CST 2120: Web Applications and Databases

David Gamez

Browser Storage

Lecture Overview

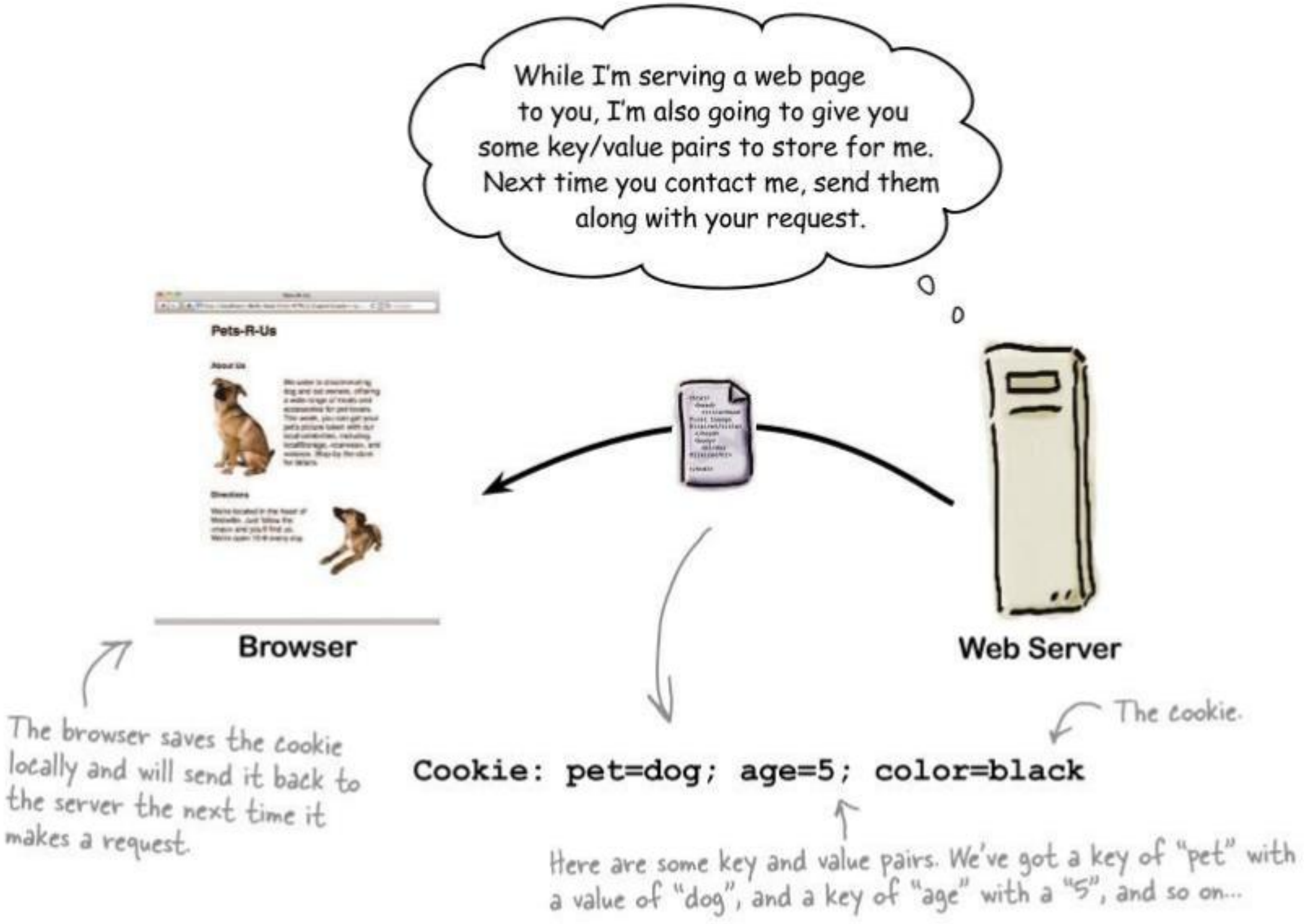
- Cookies.
- HTML5 local storage.

