

**TP7 – formatage de données en JSON,
utilisation du LocalStorage du navigateur,
utilisation de la documentation de l’API pour manipuler des fonctions**

L’objectif de ce TP est d’utiliser le localStorage du navigateur pour sauvegarder / restaurer des données.

Pré-requis : pour la réutilisation de ce TP, vous allez ré-utiliser la todolist mise en œuvre lors du TP précédent.

1 Transformation des items de votre liste en objets

Travail à réaliser :

1. Mettez en œuvre un constructeur permettant la création d’objets `ListItem`. Ces objets ont deux propriétés : une chaîne de caractères décrivant la tâche et un booléen permettant de dire si oui ou non la tâche est réalisée (rappel : dans le TP8, une classe est assignée aux tâches réalisées).
2. Vérifiez les types des arguments dans votre constructeur.
3. Testez votre constructeur :
 - Ajouter dans votre document web un bouton “Sauvegarder la liste”
 - Lorsque ce bouton est cliqué, vous devez récupérer chaque item de la liste. Pour ce faire, utilisez la méthode `each()`¹ (utiliser la documentation pour comprendre le fonctionnement de la méthode et comment l’utiliser dans votre script).
 - À chaque itération grâce à la méthode `each()`, instanciez un objet `ListItem` avec votre constructeur. Pour assigner une valeur au booléen de la classe, utilisez la méthode `hasClass()`². Affichez l’objet instancié dans votre paragraphe de démonstration.
 - (a) L’affichage vous convient-il ?
 - (b) Convertissez votre objet en texte avec JSON et affichez le à nouveau.

2 Sauvegarde de la liste

Maintenant que vous avez accès à chaque item de votre todolist, vous allez mettre en œuvre le code pour la sauvegarder dans le local storage.

Questions de cours :

1. Il y a en fait 2 espaces de stockage dans votre navigateur. Quels sont-ils ?
2. Quelle est la différence entre ces deux espaces de stockage ?
3. Quel est le seul type de données que l’on peut stocker dans le local storage ?

1. <https://api.jquery.com/each/>
2. <https://api.jquery.com/hasClass/>

4. Quel problème va donc se poser avec le modèle de données que nous avons créé pour manipuler des items de liste ?
5. Quel est le lien entre l'utilisation du format JSON et l'utilisation du local storage ?

Pour utiliser le local storage, le mieux est de s'assurer que le navigateur le supporte bel et bien. Pour ce faire, vous allez utiliser le snippet de code suivant et mettre votre code lié à l'utilisation du local storage dans le bloc :

```
if (typeof(Storage) !== "undefined") { // local storage utilisable
  // utilisation du local Storage
}
```

Travail à réaliser : pour cet exercice, vous allez modifier le code de la fonction exécutée lorsque l'événement "click" survient sur le bouton de sauvegarde de la liste.

1. Lors de l'itération sur les items de la liste, au lieu d'afficher l'objet instancié dans votre paragraphe de démonstration, vous allez l'ajouter à un tableau déclaré avant de débiter les itérations. Utilisez pour cela la méthode `push()`³
2. Stocker votre tableau dans le local storage.
 - (a) Utiliser les outils de développement de votre navigateur pour consulter le contenu du local storage.
 - (b) Le tableau est-il exploitable ? Si non, utiliser JSON avant de le stocker

3 Restaurer une liste

L'utilisation du local storage permet de terminer la session de navigation (de fermer le navigateur) tout en conservant les données sauvegardées, de façon à pouvoir les recharger au besoin.

Travail à réaliser :

1. Dans votre document HTML, ajouter un bouton "Restaurer la liste".
2. Attacher un événement "click" sur ce bouton, et écrire la fonction associée pour récupérer une todolist depuis le local storage :
 - (a) Récupérer, avec une méthode JSON, le tableau que vous avez sauvegardé dans le local storage.
 - (b) Restaurer la liste HTML avec les items que vous avez extrait du local storage
 - (c) Créer une todolist en ajoutant des items avec votre formulaire. Sauvegarder la liste, fermer votre navigateur. Rouvrez le et restaurer la liste.

3. https://www.w3schools.com/jsref/jsref_push.asp