

State et Props

C'est quoi le state ?

- Objet, où l'on va pouvoir stocker les données relatives à un composant donné.
- Accessible uniquement à travers son composant.
- Lorsque l'objet d'état change, le composant effectue un nouveau rendu.

Création du state dans le composant de type classe

this.state = { définition des données sous forme de (key : value) }

```
// On va créer notre state dans la méthode constructor() de notre composant
constructor(props) {
  super(props); // Nous permet d'utiliser les méthodes de React.Component
  this.state = {
    // On stocke ici notre data
    monTexte: "Bienvenue à cette formation React."
  };
}
```

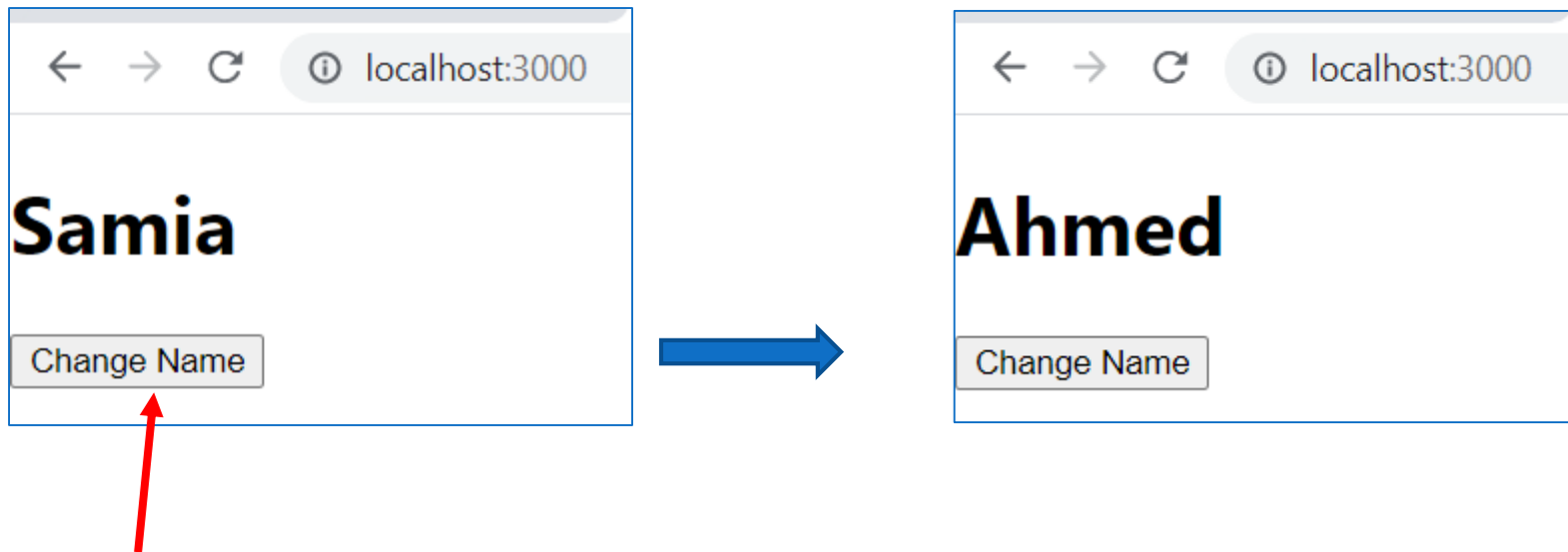
Utilisation du state

- L'accès aux données stockées dans le state se fait à travers : **this.state.nomAttribut**
- Pour l'affichage, on utilise la notation **{ }** (on parle d'interpolation)

```
class App extends Component {  
  // On va créer notre state dans la méthode constructor() de notre composant  
  constructor(props) {  
    super(props); // Nous permet d'utiliser les méthodes de React.Component  
    this.state = {  
      // On stocke ici notre data  
      monTexte: "Bienvenue à cette formation React."  
    };  
  }  
  render() {  
    return (<h2>{this.state.monTexte}</h2>);  
  }  
}  
export default App;
```

