

# Training React - JSX !

---

Pour se familiariser avec React et les outils de développement front-end

## Exercice 1

Le but de l'exercice est de se familiariser avec React et d'envisager les moyens disponibles de transmission d'information entre les composants. Pour cela, on va créer un composant React de l'élément « Notification » du framework CSS Bulma.

Le composant React réalisé devra permettre :

- De passer un message dans la notification
- De passer l'information de couleur
- De choisir d'afficher ou non le bouton d'auto-fermeture

L'ensemble des instructions où npm est requis peuvent se faire indifféremment avec Yarn.

### Question 1

- Créez un projet un projet CRA (create-react-app) react-training.
- Ajoutez le framework CSS Bulma à votre projet.
- Dans le fichier App.js, créez une structure html de type `containeur > colonnes > colonne`

### Question 2

Ajoutez un sous dossier « **components** » au dossier « **src** ». Dans **components**, créez un composant nommé **WrapperComponent** qui pour le moment va renvoyer le texte « Mes alertes » dans une div ayant la class « **box** ». Effectuez le rendu de ce composant dans **App**.

### Question 3

Créez un composant React nommé `NotificationComponent` capable de renvoyer un message dans la structure HTML de l'élément Bulma correspondant (on ne s'occupe pas du bouton de fermeture pour le moment). Il doit pouvoir recevoir les props suivantes :

- `content` : le contenu du message
- `type` : indique le type de notification (classes css du composant Bulma, ex : `is-danger`)

Vérifiez votre travail en effectuant le rendu de ce composant dans App avec pour paramètres :

- `Type` : `is-warning`
- `Content` : Je suis un message d'avertissement

Si le rendu est correct, **ajoutez vos modifications à git et créez une nouvelle branche « feature/branch » pour le développement de la suite.**

### Question 4

Modifier le composant `WrapperComponent` pour qu'il renvoie **2 occurrences** du même composant `NotificationComponent` avec des paramètres différents :

- Le premier est de type « `is-success` » et affiche le texte « Je suis un message de succès ».
- Le second est de type « `is-danger` » et affiche le texte « Je suis un message d'alerte ».

Vérifiez que le rendu est conforme aux modifications.

### Question 5

Ajoutez le code nécessaire à la prise en charge de propriétés par défaut au composant `NotificationComponent` :

- `content`: Votre message
- `displayCloseBtn`: `false`

Divisez le composant `NotificationComponent` en deux sous-composants (il faudra modifier ce que retourne le composant parent pour y inclure les 2 composants enfant :

- **NotificationButton** qui renvoie le code HTML nécessaire au rendu du bouton (cf. documentation Bulma) si la propriété **displayCloseBtn** est true. Sinon, on renvoie simplement **null**.
- **NotificationContent** qui renvoie le contenu de la propriété **content** de son composant parent dans un paragraphe.

Modifiez le **WrapperComponent** pour faire en sorte que les boutons de fermeture de chaque notification soient visibles.

## Question 6

*Dans cette question, nous allons faire en sorte d'implémenter la fonctionnalité de fermeture des notifications par clic sur le bouton X (au clic, on supprime l'élément). Chaque notification devra se fermer individuellement. Pour supprimer la notification, le composant enfant **NotificationButton** va devoir informer son parent **NotificationComponent** qu'il a été cliqué...*

- A l'intérieur du composant **NotificationComponent**, créez une nouvelle fonction expression **handleClick(e)** qui effectuera un **console.log(e)**. « e » est ici un événement générique.
- Passez la fonction **handleClick** en tant que propriété (**props**) (du même nom) au composant **NotificationButton** situé à l'intérieur de **NotificationComponent**.
- Dans le composant **NotificationButton**, ajoutez une propriété **onClick** qui a pour valeur la fonction **handleClick**.
- Vérifiez que le composant **WrapperComponent** prenne en charge cette nouvelle fonctionnalité : au clic sur un bouton X, la console doit vous renvoyer un objet JS.

## Question 7

Modifiez la fonction **handleClick(e)** déclarée dans le composant **NotificationComponent** de façon qu'elle supprime la notification lorsque le bouton de fermeture est cliqué (il renvoie alors l'objet « e »).