



Using LocalStorage

The idea behind [local storage](#) (also known as web storage) is that we can save key/value pairs of data in the browser for later retrieval and use.

A good example of where this may be useful is to set basic settings of an application and persist them between visits to the site.

The feature is supported by [all browsers except Opera Mini](#)

We'll look at setting and retrieving values to local storage and how we can use them in making an application configuration consistent across visits to the page.

How does it work?

Although the API is widely supported we should still code defensively and provide feature detection to make sure that it works before we use it.

Feature Detection

To make sure that we support the API we check that the localStorage is available in the window object. If it's not available we bail early and if it is supported then we continue working.

```
if (!'localStorage' in window) {  
  console.log('Local Storage Not Supported');  
  return;  
} else {  
  console.log('Local Storage Works!');  
}
```

We can further abstract the test into a function and use that function to test if the API is supported. Then it follows the same process as the

```
function supportsLS(){  
  return 'localStorage' in window;
```

```
}  
  
if (supportsLS()) {  
  console.log('yay!');  
} else {  
  console.log('boo!!!');  
}
```

Now that we know the feature is supported we can move forward with the rest of the code.

Setting values

at it's simplest use setting a value for Local Storage is just a matter of calling `localStorage.setItem` with the key/value pair we want to store as the parameters.

```
localStorage.setItem('firstName', 'Marty');
```

Retrieving values

Let's say that we did our setup, added the data that we wanted to save and now we need to retrieve it on subsequent visits. All we need to do is to call `localStorage.getItem` and pass it the name of the key we want to retrieve.

```
localStorage.getItem('firstName');
```

We can then attach the value of the item elsewhere on the script to make it do what we want to.

Clearing everything up

There are times when we want to reset all the settings we have added using `setItem()`. We could set them to a null value but it would be cumbersome if we have too many values.

The other alternative is to use `localStorage.clear()`

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

Basic Example

The example will take two values we enter in a form, first name and last name and add them to local storage.

The form is a standard form without the action field (we don't need it so why have it) a submit and a reset button.

```
<form id="demo1">
  <fieldset>
    <legend>Enter Your Name</legend>

    <label for="firstName">First Name</label>
    <input type="text" id="firstName" />

    <label for="lastName">Last Name</label>
    <input type="text" id="lastName" />

    <button type="submit" id="submit">Tell me!</button>
    <button type="reset" id="reset">Reset</button>
  </fieldset>
</form>
```

Javascript is where the magic happens.

First we create for constants to capture the different elements of the form.

Then we create an `onSubmit` event for the form, prevent defaults and store first and last names in the browser's local storage.

Finally we create an `onClick` event for the reset button, then we prevent the default action and, finally, we clear local storage altogether.

```
const myForm = document.querySelector('#demo1');
```

```
const firstName = document.querySelector('#firstName');
const lastName = document.querySelector('#lastName');
const reset = document.querySelector('#reset');

myForm.onSubmit = function(e) {
  e.preventDefault();

  window.localStorage.setItem('firstName', firstName.value);
  window.localStorage.setItem('lastName', lastName.value);
}

reset.onclick = function(e) {
  e.preventDefault();
  window.localStorage.clear();
}
```

Drawbacks

There are some things to keep in mind when working with Web/Local storage.

- It only works with strings. All other values are converted to strings before being stored.
- It overwrites values with the last one you entered. This is why the usefulness of LocalStorage is limited to string-based documents.