

Atelier 02.02 Propos State super

Exercice 1 : Afficher et modifier un texte

Objectif : Comprendre l'état (state) et la modification de texte via un bouton.

Instructions :

1. Crée un composant TextChanger.
2. Initialise un état text avec la valeur "Bonjour !".
3. Affiche ce texte dans un <p>.
4. Ajoute un bouton "Changer le texte" qui modifie le texte à "Salut React 19 !" lorsqu'on clique dessus.

Bonjour ! Salut React 19 !

```
// src/components/TextChanger.jsx

import { useState } from "react";

export default function TextChanger() {
  const [text, setText] = useState("Bonjour !");
  const handleClick = () => setText("Salut React 19 !");
  return (
    <div>
      <p>{text}</p>
      <button onClick={handleClick}>Changer le texte</button>
    </div>
  );
}
```

Exercice 2 : Changer une image

Objectif : Comprendre l'utilisation de state pour modifier une image à l'écran.

Instructions détaillées :

1. Crée un composant ImageSwitcher.
2. Créer 3 variables image1, image2 et image3 avec des images aléatoires :
<https://loremflickr.com/380/320/>
3. Regroupe ces images dans un tableau images
4. Initialise un état image avec la première image.
5. Affiche l'image avec .
6. Ajoute un bouton "Changer l'image" qui change l'état pour afficher une deuxième image.



[Changer l'image](#)

```
// FirstApp/app/src/components/ImageSwitcher.jsx

import { useState } from "react";

export default function ImageSwitcher() {
  const image1 = "https://loremflickr.com/380/320/tunisia";
  const image2 = "https://loremflickr.com/380/320/greece";
  const image3 = "https://loremflickr.com/380/320/egypt";

  const images = [image1, image2, image3];

  const [currentIndex, setCurrentIndex] = useState(0);

  const handleClick = () => {
    setCurrentIndex((prevIndex) => (prevIndex + 1) % images.length);
  };

  return (
    <div>
      <img src={images[currentIndex]} />
      <p>Image {currentIndex + 1} sur {images.length}</p>
      <button onClick={handleClick}>Changer l'image</button>
    </div>
  );
}
```

Exercice 3 : Compteur simple

Objectif : Manipuler un compteur via state.

Instructions détaillées :

1. Crée un composant **Counter**.
2. Initialise un état count à 0.
3. Affiche le compteur et deux boutons "+" et "-" pour l'incrémenter et le décrémenter.

Compteur : 5



```
// src/components/Counter.jsx
import { useState } from "react";
```

```
export default function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Compteur : {count}</p>
      <button onClick={() => setCount(count + 1)}>+</button>
      <button onClick={() => setCount(count - 1)}>-</button>
    </div>
  );
}
```

Autre version

```
// FirstApp/app/src/components/Counter.jsx
import { useState } from "react";
export default function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Compteur : {count}</p>
      <button onClick={()=>setCount((prev)=>prev+1)}>+</button>
      <button onClick={()=>setCount((prev)=>prev-1)}>-</button>
    </div>
  );
}
```

Exercice 4 : Timer avec intervalle

Objectif : Comprendre les effets avec setInterval et state.

Instructions détaillées :

1. Crée un composant Timer.
2. Initialise un état seconds à 0.
3. À l'aide de setInterval dans un useEffect, incrémente seconds chaque seconde.
4. Affiche le temps écoulé.
5. Ajoute un bouton "Réinitialiser" pour remettre le compteur à 0.

Temps écoulé : 124 secondes

```
//src/components/Timer.jsx
import { useState, useEffect } from "react";

export default function Timer(){
  const [seconds, setSeconds] = useState(0);
  useEffect(() => {
    const interval = setInterval(() => {
      setSeconds((prevSeconds) => prevSeconds + 1);
    }, 1000);
  }, []);
}

return (
```

```

<div>
  <h1>Timer: {seconds} seconds</h1>
  <button onClick={() => setSeconds(0)}>Reset</button>
</div>
);
}

```

Exercice 5 : Affichage dynamique avec props et state

Objectif : Combiner **props** et **state** pour afficher des données dynamiques.

Instructions détaillées :

1. Crée un composant ProductCard.
2. Reçoit les props **name** et **price**.
3. Initialise un état quantity à 0.
4. Affiche le nom, le prix et la quantité achetée.
5. Ajoute un bouton "Acheter" qui incrémente quantity.