Modification du state

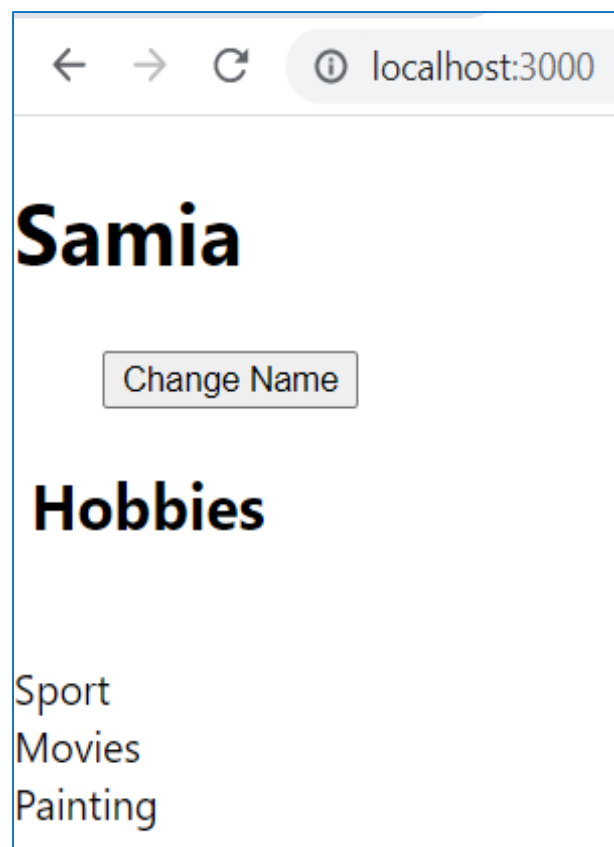
- Se fait via la méthode **this.setState** à laquelle on passera un objet avec l'attribut à modifier et la nouvelle valeur à appliquer.



```
import React, { Component } from "react";
class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "Samia",
    };
  }
  render() {
    return (
      <div>
        <h1>{this.state.name}</h1>
        <button onClick={() => this.setState({ name: "Ahmed" })}>
          Change Name
        </button>
      </div>
    );
  }
}
export default App;
```

State d'un tableau

```
import React, { Component } from "react";
class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "Samia",
      hobbies:["Sport", "Movies", "Painting"]
    };
  }
  render() {
    return (
      <div>
        <h1>{this.state.name}</h1>
        <button onClick={() => this.setState({ name: "Ahmed" })}> Change Name </button>
        <h2> Hobbies </h2>
        {this.state.hobbies.map((value,index)=>{
          return <div key = {index}>{value}</div>
        }) }
      </div>
    );
  }
}
export default App;
```



Modification du state d'un tableau

```
import React, { Component } from "react";
class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "Samia",
      hobbies: ["Sport", "Movies", "Painting"]
    };
  }
  changeHobbies={() =>{
    this.setState({ hobbies: [...this.state.hobbies, "Games"] })
  }}
  render() {
    return (
      <div> <h1>{this.state.name}</h1>
        <button onClick={() => this.setState({ name: "Ahmed" })}> Change Name </button>
        <h2> Hobbies </h2>
        {this.state.hobbies.map((value, index) =>{
          return <div key = {index}>{value}</div>
        }) }
        <button onClick={() => this.changeHobbies()}> Change Hobbies</button>
      </div>
    );
  }
}
export default App;
```

