

# React.js

# Utilisation du LocalStorage

TD sur l'utilisation du LocalStorage

# Sommaire

- Introduction
- Les fonctions natives
- Votre première stack

# Introduction

# Introduction

La propriété `localStorage` vous permet d'accéder à un objet local de type `Storage`.

(<https://developer.mozilla.org/fr/docs/Web/API/Storage>)

Les données stockées dans le `localStorage` n'ont pas de délai d'expiration.

Les clés et les valeurs sont toujours des chaînes de caractères.

# Introduction

Ces travaux vous permettront de vous familiariser avec l'utilisation du **LocalStorage** en JavaScript.

Nous vous conseillons de consulter le référentiel MDN suivant :  
<https://developer.mozilla.org/fr/docs/Web/API/Window/localStorage>

À votre clavier,

Prêt ?

Go !

# Les fonctions natives

# Les fonctions natives

L'API native « `LocalStorage` » en JavaScript expose quelques fonctions natives.

Telle que :

- **setItem** : ajoute une entrée associée à une clé définie
- **getItem** : récupère l'entrée associée à la clé donnée
- **removeItem** : supprime le contenu associé à la clé donnée
- **clear** : supprime l'intégralité du `localStorage` utilisé



Votre première stack

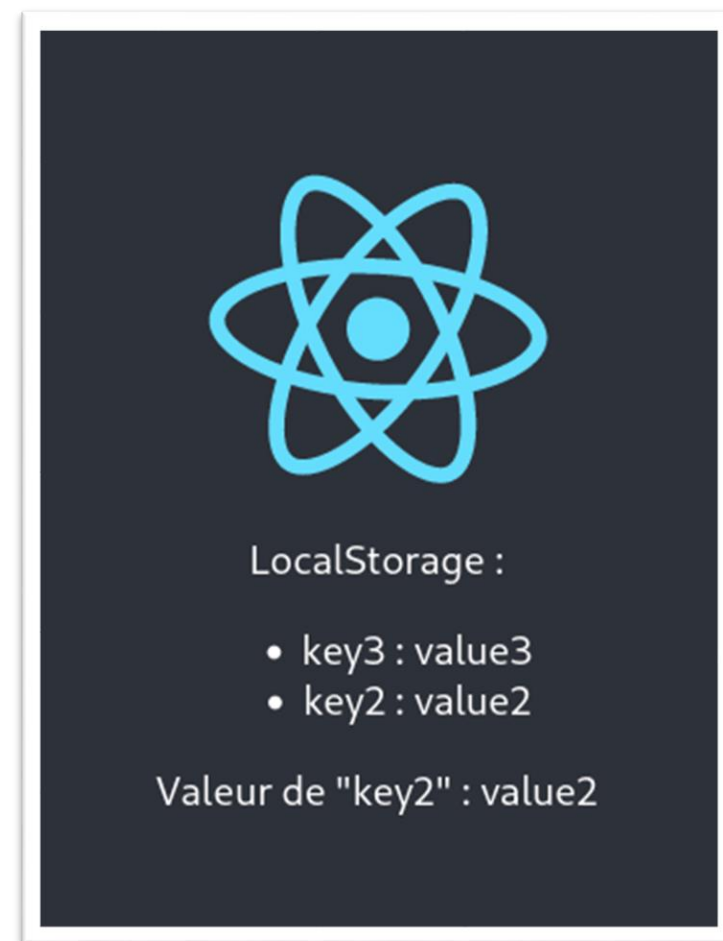
# Votre première stack

Développez le composant  
« **LocalStorageExample** » tel que le  
code ci-contre.

```
7 documents/code/react/okcapi1/src/App.js
1 function LocalStorageExample() {
2   localStorage.clear();
3
4   const ls = localStorage;
5
6   ls.setItem("key1", "value1");
7   ls.setItem("key2", "value2");
8   ls.setItem("key3", "value3");
9   ls.removeItem("key1");
10
11   return (
12     <div>
13       <span>LocalStorage :</span>
14       <ul>
15         {Object.keys(ls).map(key => (
16           <li>{key} : {ls.getItem(key)}</li>
17         ))}
18       </ul>
19
20       <p>Valeur de "key2" : {ls.getItem("key2")}</p>
21     </div>
22   );
23 }
24
25 export default LocalStorageExample;
```

# Votre première stack

Vous devriez avoir le résultat tel que celui présenté ci-contre.



# TP To-Do-List

## Objectif du TP :

L'objectif de ce TP est de créer une application de liste de tâches (to-do list) en utilisant React pour le front-end et Node.js pour le back-end. Les utilisateurs pourront ajouter des tâches, les marquer comme terminées et les supprimer. De plus, les tâches seront persistantes grâce à l'utilisation du LocalStorage

# Contrainte du TP

- Utiliser les fonctions natives de LocalStorage pour sauvegarder les taches de manière persistante
- Utiliser le hook React useState
- Utiliser des framework css comme Bootstrap ou Tailwind
- Bonus : Apporter des styles avancés aux composants ( animation, pop-up,...)

# Liens utiles :

Hooks React : <https://legacy.reactjs.org/docs/hooks-reference.html>

LocalStorage : <https://blog.logrocket.com/localstorage-javascript-complete-guide/>

Bootstrap : <https://getbootstrap.com/>

Tailwind : <https://tailwindui.com/documentation>

Pop-up : <https://fkhadra.github.io/react-toastify/introduction>

Terminé ?  
Bien joué !