

GABRIELE ROMANATO

Gestione della sessione in JavaScript

La gestione della sessione in JavaScript riguarda la conservazione temporanea di dati tra una pagina e l'altra durante la navigazione di un utente. JavaScript, eseguito lato client, utilizza principalmente **sessionStorage** e **localStorage** per memorizzare informazioni nella sessione del browser.

sessionStorage

sessionStorage consente di salvare dati per la durata della sessione della pagina. I dati vengono persi quando la scheda del browser viene chiusa.

```
// Salvare un dato
sessionStorage.setItem('utente', 'Mario');

// Recuperare un dato
const utente = sessionStorage.getItem('utente');
console.log(utente); // Mario

// Rimuovere un dato
sessionStorage.removeItem('utente');

// Pulire tutta la sessione
sessionStorage.clear();
```

localStorage

localStorage funziona in modo simile a sessionStorage, ma i dati persistono anche dopo la chiusura del browser.

```
// Salvare un dato
localStorage.setItem('tema', 'scuro');
```

```
// Recuperare un dato
const tema = localStorage.getItem('tema');
console.log(tema); // scuro

// Rimuovere un dato
localStorage.removeItem('tema');

// Pulire tutto il localStorage
localStorage.clear();
```

Conservare Dati Complessi

Poiché `sessionStorage` e `localStorage` accettano solo stringhe, è necessario convertire gli oggetti in JSON.

```
const impostazioni = {
  lingua: 'it',
  layout: 'griglia'
};

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

const dati =
JSON.parse(localStorage.getItem('impostazioni'));
console.log(dati.lingua); // it
```

Considerazioni di Sicurezza

I dati salvati con `sessionStorage` o `localStorage` non sono criptati e possono essere letti da qualsiasi script presente nella stessa origine. Evita di salvare informazioni sensibili come password o token di accesso.

Conclusione

La gestione della sessione lato client in JavaScript è utile per migliorare l'esperienza utente, ma va utilizzata con attenzione per motivi di sicurezza e coerenza dei dati.