Cookies

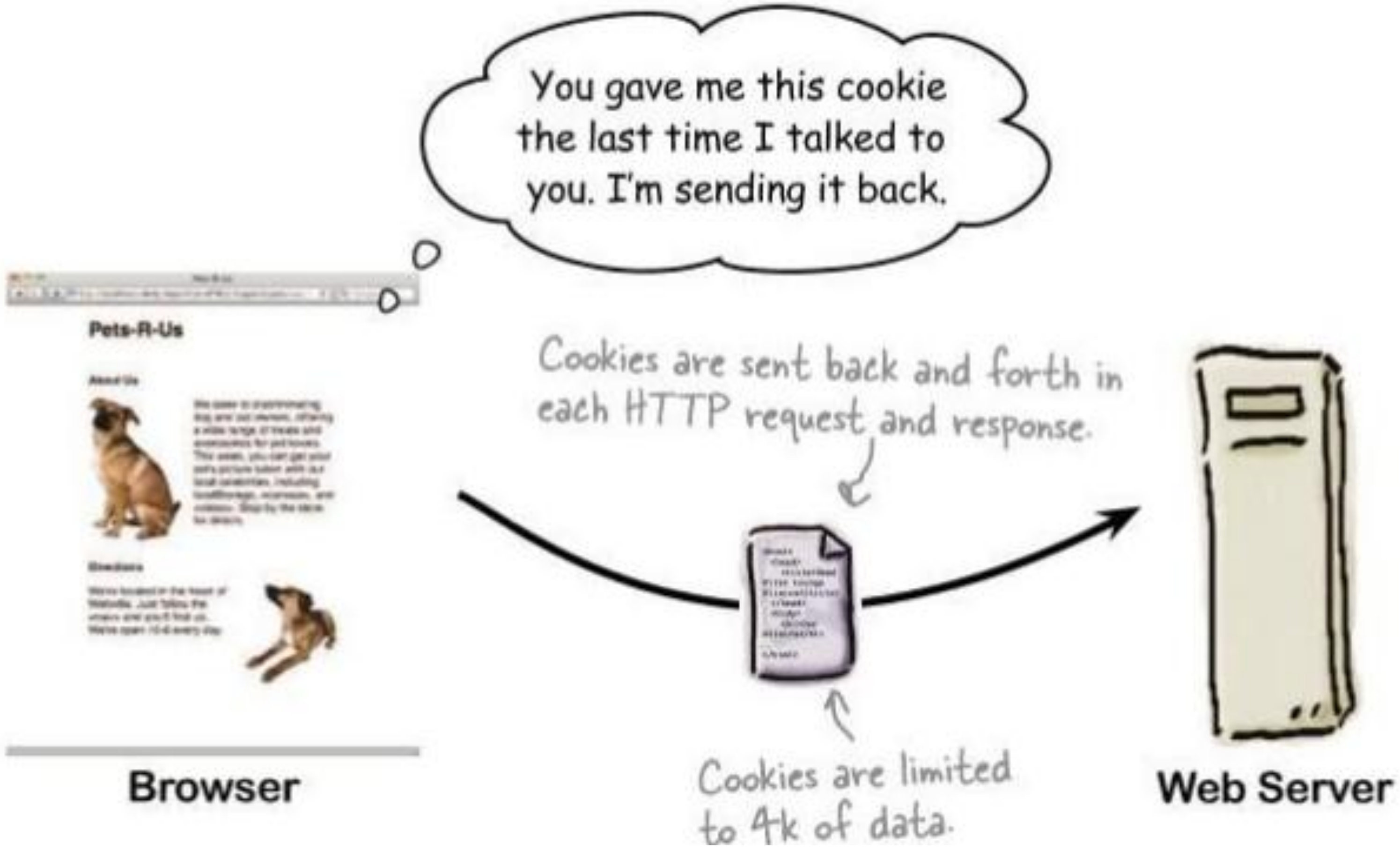
Cookies

- Data stored in small text files on user’s computer.
- Invented to solve the problem “how to remember information about the user.”
- For example, when user visits web page their name could be stored in a cookie.
- When browser requests a web page from server, cookies belonging to the page are added to the request.
- Next time user visits page, the cookie can be used to remember his or her name.

