Atelier 02.07 Styles

Dossier : src/design

Exercice 1 – Style Inline Simple

Objectif :
Apprendre à utiliser le style inline sur un composant React simple.

Instruction :

1. Crée un fichier HomeInline.jsx dans src/design.
2. Ajoute un composant qui affiche un titre et un paragraphe avec des styles inline.

Correction / Code :

```jsx
// src/design/HomeInline.jsx
import React from 'react';

function HomeInline() {
  // style directement dans l'attribut style
  return (
    <div style={{ backgroundColor: 'lightblue', padding: '20px', borderRadius: '10px' }}>
      <h1 style={{ color: 'white', textAlign: 'center' }}>Bienvenue Home Inline</h1>
      <p style={{ fontSize: '16px', color: 'darkblue' }}>
        Ceci est un exemple de style inline en React.
      </p>
    </div>
  );
}

export default HomeInline;
```

Dans le fichier main.jsx

```jsx
import { createRoot } from 'react-dom/client'

import HomeInline from './design/HomeInline.jsx'
createRoot(document.getElementById('root')).render(
  <>
  <HomeInline />

  </>
)
```

Commentaire :

- Les propriétés CSS sont **en camelCase**.
- Utiliser style inline est pratique pour des styles **dynamiques** ou très locaux.

Exercice 2 – Utiliser un objet JavaScript pour le style

Objectif :
Réutiliser des styles en créant un objet JS séparé.

Instruction :

1. Crée un fichier Box.jsx dans src/design.
2. Définis un objet boxStyle avec plusieurs propriétés CSS.
3. Applique cet objet au conteneur .

Correction / Code :

```jsx
// src/design/Box.jsx
import React from 'react';

const boxStyle = {
  backgroundColor: 'lightgreen',
  padding: '15px',
  borderRadius: '10px',
  textAlign: 'center'
};

function Box() {
  return (
    <div style={boxStyle}>
      <h2>Box avec style objet JS</h2>
      <p>On peut réutiliser cet objet dans plusieurs composants.</p>
    </div>
  );
}

export default Box;
```

Commentaire :

- Objet JS permet de **centraliser les styles** et de les réutiliser.
- Bon pour des composants modulaires avec styles communs.

Exercice 3 – CSS Classique

Objectif :
Apprendre à créer et utiliser une feuille CSS classique.

Instruction :

1. Crée Classic.css dans src/design.
2. Ajoute des classes pour un conteneur, titre et paragraphe.
3. Crée un composant **Classic.jsx** qui importe **Classic.css** et applique les classes.

Correction / Code Composant :

```jsx
// src/design/Classic.jsx
import React from 'react';
import './Classic.css';

function Classic() {
  return (
    <div className="container">
      <h1>Style CSS Classique</h1>
      <p>Texte stylisé via une feuille CSS externe.</p>
    </div>
  );
}

export default Classic;
```

Correction / Code CSS :

```css
/* src/design/Classic.css */
.container {
  background-color: lightyellow;
  padding: 20px;
  border-radius: 8px;
  text-align: center;
}

.container h1 {
  color: darkblue;
  font-size: 24px;
}

.container p {
  color: gray;
  font-size: 16px;
}
```

Commentaire :

- CSS classique est **réutilisable** et idéal pour plusieurs composants.
- Risque de conflit de noms si on ne choisit pas des noms uniques.

Exercice 4 – Module CSS

Objectif :
Appliquer des styles locaux et éviter les conflits avec d'autres composants.

Instruction :

1. Crée Card.module.css dans src/design.
2. Crée un composant Card.jsx qui importe ce module CSS et applique les classes.

Correction / Code CSS :

```css
/* src/design/Card.module.css */
.card {
  background-color: lavender;
  padding: 20px;
  border-radius: 12px;
  text-align: center;
}

.cardTitle {
  color: purple;
  font-size: 22px;
}

.cardText {
  font-size: 16px;
  color: gray;
}
```

Correction / Code Composant :

```jsx
// src/design/Card.jsx
import React from 'react';
```

```jsx
import styles from './Card.module.css';

function Card() {
  return (
    <div className={styles.card}>
      <h2 className={styles.cardTitle}>Carte avec Module CSS</h2>
      <p className={styles.cardText}>Chaque style est scoped au composant.</p>
    </div>
  );
}

export default Card;
```

Commentaire :

- Modules CSS génèrent des classes **uniques automatiquement**.
- Parfait pour des projets React avec beaucoup de composants.

Exercice 5 – Combiner les méthodes

Objectif :
Comprendre comment combiner inline, CSS classique et module CSS.

Instruction :

1. Crée Combined.jsx dans src/design.
2. Utilise un module CSS pour le conteneur.
3. Applique un style inline pour le texte.
4. Ajoute une classe CSS classique pour le titre depuis Classic.css.

Correction / Code :

```jsx
// src/design/Combined.jsx
import React from 'react';
import styles from './Card.module.css';
import './Classic.css';

function Combined() {
  return (
    <div className={styles.card}>
      <h1 className="container">Titre CSS Classique</h1>
      <p style={{ color: 'darkred', fontWeight: 'bold' }}>
        Texte avec style inline combiné au module et CSS classique.
      </p>
    </div>
  );
}

export default Combined;
```

Commentaire :

- Combiner les méthodes permet **flexibilité et modularité**.
- À utiliser avec précaution pour ne pas complexifier le code.