

# Exercice 2 - Header et Footer

---

## Objectif

Créer plusieurs composants et les assembler pour former une page complète.

## Concepts abordés

- Création de plusieurs composants
  - Import / Export de composants
  - Composition de composants
  - Fragment React (`<>...</>`)
- 

## Contexte

Vous construisez la structure de base d'un site web avec un header, un contenu principal et un footer.

## Énoncé

### Étape 1 : Créer le composant **Header**

Un header contenant :

- Le nom du site : "Mon Application"
- Un menu de navigation avec 3 liens : Accueil, À propos, Contact

### Étape 2 : Créer le composant **Footer**

Un footer contenant :

- Le texte "© 2025 - Mon Application"
- Votre nom

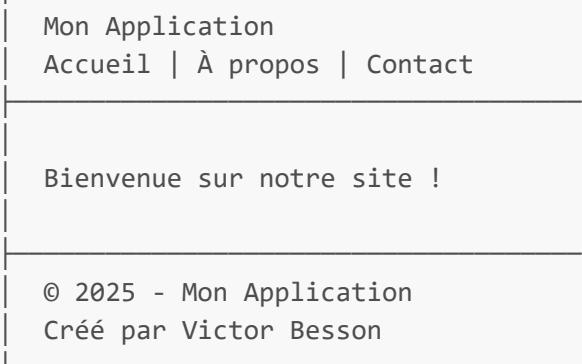
### Étape 3 : Créer le composant **Layout**

Un composant qui assemble Header, un contenu principal, et Footer.

## Structure de fichiers

```
ex02-header-footer/
  Header.jsx
  Footer.jsx
  Layout.jsx
```

## Résultat attendu



## Indices

- ▶ Indice 1 : Export d'un composant

```
// Header.jsx
function Header() {
  return <header>...</header>;
}

export default Header;
```

- ▶ Indice 2 : Import d'un composant

```
// Layout.jsx
import Header from './Header';
import Footer from './Footer';
```

- ▶ Indice 3 : Utiliser un composant

```
function Layout() {
  return (
    <div>
      <Header />      /* Comme une balise HTML */
      <main>...</main>
      <Footer />
    </div>
  );
}
```

## Points d'attention

## Le nom du fichier et du composant

Par convention, ils doivent correspondre :

- Fichier : `Header.jsx`
- Composant : `function Header()`
- Import : `import Header from './Header'`

## Balises sémantiques HTML

Utilisez les bonnes balises :

- `<header>` pour le header
- `<nav>` pour la navigation
- `<main>` pour le contenu principal
- `<footer>` pour le footer

## Le chemin d'import

```
// Si le fichier est dans le même dossier
import Header from './Header';

// Si le fichier est dans un sous-dossier
import Header from './components/Header';

// Si le fichier est dans un dossier parent
import Header from '../Header';
```

---

## Correction

### Header.jsx

```
// src/exercices/ex02-header-footer/Header.jsx

function Header() {
    // _____
    // Le header contient le titre du site et la navigation
    // On utilise les balises sémantiques HTML5 : <header>, <nav>
    // _____
    return (
        <header className="header">
            {/* Titre du site */}
            <h1>Mon Application</h1>

            {/* Navigation */}
            <nav>
                /*
                    Pour l'instant, ce sont des liens statiques.
                
```

```
    Plus tard, avec React Router, ils deviendront fonctionnels.  
  * /}  
  <a href="#accueil">Accueil</a>  
  <span> | </span>  
  <a href="#apropos">À propos</a>  
  <span> | </span>  
  <a href="#contact">Contact</a>  
  </nav>  
</header>  
);  
}  
  
export default Header;
```

## Footer.jsx

```
// src/exercices/ex02-header-footer/Footer.jsx  
  
function Footer() {  
  // =====  
  // Année dynamique : sera toujours à jour !  
  // =====  
  const annee = new Date().getFullYear();  
  const auteur = "Victor Besson";  
  
  return (  
    <footer className="footer">  
      <p>© {annee} - Mon Application</p>  
      <p>Créé par {auteur}</p>  
    </footer>  
  );  
}  
  
export default Footer;
```

## Layout.jsx

```
// src/exercices/ex02-header-footer/Layout.jsx  
  
// =====  
// IMPORTS  
// On importe les composants depuis leurs fichiers respectifs  
// Le chemin commence par ./ car ils sont dans le même dossier  
// =====  
import Header from './Header';  
import Footer from './Footer';  
  
function Layout() {  
  // =====
```

```
// COMPOSITION DE COMPOSANTS
//
// Layout "compose" Header et Footer comme des briques Lego.
// C'est le principe fondamental de React : construire des
// interfaces en assemblant des composants réutilisables.
//
// Flux visuel :
//   Layout
//     └─ Header (importé)
//     └─ main (HTML natif)
//     └─ Footer (importé)
// =====
return (
  <div className="layout">
    {/* Le composant Header s'utilise comme une balise */}
    <Header />

    {/* Contenu principal de la page */}
    <main className="main-content">
      <h2>Bienvenue sur notre site !</h2>
      <p>
        Ceci est le contenu principal de la page.
        Il est directement dans Layout pour l'instant.
      </p>
    </main>

    {/* Le composant Footer */}
    <Footer />
  </div>
);
}

export default Layout;
```

## CSS associé (optionnel)

```
/* src/exercices/ex02-header-footer/layout.css */

.layout {
  min-height: 100vh;
  display: flex;
  flex-direction: column;
}

.header {
  background-color: #2563eb;
  color: white;
  padding: 1rem 2rem;
}

.header h1 {
```

```
margin: 0 0 0.5rem 0;
}

.header nav a {
  color: white;
  text-decoration: none;
}

.header nav a:hover {
  text-decoration: underline;
}

.main-content {
  flex: 1;
  padding: 2rem;
}

.footer {
  background-color: #1f2937;
  color: white;
  padding: 1rem 2rem;
  text-align: center;
}

.footer p {
  margin: 0.25rem 0;
}
```

---

## Schéma mental

```
Layout.jsx
```

```
import Header from './Header';
import Footer from './Footer';
```

```
<Header /> ← Fichier Header.jsx
```

```
<main> ... </main>
```

```
<Footer /> ← Fichier Footer.jsx
```

## Pour aller plus loin

1. Ajouter un composant **Logo** utilisé dans le Header
2. Utiliser un Fragment **<>** au lieu d'une div dans Layout
3. Passer le titre du site en prop au Header

```
// Avec Fragment (évite une div inutile dans le DOM)
return (
  <>
    <Header />
    <main>...</main>
    <Footer />
  </>
);
```