

GABRIELE ROMANATO

JavaScript: usare il session storage

Il session storage è una nuova caratteristica introdotta con le API di HTML5. Tramite l'oggetto `sessionStorage` possiamo memorizzare sul computer dell'utente dei dati che verranno automaticamente rimossi dal browser al termine della sessione di navigazione. Vediamo insieme questa caratteristica.

Salvare i dati

I dati vengono salvati come stringhe. Ciascun dato è una coppia formata da una chiave e da un valore:

```
var Storage = {  
  set: function (item, value) {  
    sessionStorage.setItem(item, value);  
  },  
  // continua  
};
```

Esempio:

```
Storage.set('test', 'value');
```

Abbiamo appena creato la voce `test` il cui valore è `value`.

Reperire i dati

I dati vengono reperiti conoscendo il nome della voce:

```
var Storage = {  
  set: function (item, value) {  
    sessionStorage.setItem(item, value);  
  },  
  get: function (item) {  
    var value = sessionStorage.getItem(item);  
    return value;  
  },  
  // continua  
};
```

Esempio:

```
var test = Storage.get('test');  
console.log('test'); // 'value'
```

Conteggio delle voci, rimozione delle voci ed eliminazione totale dei dati

Possiamo reperire il numero di voci presenti, rimuoverle e cancellare i dati in questo modo:

```
var Storage = {  
  set: function (item, value) {  
    sessionStorage.setItem(item, value);  
  },  
  get: function (item) {  
    var value = sessionStorage.getItem(item);  
    return value;  
  },  
};
```

```
count: function () {
    var len = sessionStorage.length;
    return len;
},
remove: function (item) {
    sessionStorage.removeItem(item);
},
empty: function () {
    sessionStorage.clear();
},
// continua
};
```

Salvare oggetti

Per impostazione predefinita il session storage accetta valori solo sotto forma di stringhe. Qualora volessimo salvare oggetti è necessario far ricorso ai metodi `JSON.stringify()` e `JSON.parse()`:

```
var Storage = {
    set: function (item, value) {
        sessionStorage.setItem(item, value);
    },
    get: function (item) {
        var value = sessionStorage.getItem(item);
        return value;
    },
    count: function () {
        var len = sessionStorage.length;
        return len;
    },
    remove: function (item) {
        sessionStorage.removeItem(item);
    },
    empty: function () {
        sessionStorage.clear();
    },
    saveObject: function (item, obj) {
```

```
        if (typeof obj === 'object') {
            this.set(item, JSON.stringify(obj));
        } else {
            this.set(item, 'Could not convert to JSON
string');
        }
    },
    getObject: function (item) {
        var json = this.get(item);
        var obj = JSON.parse(json);
        return obj;
    }
};
```