

Atelier 02.01 JSX composants

Exercice 1 : Afficher du texte avec JSX

Créer un composant **Welcome** pour afficher le message « Bienvenue sur mon site React 19! »

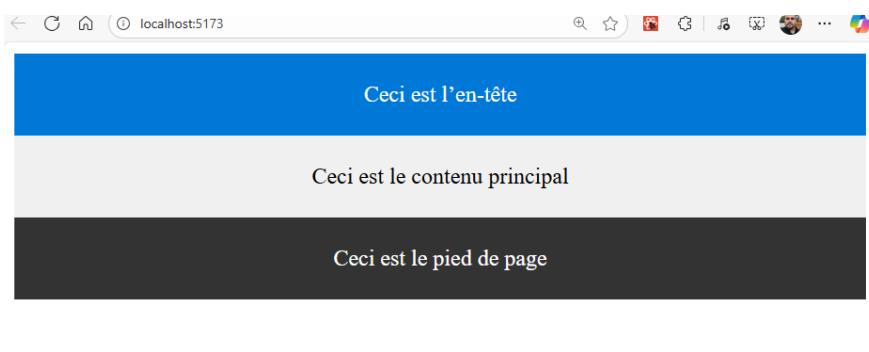
```
// Fichier : src/components/Welcome.jsx
function Welcome() {
  const message = "Bienvenue sur mon site React 19!";
  return <h1>{message}</h1>;
}
export default Welcome;
```

```
// Fichier : src/main.jsx
import React from "react";
import ReactDOM from "react-dom/client";
import Welcome from "./components/Welcome.jsx";
import "./index.css";
ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <Welcome />
  </React.StrictMode>
);
```

Exercice 2 : Créer plusieurs composants

- Créer plusieurs composants fonctionnels dans un projet React.
- Organiser les fichiers dans un dossier `components`.
- Importer et utiliser plusieurs composants dans le fichier principal `main.jsx`.

Résultat final



Étape 1 – Préparer le projet

1. Ouvrez votre projet React créé avec Vite.
Exemple :

```
cd C:\ReactProjects\MyFirstApp\app
```

2. Ouvrez le dossier dans **Visual Studio Code**.
3. Dans le dossier `src`, créez un dossier nommé **components**.

Structure attendue :

```
src/
  components/
    main.jsx
    index.css
```

Résultat attendu :

Un dossier `components` prêt à contenir vos futurs fichiers.

Étape 2 – Créer le composant Header

1. Dans le dossier `src/components`, créez un fichier **Header.jsx**.
2. Dans ce fichier, créez un composant fonctionnel `Header`.
Ce composant doit afficher un texte dans un en-tête.

Résultat attendu à l'écran (une fois utilisé dans main.jsx) :

Ceci est l'en-tête

Étape 3 – Créer le composant Main

1. Dans le même dossier `src/components`, créez un fichier **Main.jsx**.
2. Créez un composant fonctionnel `Main` qui affiche le contenu principal de la page.

Résultat attendu à l'écran :

Ceci est le contenu principal

Étape 4 – Créer le composant Footer

1. Dans le dossier `src/components`, créez un fichier **Footer.jsx**.
2. Créez un composant fonctionnel `Footer` qui affiche le pied de page.

Résultat attendu à l'écran :

Ceci est le pied de page

Étape 5 – Importer et afficher les composants

1. Ouvrez le fichier `src/main.jsx`.
2. Importez les trois composants : `Header`, `Main`, et `Footer`.
3. Affichez-les à la suite dans le rendu principal de votre application.

Résultat attendu à l'écran :

Ceci est l'en-tête
Ceci est le contenu principal

Ceci est le pied de page

Les trois lignes doivent s'afficher les unes sous les autres.

Étape 6 – Ajouter un peu de style (facultatif)

1. Ouvrez le fichier **src/index.css**.
2. Ajoutez quelques styles pour différencier les zones :
 - une couleur de fond bleue pour le header,
 - un fond gris pour le main,
 - un fond noir pour le footer.
3. Centrez le texte et ajoutez un peu d'espace autour.

Résultat attendu visuel :

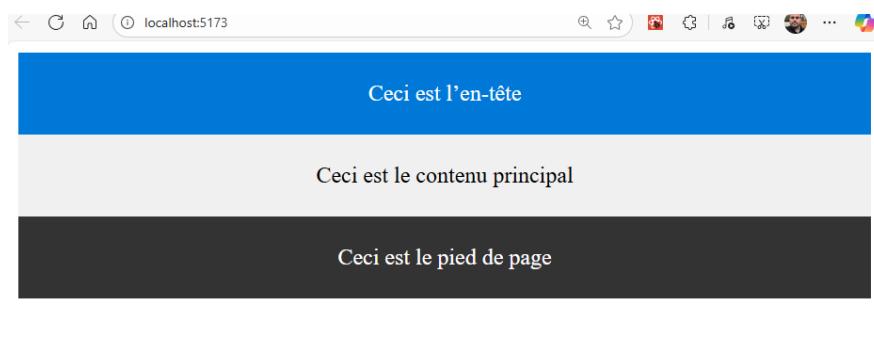
- En-tête : fond bleu, texte blanc.
- Contenu principal : fond gris clair.
- Pied de page : fond noir, texte blanc.

