
Local
Storage

Agenda

- Why do we need to use local Storage
  - What is local storage
  - Temporarily solution
- Local storage features
- Usage
  - Set
  - get
- Parsing with JSON

Content Table

- How to store something in local storage
- How to use the data stored in local storage
- persistence
- domain
- global variable
- setItem(), getItem()
- delete keyword
- removeItem()
- iterate through the items saved in local storage
- clear()
- JSON.stringify()
- JSON.parse()

Looking back on our game of war what was missing?

Looking back on our game of war

Saving the data.

We previously built the game of war in the precourse, which let you play a game and see your current score (the money you earned).

What if we wanted to record the programmers scores every time they coded? Maybe keep track of a programmers high score?

Hard coding wont work anymore!

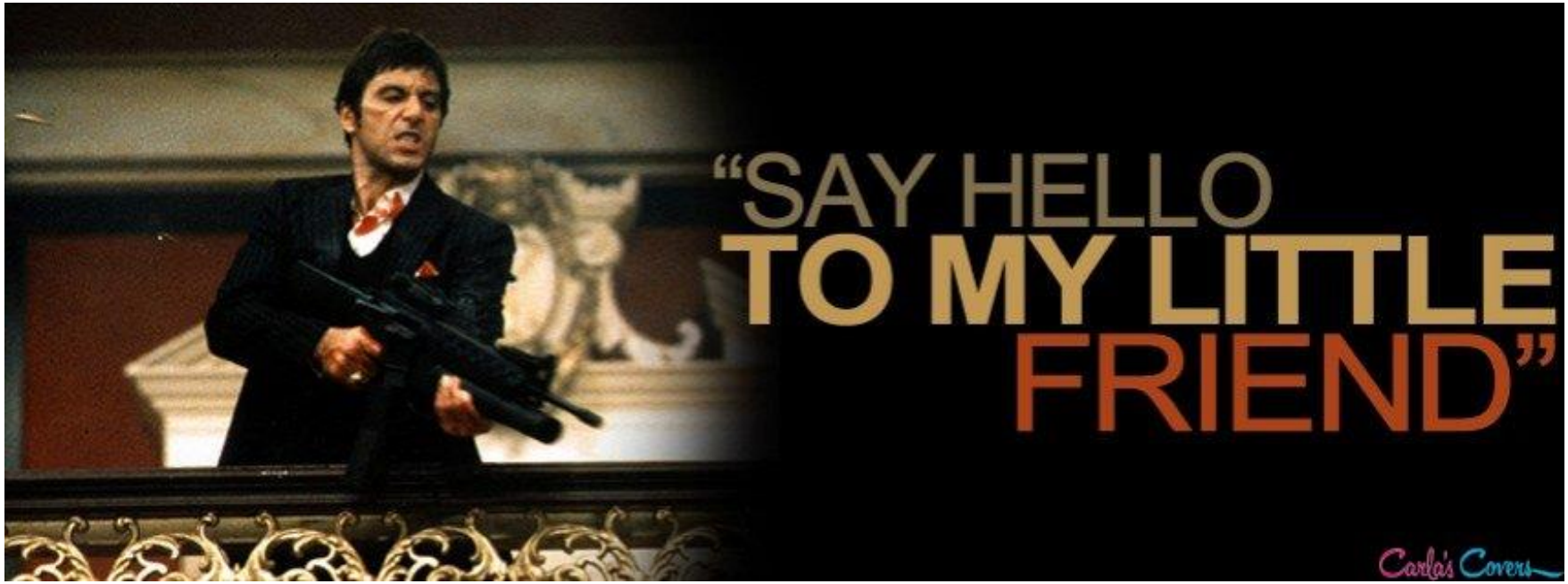
We need a way to write (save) that data!

Temporary Solution

* We Will Learn The Real Solution in the Backend Part

* But in the mean time

"Say hello to my little friend"



Local storage

Persistent data

* Data that will persist (= saved)

* Data will not be lost on a browser reload / refresh / restart.

Local storage – a quick fix

- **No expiration:** The localStorage object stores data with no expiration date. That means it will persist when the browser is closed until we explicitly delete it.
- **Save and read data:** With localStorage we can easily save and read data without setting up a database.
- **Demo:** localStorage is perfect for setting up an application with little data, or building a demo.
- **Experiment:** localStorage allows us to experiment with persistent data.

Let's take it from the top

LocalStorage Features

- Where?

  localStorage is accessed via Javascript executed in the browser.

- How much?

  localStorage can retain between 5MB-10MB (*per domain)*.

- For whom?

  localStorage keeps a record of data ***per domain***.

