

Advanced JavaScript for Web Sites and Web Applications

Local storage

Web Storage

- **localStorage** and **sessionStorage** are part of the *Web storage* API.
- Both allow you to store data in the user's browser
- Note, *Web storage* is sometimes referred to as *DOM storage*

Web Storage

- `localStorage` is for storing *persistent* data
 - Data is stored permanently, even after browser has been closed
- `sessionStorage` is for storing *temporary* data
 - Data is deleted when the browser is closed.

The same-origin policy

- `localStorage` and `sessionStorage` are both subject to the *same-origin policy*
 - Just like AJAX.
- This means you cannot read data stored by scripts running on different domains to your script.

The same-origin policy

- User visits *abc.com*, which stores some data in his/her browser
- User then visits *xyz.com*, which can store its own data in the browser, but cannot interact with data set by *abc.com*

localStorage VS sessionStorage

- We will just use `localStorage` in this session, but the same principles and techniques apply to `sessionStorage`.
 - The only difference is the amount of time the data is stored for.

The localStorage object

- localStorage is a property of the window object.
- localStorage is an object
- To access/interact with it:

```
window.localStorage
```

How the data is stored

- We store our data in the object as key/value pairs, similar to regular object properties
- The *keys* can be strings or integers
- The *values* can only be strings
- The amount of data we can store varies from browser to browser
 - 5-10mb in modern browsers, but may be restricted by user configuration

Storing data: `setItem()`

- You can add an item to the storage cache with the storage object's `setItem()` method
- It takes 2 arguments, the *key* and the *value* you want to store against it

```
// The "key" is "name", the "value" is "Joe"  
window.localStorage.setItem('name', 'Joe');
```

Removing data: `removeItem()`

- You can remove an item from the browser's storage with the object's `removeItem()` method
- We pass it the *key* of the item we want to remove

```
// Delete item with the key: "name"  
window.localStorage.removeItem('name');
```

Updating data: `setItem()`

- To update/replace the value stored in an item, use `setItem()` with the same *key* and a new *value* (overwrites stored value)

```
// Store "Joe" under "name" key  
window.localStorage.setItem('name', 'Joe');
```

```
// Now change the stored value to "Bill"  
window.localStorage.setItem('name', 'Bill');
```

Retrieving data: getItem()

- You can retrieve a stored value with the object's `getItem()` method
- We pass it the *key* for the item we want to retrieve, and it will return the stored value:

```
var data = window.localStorage.getItem('name');
```

- If there is no item associated with the *key* we pass to `getItem`, **null** is returned

Storing data: Object syntax

- It is also possible to use regular object syntax to interact with localStorage

```
// These all do the same thing  
window.localStorage.setItem('name', 'Joe');  
window.localStorage.name = 'Joe';  
window.localStorage['name'] = 'Joe';
```

- But it is better to use the getItem and setItem methods, as these are more likely to work consistently in all browsers.

Complex values

- If you want to store more complex values (arrays, objects), convert them to JSON (which is a string) and store that instead:

```
myObj = {  
  firstname: "Joe",  
  lastname: "Bloggs"  
}  
myJson = JSON.stringify(myObj);  
localStorage.setItem('storedObj', myJson);
```

Other storage object properties/methods

- The `length` property holds the number of items stored for current domain:

```
window.localStorage.length;
```

- The `clear()` method removes all stored data for current domain:

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

Web storage and the law

- In the UK and across the EU, a law was introduced in 2011 governing the client-side storage of data.
- For UK web sites, the ICO guidelines state that those storing data on a user's device must:
 - tell people that they are storing the data
 - explain why they are storing the data
 - obtain the user's consent to store data (explicit or implied)
- [More info](#)

Exercise

- Download the exercises document from Moodle and do *Exercise 1*