@Code.Clash

SessionStorage

JavaScript

SessionStorage

The sessionStorage object stores data only for a session. It means that the data stored in the sessionStorage will be deleted when the browser is closed.

- When you open a page in a new tab or window, the web browser creates a new session.
- If you open multiple tabs or windows with the same URL, the web browser creates a separate sessionStorage for each tab or window.
- if you wish to store objects, lists, or arrays, you must convert them into a string using JSON.stringify().

Accessing SessionStorage

You can access the SessionStorage via the property of the window object.

Here's an example:

```
window.sessionStorage
```

When you type Upper Code in the console. you'll see the following object:

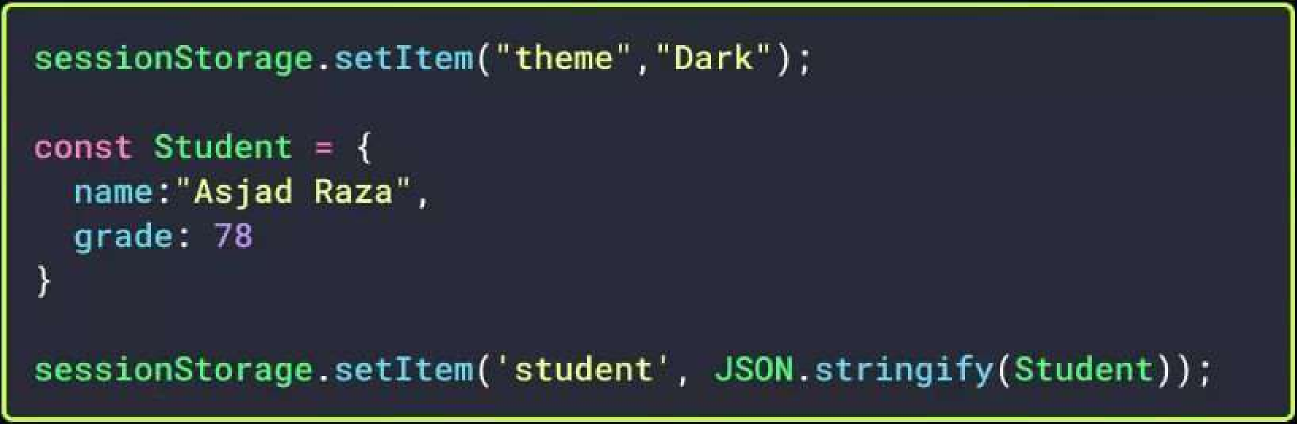
```
Storage {length: 0}
```

SessionStorage Methods

SessionStorage provides some methods and properties.

setItem(key, value)

- This method is used to store a name-value pair in the sessionStorage.
- As mentioned before, we must stringify objects before we store them in the Session Storage.

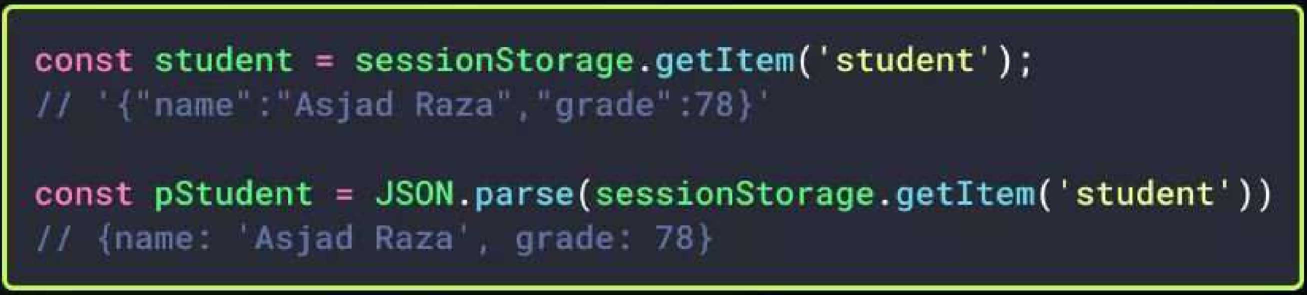


```
sessionStorage.setItem("theme","Dark");

const Student = {
  name:"Asjad Raza",
  grade: 78
}

sessionStorage.setItem('student', JSON.stringify(Student));
```

getItem(key)

- This method is used to access or retrieve data from the SessionStorage. A key is used as a parameter.

