
 <p>Se former autrement HONORIS UNITED UNIVERSITIES</p>	<p>Année Universitaire : 2024-2025</p> 
<p>Atelier : State et Props</p>	

Objectifs :

- Comprendre la différence entre "state" et "props"
- Maîtriser l'utilisation de "state"
- Manipuler les "props"

Exercice 1 : Création d'un compteur

- Créez un composant Counter avec un état initial count défini à 0.
- Ajoutez deux boutons : un pour incrémenter (+1) et un pour décrémenter (-1) la valeur de count.
- Affichez la valeur actuelle de count.



Utiliser des **props** pour modifier le step d'incrémentation et décrémentation du compteur ainsi que la valeur initiale affichée , sachant que les props sont :

- `initialCount` : la valeur initiale du compteur.
- `Step` : pas d'incrémentation et décrémentation du compteur

Compteur : 10

+5

-5

Reset

Exercice 2 : Gestion d'une liste dynamique

Utiliser les props pour afficher une liste et mettre à jour son contenu via le state.
sachant que les props de ce composant :

- initialItems : tableau qui contient la liste initiale des éléments
 - placeholder : le texte à insérer dans l'input
- a) Créez un composant ListManager qui accepte une prop initialItems (tableau de chaînes de caractères, par exemple : ['React', 'Angular', 'VueJs']).
 - b) Utilisez un state items pour gérer la liste des éléments.
 - c) Ajoutez un formulaire avec un champ texte et un bouton pour ajouter de nouveaux éléments à la liste.
 - d) Affichez les éléments dans une liste non ordonnée.

Liste :

React

Supprimer

Angular

Supprimer

Vuejs

Supprimer

Entrez un nouveau élément

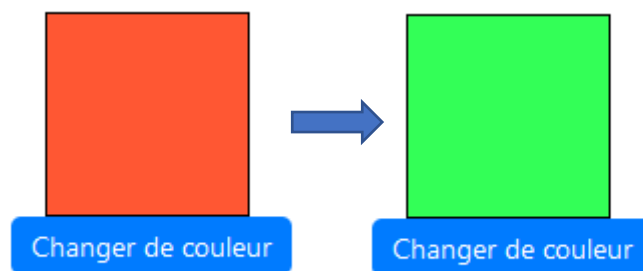
Ajouter



Exercice 3 : Changement de couleur dynamique

Utiliser les props et le state pour modifier dynamiquement un style, sachant que les props sont :

- `initialColor` le code hexadécimal de la couleur initiale
 - `colorOptions` un tableau de codes de couleurs en hexadécimal.
- a) Créez un composant `ColorBox` qui affiche une boîte carrée (div) avec une couleur de fond initiale passée via une prop `initialColor`.
- b) Ajoutez un bouton "Changer de couleur" qui, lorsqu'il est cliqué, change la couleur de fond de manière aléatoire (par exemple, utilisez `Math.random()` pour générer une couleur aléatoire en format hexadécimal).



Exercice 4: Gestionnaire de notes

a) Créer un composant qui permet de gérer une liste de notes (évaluations) avec les fonctionnalités suivantes :

- Ajouter une nouvelle note.
- Supprimer une note spécifique.
- Afficher la moyenne des notes.

Contraintes :

- Les notes doivent être comprises entre 0 et 20. Si une valeur en dehors de cet intervalle est entrée, elle ne doit pas être ajoutée.
- Le composant doit accepter une prop `initialNotes` pour initialiser la liste des notes.

Exercice 5 : Todo List avec priorités

- b) Créer un composant qui permet de gérer une Todo List avec des priorités pour chaque tâche.
- c) Ajouter une nouvelle tâche avec un nom et une priorité (par exemple : *Haute, Moyenne, Basse*).
- d) Afficher le nombre total de tâches et combien sont terminées.
- e) Marquer une tâche comme terminée
- f) Rechercher une tâche

NB : Utiliser des props pour initialiser la liste des tâches avec un tableau de tâche, sachant que chaque tâche a un nom et une priorité .

Todo List avec Priorités

Tâches :

Finir le projet React - Haute

Terminé

Supprimer

Préparer le repas - Moyenne

Terminé

Supprimer

Aller courir - Basse

Terminé

Supprimer

Total des tâches : 3

Tâches terminées : 0

Moyenne

▼

Ajouter

Tâches :

~~Finir le projet React~~ - Haute - Non terminé Supprimer

~~Préparer le repas~~ - Moyenne - Non terminé Supprimer

Aller courir - Basse Terminé Supprimer

Total des tâches : 3

Tâches terminées : 2

Todo List avec Priorités

Tâches :

Aller courir - Basse Terminé Supprimer

Total des tâches : 3

Tâches terminées : 2

Aller

Nom de la tâche

Moyenne ▾

Ajouter