Page No. | Date

✓

What is local stoeage

- Application data

① set up a database to stoee and reteieve data

② use local storage to store and reteieve data

- The local storage api is provided by the browsee and it gives a way to store the data and reteive data to and from the browsers itself. i.e. when a user enters information on our web app we can then store that in the browsers local storage on the users computee.

And if at some point, they refresh the page or crossed it up completely and come back latee then we can access that data that we previously stoeed from local storage in the browsee and injected back to our application.

✓

Stoeing and Getting Data

- local storage is always on the window object so we can type in console

window.localStorage   to get that

→ ▸ stoeage { Length: 0 }

localstorage is stoeed on window object and the window object is the global object so we can also weite it as

localstoeage

→ ▸ stoeage { Length: 0 }

- how to store data in local storage

```
localstorage.setItem('name', 'mario');
```

setItem method takes two arguments, first is the 'key' and second is 'value of that key'

- how to get the data from local storage

```
let name = localstorage. getItem('name');
          console.log(name);
```

- update data

```
localstorage.setItem('name', 'luigi');
name = localstorage. getItem('name');
console.log(name);
```

```
- localstorage.setItem('name', 'mario');
  localstorage.setItem('age', 50);
```

} setting/storing

```
  let name = localstorage.getItem('name');
  let age = localstorage.getItem('age');
```

} getting data

```
  console.log(name, age);
```

both ways you can use to update the data

```
  localstorage.setItem('name', 'luigi');
  localstorage.age = '40';
  name = localstorage.getItem('name');
  age = localstorage.getItem('age');
```

} updating data

```
  console.log(name, age);
```

→ mario 50
luigi 40

✓

Deleting storage Data

for deleting single item

```
localstorage.setItem('name', 'mario');
localstorage.setItem('age', 50);
let name = localstorage.getItem('name');
let age = localstorage.getItem('age');
console.log(name, age);
// deleting data from local storage
localstorage.removeItem('name');
name = localstorage.getItem('name');
console.log(name, age);
```

→ mario 50
null 50

for removing all items in local storage that are previously set

```
localstorage.setItem('name', 'mario');
localstorage.setItem('age', 50);
let name = localstorage.getItem('name');
let age = localstorage.getItem('age);
console.log(name, age);
// deleting all items from local storage
localstorage.clear();
      name = localstorage.getItem('name');
      age = localstorage.getItem('age');
      console.log(name, age);
```

→

```
mario 50
null. null
```

[.clear() method doesn't take any argument but just clears out local storage]

✓

stringifying and Parsing the data

- Data we store in local storage must ultamitely be a string.

Note -

```
const todos = [
    { text:'play', author: 'shaun'},
   { text:'dance', author : 'luigi'},
    { text : 'sing', author : 'mario'}
];
```

This is array of objects

here text and author do not have double quote but for json it is there. This is array of javascript object, so it don't have double quotes.

Now we need to store this in local storage.

So the first step is take data(above) and turned in into a json string.

So, in past we have seen in asynchronous chapter how to take a json string and parse it into javascript array of object.
But here now we want to do opposite

Here we got array of object and we need to turn that into a json string.
So How To Do That

for doing this you can use method called stringify()

```
- Const todos= [
    {text:'play', author:'shaun'},
     {text:'dance', author:'mario'},
    {text:'song', author:'luigi'}
    ];
```

Array of objects

```
console.log(JSON.stringify(todos));
        ↓
[{"text":"play","author":"shaun"},
   {"text":"dance","author":"mario"},
 {"text":"song", "author":"luigi"}
]
```

Now this is one gigantic string which looks like an array of object but its a json string

Notice how our properties like "text" and "author" they are all quoted and we know that json has double quotes.

Here we have array of objects and we turn that now into a valid json

And Now this is the good format to store in local storage.

```
- const todos = [
      { text: "play", author: "mario"},
      { text: "dance", author: "shaun"}
  ];

localstorage.setItem('todos', JSON.stringify(todos));
```

↓

Here now we have stored that data into local storage.

Now how do we retrieve it and then convert back into an array.
for that consider below code

```
- const todos = [
      { text: "play", author: "mario"},
      { text: "dance", author: "shaun"}
  ];

localstorage.setItem('todos', JSON.stringify(todos));
const stored = localstorage.getItem('todos');
console.log(JSON.parse(stored));
```

↓

```
▸ (2) [{…}, {…}]
```

(Now here converted it into an array)