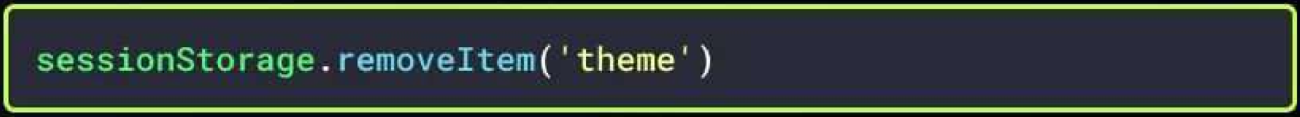
- You should convert it to an object using JSON.parse() to use it in your code



```
const student = sessionStorage.getItem('student');
// '{"name":"Asjad Raza","grade":78}'

const pStudent = JSON.parse(sessionStorage.getItem('student'))
// {name: 'Asjad Raza', grade: 78}
```

removeItem(key)

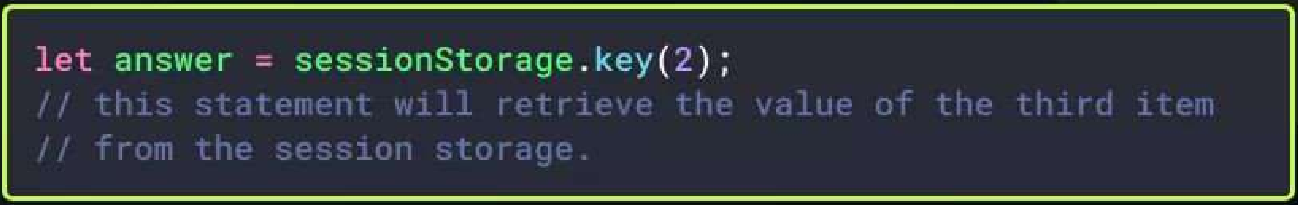
- This method is used to delete an item from SessionStorage. A key is used as a parameter.



```
sessionStorage.removeItem('theme')
```

key(index)

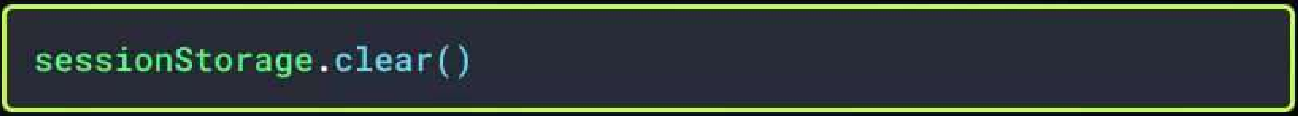
- This method retrieves a value from a specific location. The index can be passed as a parameter.



```
let answer = sessionStorage.key(2);
// this statement will retrieve the value of the third item
// from the session storage.
```

clear()

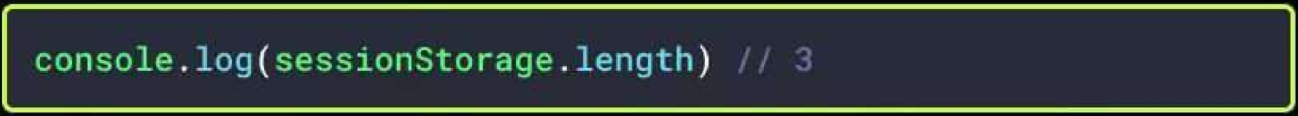
- This method is used to clear all values stored in session storage.



```
sessionStorage.clear()
```

length

- This property is used to get the number of name-value pairs from sessionStorage.



```
console.log(sessionStorage.length) // 3
```

Loop Over Keys

The following stores three name-value pairs to the Session-Storage:

```
sessionStorage.setItem('theme','light');
sessionStorage.setItem('backgroundColor','white');
sessionStorage.setItem('color','#111');
```

To iterate over name-value pairs stored in the sessionStorage, you use the `Object.keys()` method with for...of loop:

```
let keys = Object.keys(sessionStorage);
for(let key of keys) {
  console.log(`${key}: ${sessionStorage.getItem(key)}`);
}

// color: #111
// theme: light
// backgroundColor: white
```

LocalStorage Vs SessionStorage

localStorage:

- Data stored in localStorage persists even after the browser is closed and reopened.
- The data is available across all tabs and windows of the same origin (i.e., protocol, domain, and port).
- The data can be accessed by any JavaScript code on the same domain.

sessionStorage:

- Data is only persists for the duration of the user's session on a specific tab.
- The data is only available to the page that created it and is not shared across tabs or windows of the same origin.
- The data can be accessed by any JavaScript code running on the same page that created it.