JavaScript: using the local storage

The local storage is a powerful feature of the web storage which allows us to save data on the user's browser as strings organized as key/value pairs. The main object is `localStorage` and the storage persists even after the end of a browser's session.

Reading and saving data

To save data we use the `setItem()` method which accepts two parameters: the name of the key and the data string:

```
localStorage.setItem('test', '1');
```

Now we have the key `test` and a numeric string. We can read its value with the `getItem()` method:

```
console.log(localStorage.getItem('test')); // '1'
```

Deleting keys

To delete a key we can use the `removeItem()` method:

```
localStorage.removeItem('test');
```

```
console.log(localStorage.getItem('test')); // null
```

Erasing all data

To completely remove all data from the storage we use the `clear()` method:

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

Getting the name of a key

To get the name of a key we use the `key()` method:

```
console.log(localStorage.key('test')); // 'test'
```

Getting the total number of all keys

We can know how many keys are contained within our storage by using the `length` property of the `localStorage` object:

```
console.log(localStorage.length); // 1
```

Storing JavaScript objects

By default all values are treated as strings. To store objects we need to use the `JSON.stringify()` and `JSON.parse()` methods:

```
localStorage.setItem('test', JSON.stringify({name:
'value'}));
console.log(JSON.parse(localStorage.getItem('test')).
name); // 'value'
```

Conclusions

The local storage has currently two limits: the lack of support in older versions of Internet Explorer and the maximum size allowed for the storage, which varies from browser to browsers and it's about from 5 to 10 Megabytes, depending on the browser.