← → ↻ ⓘ localhost:3000

Samia

Change Name

Hobbies

Sport
Movies
Painting

Change Hobbies



← → ↻ ⓘ localhost:3000

Samia

Change Name

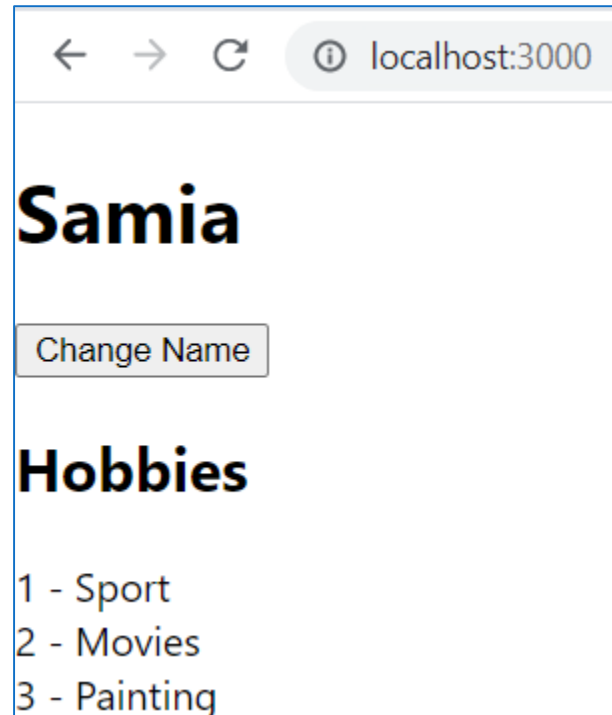
Hobbies

Sport
Movies
Painting
Games

Change Hobbies

State d'un tableau d'objets

```
import React, { Component } from "react";
class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "Samia",
      hobbies:[{"id":1,"type":"Sport"}, {"id":2,"type":"Movies"}, {"id":3,"type":"Painting"}]
    };
  }
  render() {
    return (
      <div>
        <h1>{this.state.name}</h1>
        <button onClick={() => this.setState({ name: "Ahmed" })}>
          Change Name </button>
        <h2> Hobbies </h2>
        {this.state.hobbies.map((value,index)=>{
          return <div key = {index}>{value.id} - {value.type}</div>
        }) }
      </div>
    );
  }
}
export default App;
```



Modification du state d'un tableau d'objets

```
import React, { Component } from "react";
class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "Samia",
      hobbies:[{"id":1,"type":"Sport"}, {"id":2,"type":"Movies"}, {"id":3,"type":"Painting"}]
    };
  }
  changeHobbies={() => {
    this.setState({ hobbies: [...this.state.hobbies, {"id":4,"type":"Games"}] })
  }}
  render() {
    return (
      <div>
        <h1>{this.state.name}</h1>
        <button onClick={() => this.setState({ name: "Ahmed" })}> Change Name </button>
        <h2> Hobbies </h2>
        {this.state.hobbies.map((value, index) => {
          return <div key = {index}>{value.id} - {value.type}</div>
        }) }
        <button onClick={() => this.changeHobbies()}> Change Hobbies</button>
      </div>
    );
  }
}
export default App;
```

← → ↻ ⓘ localhost:3000

Samia

Change Name

Hobbies

1 - Sport
2 - Movies
3 - Painting

Change Hobbies



← → ↻ ⓘ localhost:3000

Samia

Change Name

Hobbies

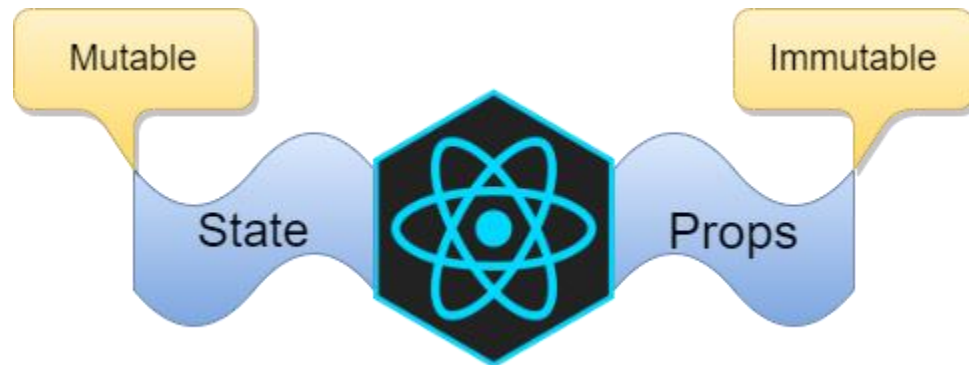
1 - Sport
2 - Movies
3 - Painting
4 - Games

Change Hobbies

C'est quoi props

- Abréviation de " properties " pour contenir les informations relatives à un composant.
- Technique qui consiste à partager du code entre des composants React.
- une props est une propriété que l'on passe à un composant. C'est donc une information qui vient de "l'extérieur" du composant, généralement, de son parent.
- Lorsque React voit un élément représentant un composant défini par l'utilisateur, il transmet les attributs JSX à ce composant en tant qu'objet unique qui est le props.
- props sont des propriétés immuables : on ne peut pas affecter des valeurs ou des objets à " props " (chose qu'on peut faire par contre au niveau de " state ").

