# TD 3 : Gestion des événements et des données dans Vue.js

## Objectifs du TD :

- Comprendre comment **gérer les événements** utilisateur dans Vue.js.
- Manipuler les données de manière réactive avec **ref() et reactive()**.
- Apprendre à utiliser **les événements natifs et personnalisés**.

## Étape 1 : Comprendre les événements dans Vue.js

**Réflexion utile** :

- Lire la documentation sur la gestion des événements dans Vue.js.
- Expliquer avec vos mots la différence entre **un événement natif (@click, @input)** et un événement personnalisé ($emit).
- Donner **un exemple d'événement utilisateur** utilisé dans une application réelle.

## Étape 2 : Manipuler les événements utilisateur

**Consignes** :

1. Dans **App.vue**, créer un **bouton** qui affiche une alerte lorsqu'on clique dessus.
2. Modifier le bouton pour **afficher un message dans une div au lieu d'une alerte**.
3. Ajouter un **champ de texte (input)** qui met à jour un message en temps réel.

**Résultat attendu** :

- Lorsque l'utilisateur clique sur le bouton, le message s'affiche dynamiquement dans la div.
- Lorsqu'il écrit dans l'input, le texte affiché est mis à jour instantanément.

## Étape 3 : Gérer les données avec ref() et reactive()

**Consignes** :

1. Créer une **variable réactive** avec ref() pour stocker un nombre.
2. Ajouter un bouton **+1** et un bouton **-1** pour incrémenter et décrémenter cette valeur.
3. Modifier le code pour utiliser reactive() au lieu de ref().
4. Expliquer la **différence** entre ref() et reactive().

**Résultat attendu** :

- Le nombre s'affiche et **change dynamiquement** à chaque clic sur les boutons.
- **Comparez ref() et reactive()** et expliquez votre choix.

## Étape 4 : Communication entre composants avec les événements personnalisés

**Consignes** :

1. Créer un composant Child.vue qui affiche un bouton.
2. Lorsque l'utilisateur clique sur ce bouton, **un événement doit être envoyé au parent (App.vue)** avec $emit.
3. Dans App.vue, écouter cet événement et **mettre à jour un message affiché**.

**Résultat attendu** :

- Un **message change dans App.vue** lorsque l'utilisateur clique sur le bouton dans Child.vue.

## Questions / Réflexions :

- Quelle est la différence entre un événement natif et un événement personnalisé ?
- Pourquoi utilise-t-on ref() et reactive() pour stocker les données ?
- Dans quel cas $emit est-il indispensable ?

## Résumé du TD 3 :

⇢ Utilisation des **événements natifs** (@click, @input).
⇢ Gestion des données avec **ref() et reactive()**.
⇢ Communication entre composants avec **événements personnalisés ($emit)**.