      <div>
        <Bouton contenu="1" click={this.handleClick}/>
        <Bouton contenu="2" click={this.handleClick}/>
        <Bouton contenu="3" click={this.handleClick}/>
        <Bouton contenu="AC" click={this.handleReset}/>
      </div>
      <div>
        <Bouton contenu="4" click={this.handleClick}/>
        <Bouton contenu="5" click={this.handleClick}/>
        <Bouton contenu="6" click={this.handleClick}/>
        <Bouton contenu="+" click={this.handleClick}/>
      </div>
      <div>
        <Bouton contenu="7" click={this.handleClick}/>
        <Bouton contenu="8" click={this.handleClick}/>
        <Bouton contenu="9" click={this.handleClick}/>
        <Bouton contenu="-" click={this.handleClick}/>
      </div>
      <Bouton contenu="0" click={this.handleClick}/>
      <Bouton contenu="." click={this.handleClick}/>
      <Bouton contenu="=" click={this.handleResultat}/>
      <Bouton contenu="*" click={this.handleClick}/>
    </div>
  )
}
```



TP - Calculatrice

```
class Bouton extends Component {
  render(){
    return(
      <button className="boutonCalc" onClick={this.props.click} value={this.props.contenu}>{this.props.contenu}</button>
    );
  }
}
```

```
class Ecran extends Component{

  render(){
    return(
      <p className="ecran">{this.props.afficher}</p>
    );
  }
}
```

Remonter les States



- Si plusieurs composants dépendent d'un même State, il faut le déclarer dans leur parent commun le plus proche

Liste

- Fonction map()
- Key unique pour chaque élément

```
{props.chaine.split(' ').map((mot, index) => {  
    return <tr key={index}>  
        <td>{mot}</td>  
        <td>{mot.length}</td>  
    </tr>  
}}}
```

Fonction fléchée

- `() => {...}`
- Syntaxe raccourcie
- Ne possède pas son propre `this`

```
{table.map((x)=>{return <p>{x*2}</p>})}
```

```
const table=[1,2,3].map((x)=>{return x*2});
```

Rendu conditionnel

- Fonction qui return différents contenus
- Conditions dans le render()

```
{this.state.calculette=== 'true' && <div id="appli"><Calculette/></div>}
```

```
if(props.chaine.length>0){  
  return (  
    <div>
```

Cycle de vie des composants

- `componentWillMount()` -> exécuté avant le rendu client et serveur
- `componentDidMount()` -> exécuté après le rendu
- `componentWillUnmount()` -> exécuté après que le composant soit sorti du DOM
- ...

```
componentDidMount() {  
  alert('ola');  
}  
  
componentWillUnmount() {  
  alert('Bye');  
}
```

TP

- Objectif: Créer un compte à rebours
- Utiliser Date -> accès à la date actuel et d'autres fonctions utiles
- setInterval() -> permet d'actualiser chaque seconde

Résultat

Compte à rebours jusqu'au 2018-06-05

06 jours 13 heures 41 minutes 41 secondes