

## JavaScript Breakdown for the To-Do Card App

---

### 1. Variable Setup

---

```
const addCard = document.getElementById('add-card');
```

```
// Finds the "Add Card" button in your HTML.
```

```
const deleteCard = document.getElementById('delete-card');
```

```
// Finds the "Delete Card" button.
```

```
const cardContainer = document.querySelector('#card-list');
```

```
// Finds the list (ul) where all cards will be placed.
```

---

### 2. Load Saved Cards on Page Load

---

```
window.addEventListener('DOMContentLoaded', loadCardsFromStorage);
```

```
// When the page finishes loading, it runs the function to load saved cards from localStorage.
```

---

### 3. Add New Card

---

```
addCard.addEventListener("click", () => {
```

```
const cardData = {  
  title: "",  
  text: "",  
  id: Date.now()  
};  
  
// Creates a blank card object with a unique ID (based on the current time).  
  
const cardElement = createCardElement(cardData);  
  
// Builds the HTML for the card.  
  
cardContainer.appendChild(cardElement);  
  
// Adds the card to the page.  
  
savedCard(cardData);  
  
// Saves the card in localStorage.  
});
```

-----

#### 4. Delete Checked Cards

-----

```
deleteCard.addEventListener("click", () => {  
  const checkboxes = document.querySelectorAll('#card-list .checkbox-1');  
  
  // Finds all the delete checkboxes.  
  
  let cards = getSavedCards();  
  
  // Gets current saved cards from localStorage.
```

```
checkboxes.forEach(checkbox => {
  if (checkbox.checked) {
    const li = checkbox.closest('li');
    const cardId = li.dataset.cardId;
    li.remove();
    cards = cards.filter(c => c.id !== cardId);
  }
});

localStorage.setItem('cards', JSON.stringify(cards));

// Updates localStorage with the new (filtered) list of cards.
});
```

## -----

### 5. Create Card Element (HTML + Functionality)

## -----

```
function createCardElement(cardData) {
  const li = document.createElement('li');
  li.dataset.cardId = cardData.id;
  // Stores the card's ID in a custom attribute.

  li.innerHTML = `...`;
  // HTML structure for the card is added.

  const titleInput = li.querySelector('.card-title');
```

```
const textArea = li.querySelector('.card-text');

titleInput.value = cardData.title;

textArea.value = cardData.text;

// Sets saved text if the card existed before.

titleInput.addEventListener('input', () => updateCardData(li));
textArea.addEventListener('input', () => updateCardData(li));

// Whenever the user types, update storage.

return li;
}
```

---

## 6. Update Card Info

---

```
function updateCardData(li) {

  const cards = getSavedCards();

  const id = parseInt(li.dataset.cardId);

  const title = li.querySelector('.card-title').value;
  const text = li.querySelector('.card-text').value;

  const cardIndex = cards.findIndex(c => c.id === id);

  if (cardIndex !== -1) {

    cards[cardIndex].title = title;

    cards[cardIndex].text = text;

    localStorage.setItem('cards', JSON.stringify(cards));
```

```
}  
  
}
```

// Finds the card in saved data and updates the title/text.

-----

## 7. Save a New Card to Storage

-----

```
function savedCard(cardData) {  
  const cards = getSavedCards();  
  cards.push(cardData);  
  localStorage.setItem('cards', JSON.stringify(cards));  
}
```

// Adds a new card to saved list and saves it.

-----

## 8. Get All Cards from Storage

-----

```
function getSavedCards() {  
  return JSON.parse(localStorage.getItem('cards')) || [];  
}
```

// Reads the JSON string from localStorage and turns it into an array.

-----

## 9. Load Cards from Storage

-----

```
function loadCardsFromStorage() {  
  const cards = getSavedCards();  
  cards.forEach(cardData => {  
    const cardElement = createCardElement(cardData);  
    cardContainer.appendChild(cardElement);  
  });  
}
```

// Loops through each saved card and adds it to the page.

-----

Extra: Date.now()

-----

Date.now() gives the number of milliseconds since January 1, 1970.

// It's a quick way to get a unique number to use as an ID.