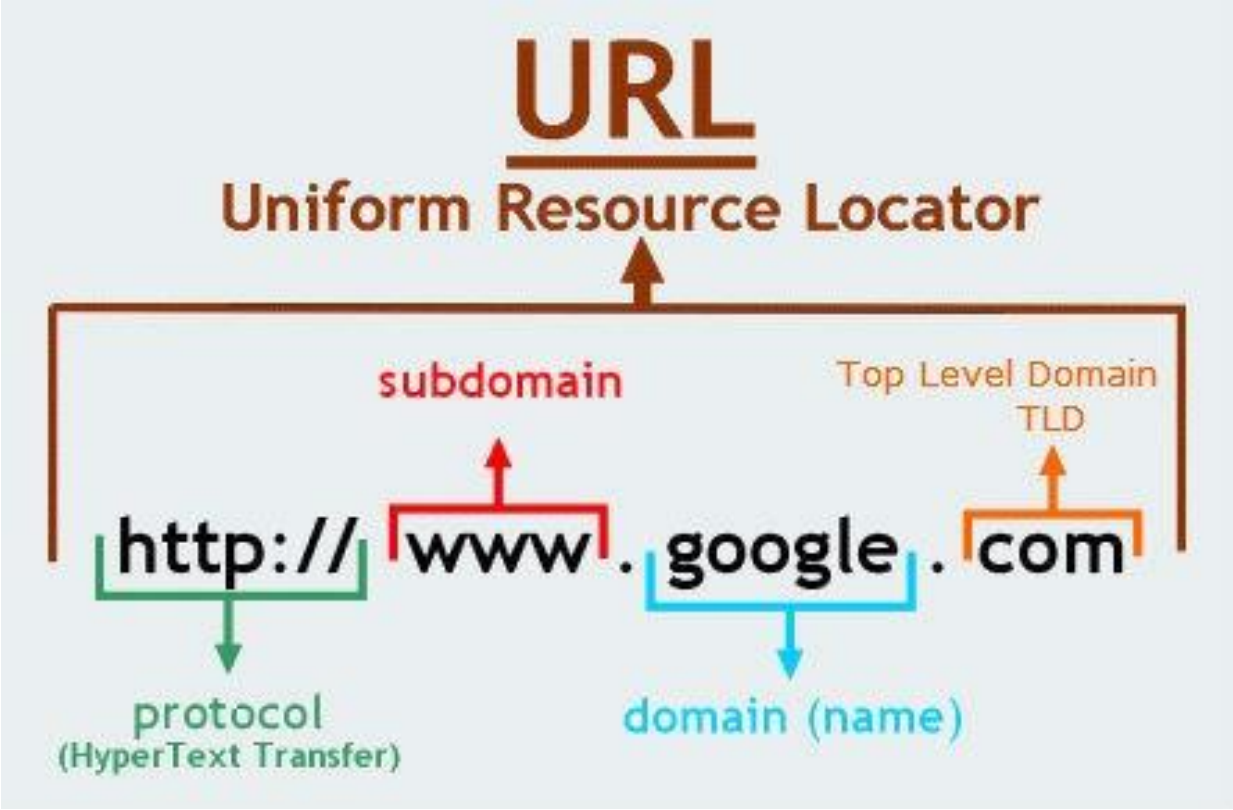
- Limitations?

  Data set to localStorage can only be read via pages with an identical domain (site name).

What is a Domain?

Parts of a URL

What is a Domain?



Domain Basis Access

Why does local storage persisting on a per domain basis even make sense?

Or in other words: If a page from itcbootcamp.com sets data to localStorage, only a page from itcbootcamp.com can read it. WHY?

Security reasons!

If every tab had access to a global localStorage object, every app would have access to other applications data!
Imagine your app reading data written to localStorage by Facebook for example…

And encapsulation! (maintain seperate scopes)

localStorage Object

Type in the console:

```
localStorage;
```

or

```
window.localStorage;
```

localStorage is a **global variable** available in all browser windows.

What is the type of localStorage?

Object

LocalStorage Object

How Do We Set Data in the LocalStorage?

- Well, the localstorage is an object.
- Saving data in the localstorage is just like setting properties on an object
- Getting data from the localstorage is just like getting the value of a property from any other object
- We will also have special functions that will help us.

localStorage Set

Save data with a dot notation assignment.

```
localStorage.username = "Peter Quill";
```

```
> localStorage.userName = "Peter Quill";
"Peter Quill"
> localStorage.rocketScores = [2222, 4354, 1314];
▶ [2222, 4354, 1314]
```

Setting data with string keys

Bracket notations

```
localStorage["my secrect name"] = "Star lord";
```

Setting data with setItem

`setItem` function

Function signature

```
setItem(key: string, data: string): void
```

```
localStorage.setItem("my secret name", "Star lord");
```

Accessing the data

We can access data via

- Bracket notation
- Dot notation
- The built-in *getItem()*.

```
> localStorage[" My secret name is "] = "Star lord";
<· "Star lord"
> localStorage[" My secret name is "]
<· "Star lord"
> localStorage.setItem("Groot Motto", "My name is GROOOOOOOT");
<· undefined
> localStorage.getItem("Groot Motto")
<· "My name is GROOOOOOOT"
```

Accessing the data After refresh

Even after refresh, or reopen.

