



@pyplane\_code

# Javascript Local Storage

JS





# Introduction

The local storage object enables saving key/value pairs in the users browser without any expiration date (gets cleared only through JavaScript or clearing the browser cache). It's used for storing information on the client side

Table of contents:

- setItem method
- getItem method
- check local storage objects in dev tools
- removeItem method
- clear method
- working with objects





# setItem

This method enables storing values in the local storage. It takes 2 arguments – first the key name and the second its value. If the key exists the value will be updated



```
localStorage.setItem('name', 'Rex')  
localStorage.setItem('age', 5)  
localStorage.setItem('toys', ['teddy bear', 'bone', 'duck'])
```



# getItem

This method returns the value of the given Storage object item specified by the key



```
const name = localStorage.getItem('name')
const age = localStorage.getItem('age')
const toys = localStorage.getItem('toys').replaceAll(", ", ", ")

console.log(`My dog's name is ${name}, age is ${age}, toys: ${toys}`)
```

output:

```
My dog's name is Rex, age is 5, toys: teddy bear, bone, duck
```



# dev tools

Open dev tools (in Chrome right click and select inspect). Next go to application tab and select Local Storage to view all your browser's local storage

The screenshot shows the Chrome DevTools Application tab. The left sidebar lists various storage areas: Manifest, Service Workers, Storage, Local Storage (expanded), Session Storage, IndexedDB, Web SQL, Cookies, Trust Tokens, and Interest Groups. The 'http://localhost:3000' entry under Local Storage is selected. The main panel displays a table of stored data.

Key	Value
age	5
name	Rex
toys	teddy bear, bone, duck



# remove item

This method takes in an argument as a key name that will be removed from the storage object if it does exist



```
const name = localStorage.getItem('name')
const age = localStorage.getItem('age')

localStorage.removeItem('toys')
const toys = localStorage.getItem('toys')

console.log(`My dog's name is ${name}, age is ${age}, toys: ${toys}`)
```

output:

```
My dog's name is Rex, age is 5, toys: null
```



# clear all items

The clear method will remove all items in the local storage



```
localStorage.clear()

const name = localStorage.getItem('name')
const age = localStorage.getItem('age')
const toys = localStorage.getItem('toys')

console.log(`My dog's name is ${name}, age is ${age}, toys: ${toys}`)
```

output:

```
My dog's name is null, age is null, toys: null
```



# Working with objects

Instead of creating separate key/value pairs you can create an object and store it in the local storage, however this will only work with the help of json



1

```
const dog = {name: 'Rex', age: 5, toys: ['teddy bear', 'bone', 'duck']}  
localStorage.setItem('myDog', JSON.stringify(dog))
```

2

```
const {name, age, toys} = JSON.parse(localStorage.getItem('myDog'))  
console.log(`My dog's name is ${name}, age is ${age}, toys: ${toys}`)
```

output:

```
My dog's name is Rex, age is 5, toys: teddy bear,bone,duck
```



# Did You Find it Useful?



**Alamin CodePapa**  
@CodePapa360

Follow for more

Like | Comment | Repost