Correction

```

//src/components/ProductCard.jsx
import { useState } from 'react';
export default function ProductCard(props) {

  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Product Name: {props.name}</p>
      <p>Price: {props.price} TND </p>
      <p>Quantity: {count} </p>
      <button onClick={() => setCount(count + 1)}>Acheter</button>
      <hr />
    </div>
  );
}

```

Exemple d'appel dans main.jsx

```

// Fichier : src/main.jsx
import React from "react";
import ReactDOM from "react-dom/client";
import ProductCard from "./components/ProductCard.jsx";
import "./index.css";
ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <ProductCard name={"PC"} price ={1500} />
    <ProductCard name={"Clavier"} price ={15} />
  </React.StrictMode>
);

```

Product Name: PC

Price: 1500 TND

Quantity: 1

Product Name: Clavier

Price: 15 TND

Quantity: 4

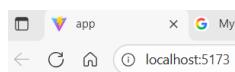
Exercice 6 : Texte changeant automatiquement

Objectif : Appliquer state avec setInterval pour faire défiler des textes.

Instructions détaillées :

1. Crée un composant RotatingText.
2. Initialise un état index à 0 et un tableau de textes ["React", "JavaScript", "CSS"].
3. Utilise setInterval pour changer l'index toutes les 2 secondes.
4. Affiche le texte courant.

résultat



My skills

Angular

Correction

```
//src/components/RotatingText.jsx
import { useEffect, useState } from "react";

export default function RotatingText() {

  const messages = ["React", "Vue", "Angular", "Svelte"]
  const [texts, setTexts] = useState(0);
  useEffect(() => {
    const interval = setInterval(() => {
      setTexts((prevTexts) => (prevTexts + 1) % messages.length);
    }, 2000);
    /* return () => clearInterval(interval); */
  }, []);
  return (
    <div>
      <h1>My skills</h1>
      <h2>{messages[texts]}</h2>
    </div>
  )
}
```

Exercice 7 On peut combiner tous les composants dans un **dashboard.jsx** pour créer un mini-dashboard interactif.

Mini Dashboard React 19

1. Changer le texte

Salut React 19 !

2. Changer l'image



Image 2 sur 3

3. Compteur

Compteur : 6

4. Timer

Timer: 166 seconds

5. Product Card

Product Name: React Livre

Price: 30 TND

Quantity: 2

6. Texte rotatif

My skills

Angular

```
// Fichier : src/components/Dashboard.jsx
import TextChanger from "./TextChanger.jsx";
import ImageSwitcher from "./ImageSwitcher.jsx";
import Counter from "./Counter.jsx";
import Timer from "./Timer.jsx";
import ProductCard from "./ProductCard.jsx";
import RotatingText from "./RotatingText.jsx";

export default function Dashboard() {
  return (
    <div style={{ padding: "20px", maxWidth: "900px", margin: "auto", fontFamily: "Arial, sans-serif" }}>
      <h1 style={{ textAlign: "center", marginBottom: "30px", color: "#333" }}>Mini Dashboard React 19</h1>

      <section style={{ marginBottom: "30px", padding: "15px", border: "1px solid #ccc", borderRadius: "8px" }}>
        <h2>1. Changer le texte</h2>
        <TextChanger />
      </section>

      <section style={{ marginBottom: "30px", padding: "15px", border: "1px solid #ccc", borderRadius: "8px" }}>
        <h2>2. Changer l'image</h2>
        <ImageSwitcher />
      </section>
    </div>
  )
}
```

```
</section>

<section style={{ marginBottom: "30px", padding: "15px", border: "1px solid #ccc",
borderRadius: "8px" }}>
  <h2>3. Compteur</h2>
  <Counter />
</section>

<section style={{ marginBottom: "30px", padding: "15px", border: "1px solid #ccc",
borderRadius: "8px" }}>
  <h2>4. Timer</h2>
  <Timer />
</section>

<section style={{ marginBottom: "30px", padding: "15px", border: "1px solid #ccc",
borderRadius: "8px" }}>
  <h2>5. Product Card</h2>
  <ProductCard name="React Livre" price={30} />
</section>

<section style={{ marginBottom: "30px", padding: "15px", border: "1px solid #ccc",
borderRadius: "8px" }}>
  <h2>6. Texte rotatif</h2>
  <RotatingText />
</section>
</div>
);
}
```