After storing data to localStorage, that data can be accessed again after we reopen the window and return to the same domain.

Ways to set data to localStorage - Summary

Setting Data

1. Dot notation: *localStorage.score = 756;*
2. Bracket notation: *localStorage["best-score"] = "132";*
3. Built-in *setItem()* function: *localStorage.setItem("deadpool", "good movie");*

```
> localStorage.score = "56";
< "56"
> localStorage["best-score"] = "132";
< "132"
> localStorage.setItem("deadpool", "good movie");
< undefined
```

Ways to get data from localStorage - Summary

Getting Data

1. dot notation. Like getting a property from an object.

   localStorage.rocketScores;
2. bracket notation. *localStorage["newRocketScores"];*
3. built in *getItem()* function. *localStorage.getItem("PeterNickName")*

```
> localStorage.rocketScores
"1234,3452,654,234"
> localStorage["newRocketScores"]
"1243,432,332"
> localStorage.getItem("PeterNickName")
"Starlord"
```

localStorage examples

Ways to get data from localStorage - Summary

What happens if we try to access an object that does not exist?!

Edge Case – key doesn't exists

What happens if we try to access an object that does not exist?!

When using getItem function:

```
> localStorage.getItem("User2Data");
<· null
```

Deleting data

So we have set a bunch of data, and now we don't need it anymore. Let's see two ways we can delete items from *localStorage:*

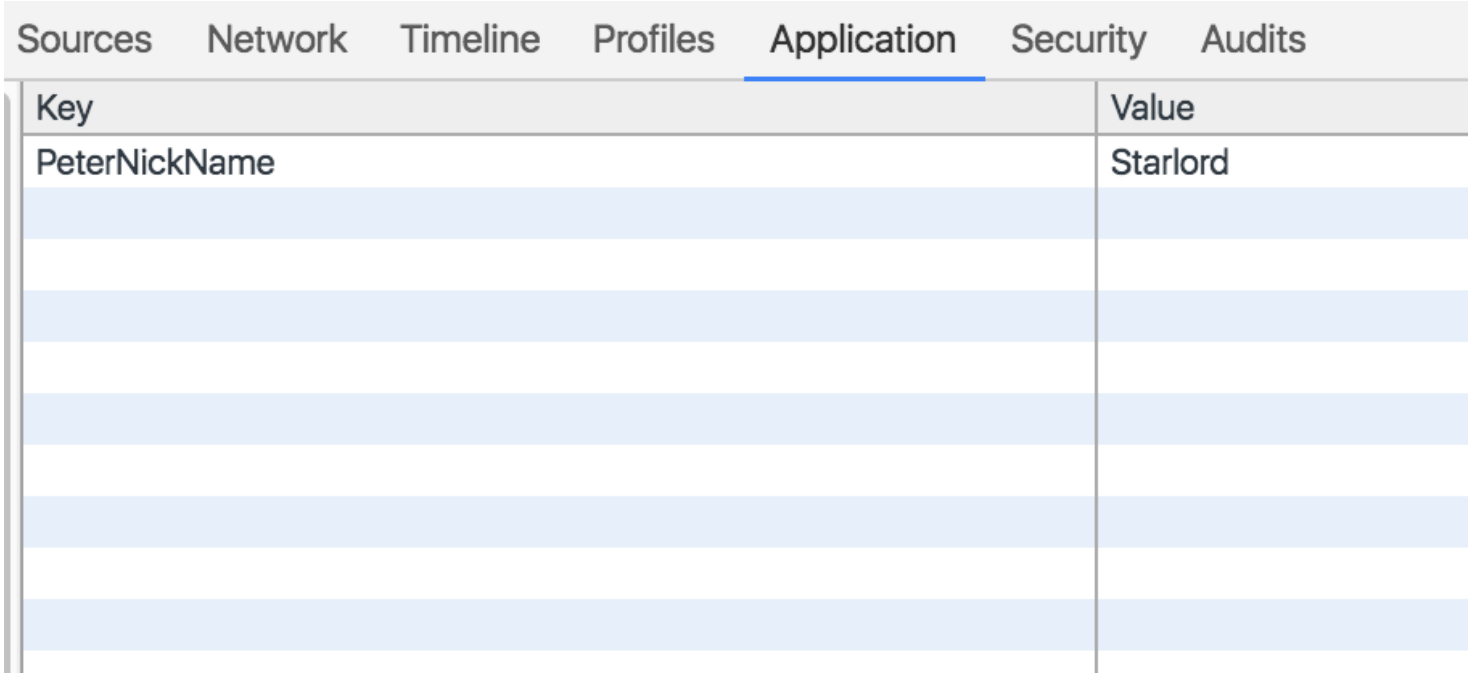
1. *delete* keyword
2. *removeItem* function.

```
> delete localStorage[" My secret name is "]
<· true
> localStorage.removeItem("Groot Motto")
```

