

Todo

Ein todo-Programm mit dem man die todo's im Browser speichern kann. Es sieht wie folgt aus:



HTML

Im HTML Bereich habe ich die CSS- und JavaScript Dateien in externe Dateien ausgelagert. Im HTML Bereich gibt es kein JavaScript.

JavaScript Ansicht

Die Datei hat als erstes das Array:

- `let todos = [];`

in diesem Array werden die einzelnen todos als Objekt gespeichert.

Als nächstes werden HTML-Elemente „geholt“ und in einer Variable gespeichert:

- `const todoInput = document.querySelector('.todo-input');`
- `const todoItemsList = document.querySelector('.todo-items');`
- `const todoForm = document.querySelector('.todo-form');`

Mit der `todoForm` möchte ich anfangen denn hier wir ein Event-Listener dran gebaut. Der Event-Listener horcht auf ein `submit`. Ein Nebeneffekt ist auch der, dass ein Klick auf die Return Taste das `submit`-Event auch auslöst.

```
todoForm.addEventListener('submit', function(event) {  
  // preventDefault verhindert, dass die Seite beim Absenden des Formulars neu geladen wird  
  event.preventDefault();  
  addTodo(todoInput.value); // addTodo aufrufen & Inhalt der Textbox mitgeben  
});
```

addTodo

Das ist die Methode, die aufgerufen wird, wenn der Benutzer den Speicherbutton drückt oder die Enter Taste. Dieser Methode wird der Textinhalt mitgegeben, den der Benutzer im Textfeld eingetragen hat.

In dieser Methode wird – falls es einen Text gibt, sonst nix, zuerst ein Objekt erzeugt. Dieses neu erstellte Objekt wird dem `todo`-Array mittels `push` ans Ende hinzugefügt. Der Textinhalt wird geleert und dann eine Funktion (`addDataToLocalStorage`) aufgerufen, die das `todos`-Array speichert.

ACHTUNG!! Diese Methode braucht eigentlich keinen Parameter, da das `todos`-Array `public` ist. Ihr könnt also den Parameter ruhig weg machen.

```
function addTodo(item) {  
  if (item !== "") {  
    // Ein todo object erzeugen  
    const todo = {  
      id: Date.now(),  
      name: item,  
      completed: false  
    };  
  
    // input box leeren  
    todoInput.value = "";  
  
    // Das neu erzeugte todo Objekt wird dem public deklarierten todos Array hinzufügen  
    todos.push(todo);  
    addDataToLocalStorage(todos); // das "neue" todos Array speichern  
  }  
}
```

Die Funktionen in diesem Programm sind recht gut kommentiert, darum verzichte ich darauf jetzt weiter auf die Funktionen einzugehen.

Im `zip`-Verzeichnis habe ich die gesamte Applikation zur Verfügung gestellt. Hier habe ich eine Änderung zum Video-Tutorial eingebaut:

In der Methode `renderTodos` hatte ich ursprünglich mit einem `innerHTML` gearbeitet und auf diese Art eine Checkbox, den Text und den Löschen-Button eingebaut. Das habe ich in dem Programm, dass Ihr Euch downloaden könnt mit einem „`createElement`“ realisiert. Der Original-Code ist dennoch mit dabei allerdings auskommentiert.