State vs. props



state

created within component

`getInitialState`

`this.state` to read

`this.setState()` to update

props

passed in from parent

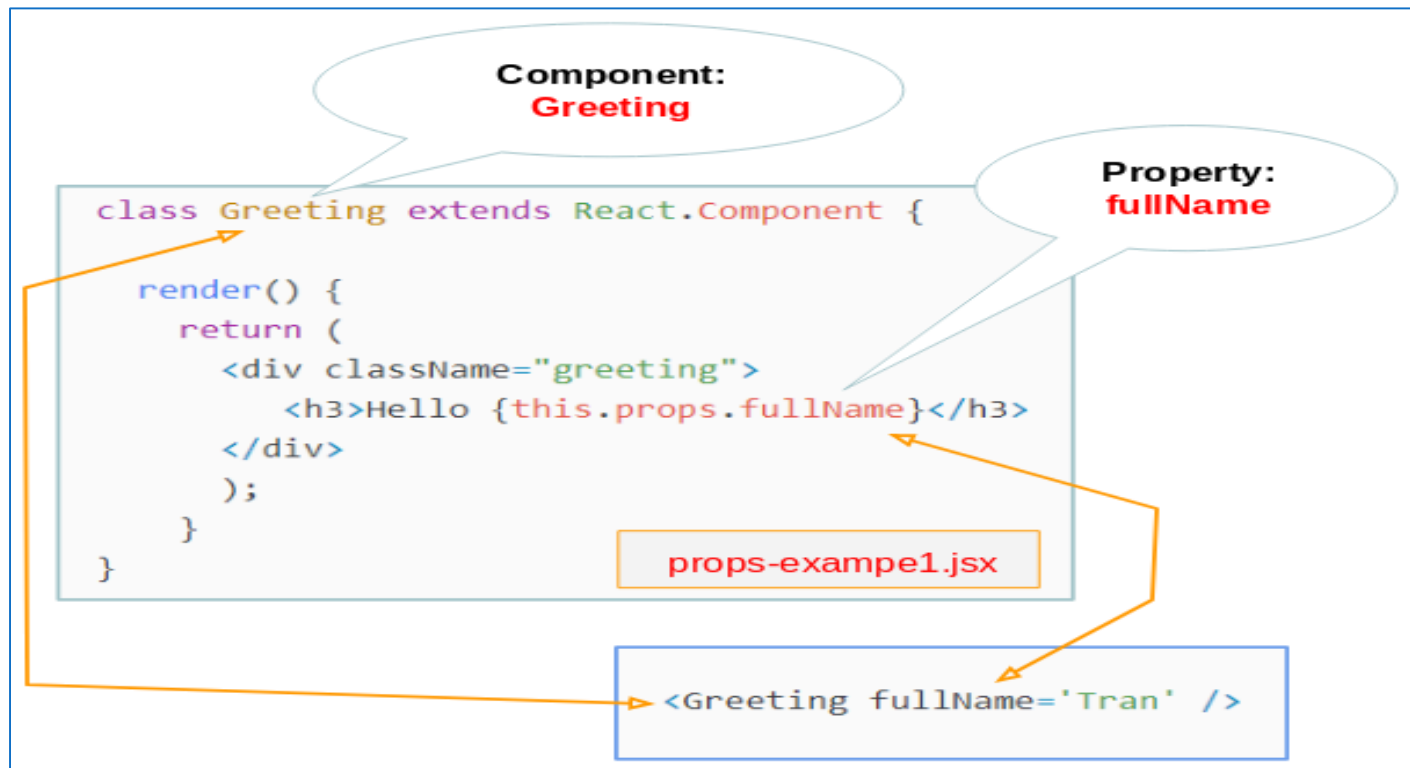
`<MyComp foo="bar" />`

`this.props` read-only within

can be defaulted & validated

Props dans un composant de type classe

- A la base, props est un objet. Il stocke les valeurs des attributs (Attributes) d'une étiquette (Tag) :
 - ▣ arguments transmis aux composants React via des attributs HTML
 - ▣ arguments transmis par le parent et accessibles via : **this.props**



