



WEB ENGINEERING

HANDLE DATA WITH LOCAL STORAGE

- Data Structure
- **CRUD** Operations
 - **C**reate
 - **R**ead
 - **U**ppdate
 - **D**eleate
- JSON with `localStorage`

- The [Web Storage API](#) offers two ways to handle data: **sessionStorage** and **localStorage**
- **localStorage** keeps data across browser sessions
- Similar to JSON, the **localStorage** stores data in the form of key / value pairs
- **localStorage** stores only character strings

LocalStorage

key	value
name	"Mike Miller"
age	"41"

Create Data

Read Data

Uppdate Data

Deleate Data

- We can **add** new key / value pairs:

```
localStorage.setItem("name", "Mike");
```

- We can **read** the value of a particular key:

```
let name = localStorage.getItem("name");
```

CHANGE DATA

UPDATE

- We can **update** the value of a particular key by setting a new value:

```
let name = localStorage.getItem("name");  
let fullName = name + " Miller";  
localStorage.setItem("name", fullName);
```

DELETE DATA

DELETE

- We can **delete** a particular key from storage:

```
localStorage.removeItem("name");
```

- We can **delete** all data, too:

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


- `localStorage` can only store character strings
- To store JSON data, we need to store JSON as a string and parse it back when reading:

```
profile = { name : "Mike Miller", age: 42 };  
profileString = JSON.stringify(profile);  
localStorage.setItem("profile", profileString);
```

- The same way, we can parse a string back into a JSON object:

```
let profileString = localStorage.getItem("profile");  
let profile = JSON.parse(profileString);
```