

Les composants React

Composant basé sur une fonction

Les composants React

Composant basé sur une fonction

Un composant React est une fonction qui renvoie un rendu "React" à l'aide du JSX .

```
const Welcome = function(props) {  
  return (  
    <h1>Hello {props.name}!</h1>  
  );  
};  
  
export default Welcome;
```

Utiliser un Composant dans JSX

En JSX, un Composant React est appelé sous forme de balise XML.

Les différents attributs transmis pourront être récupérés dans le composant à l'aide de son paramètre d'entrée.

```
const element = (  
  <div>  
    <Welcome name='Zaza' />  
    <p>Lorem ipsum...</p>  
    <GoodBye />  
  </div>  
)
```

⚠ Attention aux nom des éléments

→ React en **PascalCase**.

→ HTML en **lowercase**.

Le paramètre d'entrée

Les composants React

Paramètre d'entrée

Lors de l'utilisation d'un composant, celui-ci peut recevoir plusieurs attributs.

```
<DemoParametre arg1="Hello" arg2={42} />
```

Chaque composant possède un paramètre d'entrée « props », celui-ci est **immutable** et nous permet de récupérer les différentes valeurs sous la forme d'un objet JavaScript.

Les arguments sont accessibles à l'aide de « **props.argument** »

```
const DemoParametre = function(props) {  
  const {arg1, arg2} = props;  
  //...  
};
```

 Il ne faut jamais modifier l'objet « props » d'un composant React!

Paramètre d'entrée - Valeur par défaut

Il est possible de définir une valeur par défaut aux variables de « props ».

Pour cela, il faut ajouter la propriété « defaultProps » sur le composant.

```
Welcome.defaultProps = {  
  name: 'Riri'  
};
```

Paramètre d'entrée - Vérification du type

Il est également possible d'ajouter une vérification du type des variables de « props », sans avoir recours aux extensions du javascript telles que **TypeScript** ou **Flow**.

Pour cela, installer le package « prop-types » et ajouter la propriété « propTypes ».

“ npm install –save prop-types ”

```
import PropTypes from 'prop-types';  
  
// ...  
  
Welcome.propTypes = {  
  |   name: PropTypes.string  
};
```


Modifier l'apparence

Les composants React

Utilisation du CSS au sein d'un composant

Pour modifier l'apparence d'un composant, il faut lui appliquer des règles CSS.

Il existe 3 méthodes pour cela :

- L'importation de fichiers CSS.
- L'ajout de module CSS dédié au composant.
- Le CSS-in-JS : Création du style sous forme d'objet Javascript.

Ce cours n'aborde pas le « CSS-in-JS ». Pour plus d'information sur son utilisation, vous pouvez consulter le site : <https://fr.reactjs.org/docs/dom-elements.html#style>

Import de fichier CSS

Pour ajouter des fichiers « css » au composants , nous allons utiliser le mécanisme d'import de « webpack » étendu au fichier « css ».

Cela permet de lier les fichiers de style directement avec le composant.

Lors de l'exécution, les styles sont appliqués avec une balise « style » dans le header.

Il est possible d'utiliser tous les sélecteurs CSS.

 Attention, il est possible qu'une règle CSS en écrase une autre !

Exemple d'import de fichier CSS

stylized-title.css

```
.title {  
  color: ■ orangered;  
  font-size: 2rem;  
}
```

stylized-title.jsx

```
import './stylized-title.css';  
  
const StylizedTitle = function(props) {  
  const title = props.title;  
  
  return (  
    <h1 className='title'>{title}</h1>  
  );  
}  
  
export default StylizedTitle;
```

Module CSS


Pour utiliser les modules CSS, il est nécessaire de

- Respecter la convention « **[name].module.css** » pour les noms de fichier.
- Les règles CSS doivent être composées d'un sélecteur de type classe.
- D'importer le fichier à l'aide de « `import ... from './...'` »
- D'utiliser en JSX l'attribut « `className` »

L'avantage de cette méthode est que chaque nom de classe CSS sera automatiquement généré sous la forme de « `[filename]_[classname]_[hash]` », pour éviter des conflits.

Exemple d'import de Module CSS

stylized-title.module.css

```
.title {  
  color:  orangered;  
  font-size: 2rem;  
}
```

stylized-title.jsx

```
import style from './stylized-title.module.css';  
  
const StylizedTitle = function(props) {  
  const title = props.title;  
  
  return (  
    <h1 className={style.title}>{title}</h1>  
  );  
}  
  
export default StylizedTitle;
```

Exercice • Création d'un composant

Les composants React

Exercice

- Créer un composant basé sur une fonction et ajouter le dans l'App
 - Il reçoit en paramètre d'entrée un « nom » et un « âge ».
 - La valeur par défaut de l'âge est de 18.
 - Ajouter la vérification de type sur les 2 paramètres d'entrées
 - Le rendu du composant doit être composé de deux balises « paragraphe »
 - Mockup du rendu à obtenir

Bienvenue [Nom] sur l'application React !

Vous avez [Age] ans!

 Attention aux dépendances entre les différents fichiers.