Affichage attendu :

```
[ Zone bleue ] Ceci est l'en-tête
[ Zone grise ] Ceci est le contenu principal
[ Zone noire ] Ceci est le pied de page
```

Étape 7 – Tester votre application

1. Dans le terminal, exécutez la commande : **npm run dev**
2. Ouvrez votre navigateur à l'adresse indiquée (par défaut : <http://localhost:5173>).
3. Vérifiez que les trois sections s'affichent correctement.

**Correction**

```
// Fichier : src/components/Header.jsx
export default function Header() {
  return <header>Ceci est l'en-tête</header>;
}
```

```
// Fichier : src/components/Main.jsx
export default function Main() {
```

```
    return <main>Ceci est le contenu principal</main>;
}
```

```
// Fichier : src/components/Footer.jsx
export default function Footer() {
  return <footer>Ceci est le pied de page</footer>;
}
```

```
// Fichier : src/main.jsx
import React from "react";
import ReactDOM from "react-dom/client";
import Header from "./components/Header.jsx";
import Main from "./components/Main.jsx";
import Footer from "./components/Footer.jsx";
import "./index.css";
ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <Header />
    <Main />
    <Footer />
  </React.StrictMode>
);
```

```
/* index.css */

header, main, footer {
  text-align: center;
  padding: 20px;
}

header {
  background-color: #0078d7;
  color: white;
}

main {
  background-color: #f0f0f0;
}

footer {
  background-color: #333;
  color: white;
}
```

Exercice 3 : Réutiliser des composants

- Réutiliser plusieurs composants déjà créés.
- Composer une page complète à partir de plusieurs composants React.
- Simplifier le fichier principal (main.jsx) en regroupant les composants.

Contexte

Dans l'exercice précédent, vous avez créé trois composants : **Header**, **Main** et **Footer**.

Vous allez maintenant les regrouper dans un seul composant “**Page**” pour rendre le code plus propre et plus organisé.

Étape 1 – Créer le composant Page

1. Dans le dossier src/components, créez un nouveau fichier nommé **Page.jsx**.
2. Ce composant doit **importer** les trois composants précédents :
 - o Header.jsx
 - o Main.jsx
 - o Footer.jsx
3. Le composant Page doit **afficher** ces trois éléments les uns sous les autres.

Structure du composant :

- Une balise principale <div> qui contient les trois composants.

Résultat attendu à l'écran (quand Page sera utilisé dans main.jsx) :

Ceci est l'en-tête

Ceci est le contenu principal

Ceci est le pied de page

Étape 2 – Modifier le fichier principal main.jsx

1. Ouvrez le fichier **src/main.jsx**.
2. Supprimez les imports de Header, Main et Footer.
3. Importez uniquement le composant **Page**.
4. Dans le rendu (ReactDOM.createRoot(...).render()), affichez le composant <Page />.

Résultat attendu à l'écran :

- L'affichage doit rester identique à celui de l'exercice précédent :
- Ceci est l'en-tête
- Ceci est le contenu principal

- Ceci est le pied de page
- Mais cette fois, tout est rendu à partir d'un seul composant : Page.

Étape 3 – Vérifier le rendu global

1. Lancez le projet dans le terminal :
2. npm run dev
3. Ouvrez votre navigateur à l'adresse indiquée (souvent http://localhost:5173).

Résultat attendu visuel :

- Une page avec les trois zones :
 - L'en-tête (header)
 - Le contenu principal (main)
 - Le pied de page (footer)

L'apparence reste la même que dans l'exercice 2.

Correction

```
// Fichier : src/components/Page.jsx
import Header from "./Header.jsx";
import Main from "./Main.jsx";
import Footer from "./Footer.jsx";

export default function Page() {
  return (
    <div>
      <Header />
      <Main />
      <Footer />
    </div>
  );
}
```

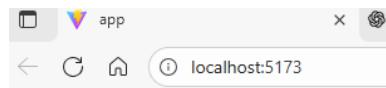
```
// Fichier : src/main.jsx
import React from "react";
import ReactDOM from "react-dom/client";
import Page from "./components/Page.jsx";
import "./index.css";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <Page />
  </React.StrictMode>
);
```

Exercice 4 : JSX avec expressions

Objectif :

Apprendre à utiliser des **expressions JavaScript** dans le code JSX (calculs, variables, conditions).

