

Cheat Sheet - JSX

Qu'est-ce que le JSX ?

JSX = JavaScript XML - Une syntaxe qui permet d'écrire du "HTML" dans JavaScript.

Le JSX n'est PAS du HTML ! C'est du JavaScript déguisé.

Les 5 règles d'or du JSX

1. Un seul élément parent

```
// ERREUR : deux éléments racines
return (
  <h1>Titre</h1>
  <p>Texte</p>
)

// OK : un seul parent (div)
return (
  <div>
    <h1>Titre</h1>
    <p>Texte</p>
  </div>
)

// OK : fragment (pas de div inutile dans le DOM)
return (
  <>
    <h1>Titre</h1>
    <p>Texte</p>
  </>
)
```

2. Attributs en camelCase

HTML classique	JSX
class	className
for	htmlFor
onclick	onClick
onchange	onChange
tabindex	tabIndex
maxlength	maxLength

```
// HTML
<label for="email" class="label">Email</label>
<input type="text" maxlength="50" tabindex="1" />

// JSX
<label htmlFor="email" className="label">Email</label>
<input type="text" maxLength="50" tabIndex="1" />
```

3. Les accolades {} pour le JavaScript

```

const name = "Victor";
const age = 26;
const isAdmin = true;

return (
  <div>
    {/* Variable */}
    <h1>Bonjour {name}</h1>

    {/* Calcul */}
    <p>Dans 10 ans : {age + 10} ans</p>

    {/* Condition ternaire */}
    <span>{isAdmin ? "Admin" : "User"}</span>

    {/* Appel de fonction */}
    <p>{name.toUpperCase()}</p>
  </div>
)

```

4. Balises auto-fermantes

```

// HTML :  <br> <input type="text">
// JSX : TOUJOURS fermer avec />


<br />
<input type="text" />
<hr />

```

5. Commentaires en JSX

```

return (
  <div>
    {/* Ceci est un commentaire JSX */}
    <h1>Titre</h1>

    {/*
      Commentaire
      sur plusieurs lignes
    */}
  </div>
)

```

Tableau récapitulatif : Ce qui change entre HTML et JSX

Concept	HTML	JSX
Classes CSS	class="btn"	className="btn"
Label for	for="input"	htmlFor="input"
Style inline	style="color: red"	style={{ color: 'red' }}
Événements	onclick="fn()"	onClick={fn}
Balises vides	 	
Booléens	disabled="disabled"	disabled OU disabled={true}

Style inline en JSX

```
// ATTENTION : double accolades et camelCase !  
  
// HTML  
<div style="background-color: blue; font-size: 16px;">  
  
// JSX  
<div style={{ backgroundColor: 'blue', fontSize: '16px' }}>  
  
// Avec une variable  
const monStyle = {  
  backgroundColor: 'blue',  
  fontSize: '16px',  
  padding: '1rem'  
};  
  
<div style={monStyle}>
```

Pourquoi double accolades ?

- Première paire {} : "j'injecte du JavaScript"
- Deuxième paire {} : "voici un objet JavaScript"

Piège classique : Afficher des valeurs "falsy"

```
const count = 0;  
  
// PIÈGE : affiche "0" dans le DOM !  
{count && <p>Il y a {count} éléments</p>}  
  
// SOLUTION : comparer explicitement  
{count > 0 && <p>Il y a {count} éléments</p>}  
  
// Ou avec ternaire  
{count ? <p>Il y a {count} éléments</p> : null}
```

Valeurs "falsy" en JS : false, 0, "", null, undefined, NaN

Mémo visuel

```
| JSX = JavaScript + "HTML"
|
| function MonComposant() {
|   const data = "valeur";      ← JavaScript normal
|
|   return (
|     <div>                  ← Début du JSX
|       <h1>{data}</h1>      ← {} = JS dans JSX
|     </div>                  ← Fin du JSX
|   );
| }
```