

# Veranstaltungseinheit Local Storage

## Local Storage in der Console verwenden

### Ziel der Übung

- Sie können mit der JavaScript Console in den Local Storage schreiben und vom Local Storage lesen

### Aufgabe(n)

- Legen Sie eine neue Taskliste auf der Konsole an und speichern dieses in den Local Storage.
- Holen sie die so gespeicherte Taskliste aus dem Local Storage zurück

### Hinweis

Da im Local Storage alles als String gespeichert wird, müssen sie die TaskListe beim erneuten Laden wieder in ein `TaskList`-Objekt umwandeln.

Hier ist ein Beispiel von einem TaskListen Objekt, das mit `JSON.stringify` im Local Storage gespeichert wird:

```
""{"id":null,"title":"","tasks":
[{"title":"first task","done":false},
{"title":"second task","done":false}]""
```

Diesen String wieder in ein Objekt zu konvertieren passiert nicht automatisch. Da sie bereits eine Methode `toJSON` geschrieben haben, verwenden sie diese um das Objekt zu speichern. Dadurch ist es einfacher den String beim Auslesen in der Methode `load` in ein Objekt zu konvertieren.

### Erwartete Resultate und Abgabe

- Sie können zeigen, wie man eine Taskliste in den Local Storage schreibt und zurückholt
- Sie können zeigen wie man, den im Local Storage gespeicherten String in ein `TaskList` Objekt umwandeln kann.
- Bei Erfüllung analoger Kriterien kann das Resultat auch in ihr Projekt aus der Projektschiene integriert werden.
- Abgabe während dem Labor.

## Local Storage in das TaskListen Projekt einbinden

### Ziel der Übung

- Sie können den Local Storage in eine bestehende Applikation einbinden

### Aufgabe(n)

- Ändern Sie das abgegebene Tasklisten Projekt so ab, dass die Daten nicht mehr auf dem Server gespeichert werden, sondern im Local Storage.
- Fügen sie als erstes einen neuen Button hinzu, der den Local Storage löscht und die Seite neu lädt. Dazu sollen sie in der Datei `TaskList.js` die Funktion `TaskList.clear()` hinzufügen und diese dann über ein click Event am Button aufrufen. Für die `clear` Methode ist ein Test mit einem Mock vorhanden, der die Seite nicht neu lädt, da dies beim Aufrufen eine Endlosschleife auslöst. Daher müssen sie das Nachladen der Seite manuell testen.
- Passen Sie die Methoden `save` und `load` so an, dass diese in den Local Storage schreiben In der Datei `spec/TaskListSpec.js` sind bereits Tests vorbereitet, welche die Funktion überprüfen.
- Falls im Local Storage eine Taskliste vorhanden ist soll diese beim Aufruf der Seite automatisch geladen und angezeigt werden

### Erwartete Resultate und Abgabe

- Sie haben das TaskListen Projekt so angepasst, dass die Daten in den Local Storage gespeichert und ausgelesen werden, sowie der gesamte Local Storage gelöscht werden kann.
- Bei Erfüllung analoger Kriterien kann das Resultat auch in ihr Projekt aus der Projektschiene integriert werden.
- Abgabe während dem Labor

## Local Storage Funktionen mocken - freiwillige Übung

### Ziel der Übung

- Sie können die Funktionen des Local Storages für die Tests mocken

### **Aufgabe(n)**

- In der Datei TaskListSpec.js mocken sie die Methoden `setItem` und `getItem` des Local Storages und passen die Tests dahingehend an, dass das Mock Objekt verwendet werden.

### **Erwartete Resultate und Abgabe**

- Sie haben die Test Datei dahingehend angepasst, dass die `setItem` und `getItem` Methoden als mocks vorhanden sind und den Tests aufgerufen werden.
- Freiwillige Übung