

Font awesome CDN

<https://cdnjs.com/libraries/font-awesome>

Font Awesome Icons

<https://fontawesome.com/icons/trash-alt?style=solid>

localStorage

The read-only `localStorage` property allows you to access a Storage object for the Document's origin; the stored data is saved across browser sessions. `localStorage` is similar to `sessionStorage`, except that while data stored in `localStorage` has no expiration time, data stored in `sessionStorage` gets cleared when the page session ends — that is, when the page is closed (Data in a `localStorage` object created in a "private browsing" or "incognito" session is cleared when the last "private" tab is closed.)

The following snippet accesses the current domain's local Storage object and adds a data item to it using `Storage.setItem()`.

```
localStorage.setItem('myCat', 'Tom');
```

The syntax for reading the `localStorage` item is as follows:

```
const cat = localStorage.getItem('myCat');
```

The syntax for removing the `localStorage` item is as follows:

```
localStorage.removeItem('myCat');
```

The syntax for removing all the `localStorage` items is as follows:

```
localStorage.clear();
```

The `JSON.stringify()` method converts a JavaScript object or value to a JSON string, optionally replacing values if a replacer function is specified or optionally including only the specified properties if a replacer array is specified.

```
console.log(JSON.stringify({ x: 5, y: 6 }));
```

```
// expected output: '{"x":5,"y":6}'
```

The `JSON.parse()` method parses a JSON string, constructing the JavaScript value or object described by the string. An optional `reviver` function can be provided to perform a transformation on the resulting object before it is returned.

```
const json = '{"result":true, "count":42}';
```

```
const obj = JSON.parse(json);
```