Parent

□ Solution 1 :

```
import React, { Component } from "react";
import Personne from "./Personne";
class App extends Component {
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <>
        <Personne name="Mohamed"/>
      </>
    );
  }
}
export default App;
```

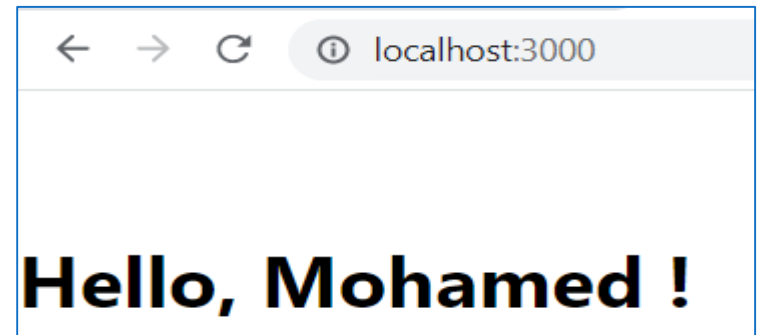
□ Solution 2 :

```
import React, { Component } from "react";
import Personne from "./Personne";
class App extends Component {
  constructor(props) {
    super(props);
  }
  render() {
    const myName="Mohamed";
    return (
      <>
        <Personne name={myName}/>
      </>
    );
  }
}
export default App;
```

Child

```
import React from 'react';
class Personne extends React.Component {
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <div>
        <h1> Hello, {this.props.name} ! </h1>

        </div>
      );
    }
  }
export default Personne;
```



Remarque :

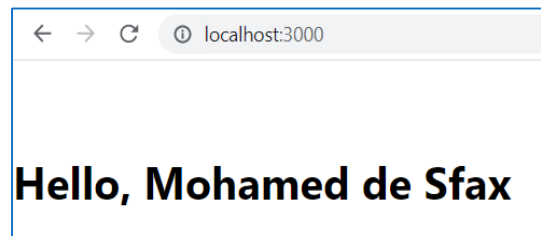
Il est recommandable de toujours appeler `super(props)`, même si ce n'est pas strictement nécessaire car ça garantit que `this.props` est défini même avant la fin du constructeur.

Multiple props

```
import React, { Component } from "react";
import Personne from "./Personne";
class App extends Component {
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <>
        <Personne name="Mohamed" ville="Sfax" />

      </>
    );
  }
}
export default App;
```

```
import React from 'react';
class Personne extends React.Component {
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <div>
        <h1> Hello, {this.props.name}
de {this.props.ville} </h1>
      </div>
    );
  }
}
export default Personne;
```



Du composant child vers le composant parent 1 / 2

```
import React from "react";
import Content from "../Content";
class App extends React.Component {
  constructor(props){
    super(props);
    this.state = {
      data: null
    }
  }
  handleCallback = (childData) =>{
    this.setState({data: childData})
  }
  render(){
    const {data} = this.state;
    return(
      <div> <Content parentCallback = {this.handleCallback}/>
        {data}
      </div>
    )
  }
  export default App;
```

Du composant child vers le composant parent 2/2

```
import React from "react";
class Content extends React.Component{
  constructor(props){
    super(props);
  }
  render(){
    return(
      <div>
        <button onClick={() => this.props.parentCallback("Data from child"
)}}>
          CLICK
        </button>
      </div>
    )
  }
}
export default Content;
```



Props dans un composant fonctionnel

```
import Personne from "./Personne";
function App() {
  const myName="Mohamed";
  return (
    <>
      <Personne name={myName}/>
    </>
  )
}
export default App;
```

```
function Personne(props){
  return (
    <h1> Hello, {props.name} </h1>
  )
}
export default Personne;
```