Local Storage GUI

We also have a GUI(=Graphic User Interface) in the dev tools

For easy access and usage



Using the console

We have demonstrated storing and fetching data via the built in browser API. We can also view and delete all of the data in *localStorage* via the console tools.

- Open the dev tool
- Navigate to the *Application* tab.
- Click on *localStorage*
- Select the domain name you are working on.
- Now you can view all the data.
- To delete just right click on it and select delete.

Length

Say we wanted to programmatically check, how many items have been saved to the *localStorage object*
Simple!

```
> localStorage.setItem("PeterNickName", "Starlord");
< undefined
> localStorage.setItem("GrootNickName", "My NAME IS GROOOOOT");
< undefined
> localStorage.setItem("Drax", "The Destroyer");
< undefined
> localStorage.length
< 3
>
```

Iterating through keys in localStorage

So we know how many items have been saved to our *localStorage*.
How can we iterate through all of them?

```
> for(var i=0; i < localStorage.length; i++){

      var propertyName = localStorage.key(i);

      console.log(  i + " : " + propertyName + " = " +
              localStorage.getItem(propertyName));
  }
  0 : Drax = The Destroyer
  1 : GrootNickName = My NAME IS GROOOOOT
  2 : LH::tabs = {"s-x5FSWLuWKYvxUKqAi-AG":true}
  3 : LH::tabs-md-s-x5FSWLuWKYvxUKqAi-AG = {"tabId":"s-x5FSWLuWKYvxUKqAi-
  AG","lastSent":5,"savedIndex":5,"lastTS":1481811141741,"uid":"d92e5676"}
  4 : PeterNickName = Starlord
```

Clear all data

Finally, what if we want to delete all the data we have stored to our localStorage?

```
> localStorage.clear();
< undefined
```

Key	Value

What's the deal with strings?

Save Data as a String

- We have seen a bunch of examples so far, most of which have used strings as keys and values?
- Why is that?
- localStorage only supports strings.
- It is more comfortable for the computer to handle data as a string

Parsing Data

To play it safe: **always parse the data to a string format before saving to localStorage**.

Turning our data into Strings

The function we will use to turn all our data into strings is *JSON.stringify()*.

```
1  JSON.stringify({});                    // '{}'
2  JSON.stringify(true);                  // 'true'
3  JSON.stringify('foo');                 // '"foo"'
4  JSON.stringify([1, 'false', false]); // '[1,"false",false]'
5  JSON.stringify({ x: 5 });              // '{"x":5}'
6
7  JSON.stringify(new Date(2006, 0, 2, 15, 4, 5))
```

Reverted our data to its original form

The function we will use to revert our data to its original form is *JSON.parse ()*.

```
JSON.parse('{}');              // {}
JSON.parse('true');            // true
JSON.parse('"foo"');           // "foo"
JSON.parse('[1, 5, "false"]'); // [1, 5, "false"]
JSON.parse('null');            // null
```

Example

```
var user = {
    name: "Lotem",
    username: "Lolo"
}
var parsedUser = JSON.stringify(user);
localStorage.setItem("user", parsedUser);
parsedUser = localStorage.getItem("user");
JSON.parse(parsedUser);
```

Summary

- You needed to understand:
  - What is local storage and how to store, get, remove and clear items from it
- You need to remember:
  - How useful localstorage is
- You need to be able to do:
  - Use localstorage properties with regular JS
  - Keep track of your users information, his score ….

Questions ?

Cheat Sheet

Set Data

```
localStorage.userName= "Peter";
localStorage["name"] = "Zoro";
localStorage.setItem("name", "Zoro");
```

Get Data

```
localStorage.userName;
localStorage["name"];
localStorage.getItem("name");
```

Delete Data

```
delete localStorage["name"];
localStorage.removeItem("name");
```

Clear

```
localStorage.clear();
```

JSON

```
JSON.stringify(true);//"true"
JSON.parse("[1,2,3]");//[1,2,3]
```

Iterating

```
for (var i=0; i < localStorage.length; i++){
  var propertyName = localStorage.key(i);
  console.log(localStorage.getItem(propertyName));
}
```