Résultat

5 + 7 = 12

REACT 19 EST GÉNIAL!

Deux est plus grand que zéro

Étape 1 — Créer le composant MathDisplay.jsx

1. Dans le dossier src/components, crée un fichier nommé **MathDisplay.jsx**.
2. Crée un composant fonctionnel nommé **MathDisplay**.
3. Dans ce composant, affiche plusieurs éléments <p> :
 - o Le premier doit afficher le résultat du calcul 5 + 7 en utilisant des **accolades JSX**.
 - o Le deuxième doit afficher une **chaîne de texte stockée dans une variable**, convertie en **majuscules**.
 - o Le troisième doit afficher un message conditionnel avec un **opérateur ternaire**.

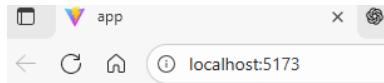
Étape 2 — Modifier src/main.jsx

1. Importe ton composant **MathDisplay** dans le fichier main.jsx.
2. Rends ce composant à l'écran dans la fonction render().

Étape 3 — Lancer le projet

1. Démarre ton serveur de développement avec :
2. npm run dev
3. Ouvre ton navigateur et observe le résultat.

Ce que l'écran doit afficher :



5 + 7 = 12

REACT 19 EST GÉNIAL!

Deux est plus grand que zéro

Correction

```
// Fichier : src/components/MathDisplay.jsx
export default function MathDisplay() {
  const message = "React 19 est génial!";
  return (
    <div>
      <p>5 + 7 = {5 + 7}</p>
      <p>{message.toUpperCase()}</p>
      <p>{2 > 0 ? "Deux est plus grand que zéro" : "Ceci n'apparaîtra jamais"}</p>
    </div>
  );
}
```

```
// Fichier : src/main.jsx
import React from "react";
import ReactDOM from "react-dom/client";
import MathDisplay from "./components/MathDisplay.jsx";
import "./index.css";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <MathDisplay />
  </React.StrictMode>
);
```

Exercice 5 : Combiner texte et emojis

```
// Fichier : src/components/Description.jsx
export default function Description() {
  const text = "Apprenez React facilement";
  const emoji = "🚀";
  return <p>{text + " " + emoji}</p>;
}
```

```
// Fichier : src/main.jsx
import React from "react";
import ReactDOM from "react-dom/client";
import Description from "./components/Description.jsx";
import "./index.css";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <Description />
  </React.StrictMode>
);
```

Exercice 6 : Mini-projet complet (Header, Main, Footer, Description)

Résultat



```
// Fichier : src/components/Main.jsx
import Description from "./Description.jsx";

export default function Main() {
  return (
    <main>
      <h2>Contenu principal</h2>
      <Description />
    </main>
  );
}
```

```
// Fichier : src/components/Page.jsx
import Header from "./Header.jsx";
import Main from "./Main.jsx";
import Footer from "./Footer.jsx";

export default function Page() {
  return (
    <div>
      <Header />
      <Main />
      <Footer />
    </div>
  );
}
```

```
// Fichier : src/main.jsx
import React from "react";
import ReactDOM from "react-dom/client";
import Page from "./components/Page.jsx";
import "./index.css";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <Page />
  </React.StrictMode>
);
```

```
/* Fichier : src/components/Page.css */

/* Reset de base */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  color: #333;
}

/* Corps de page */
body {
  background-color: #f0f4f8;
  display: flex;
  justify-content: center;
  padding: 20px;
}

/* Conteneur principal */
div {
  width: 100%;
  max-width: 800px;
  background-color: #fff;
  border-radius: 12px;
  box-shadow: 0 4px 12px rgba(0,0,0,0.1);
  overflow: hidden;
}

/* Header */
header {
  background-color: #4a90e2;
  color: #fff;
  padding: 20px;
  text-align: center;
  font-size: 1.8rem;
  font-weight: bold;
  border-bottom: 2px solid #357ab7;
}

/* Main */
main {
  padding: 30px 20px;
}

main h2 {
  margin-bottom: 20px;
  font-size: 1.5rem;
  color: #4a90e2;
}

/* Description */
p {
  font-size: 1.1rem;
  line-height: 1.6;
  color: #555;
}

/* Footer */
footer {
  background-color: #4a90e2;
  color: #fff;
  text-align: center;
  padding: 15px;
  font-size: 1rem;
  border-top: 2px solid #357ab7;
}

/* Ajouter un effet hover léger sur main */
main:hover {
```

```
background-color: #f7f9fc;  
transition: 0.3s ease-in-out;  
}
```