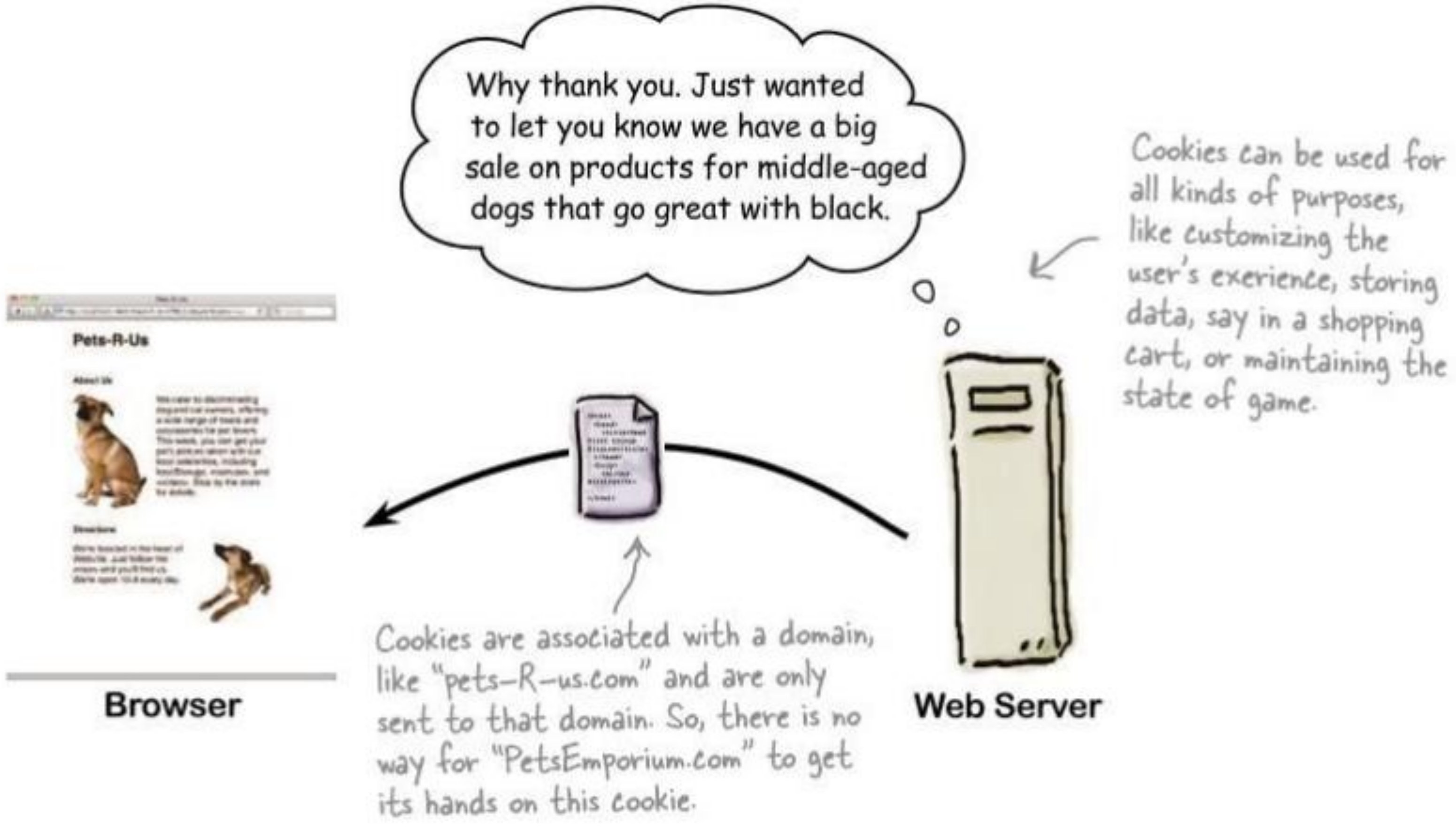
Cookies


While I'm serving a web page to you, I'm also going to give you some key/value pairs to store for me. Next time you contact me, send them along with your request.
Pets-R-Us
Browser
Web Server
The cookie.
The browser saves the cookie locally and will send it back to the server the next time it makes a request.
Cookie: pet=dog; age=5; color=black
Here are some key and value pairs. We've got a key of "pet" with a value of "dog", and a key of "age" with a "5", and so on...

Cookies


You gave me this cookie the last time I talked to you. I'm sending it back.
Pets-R-Us
Cookies are sent back and forth in each HTTP request and response.
Cookies are limited to 4k of data.
Browser
Web Server

Cookies


Why thank you. Just wanted to let you know we have a big sale on products for middle-aged dogs that go great with black.
Cookies can be used for all kinds of purposes, like customizing the user's exerience, storing data, say in a shopping cart, or maintaining the state of game.
Pets-R-Us
Browser
Web Server
Cookies are associated with a domain, like "pets-R-us.com" and are only sent to that domain. So, there is no way for "PetsEmporium.com" to get its hands on this cookie.

Setting Cookies

- Use document.cookie property to create, read and delete cookies.

```
document.cookie = "username=John Doe";
```

- By default the cookie is deleted when the browser is closed.
- You can also set an expiry date (in UTC time).

```
document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC";
```

- By default cookie belongs to the current page.
- Can also set a path parameter.

```
document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";
```

Reading, Changing and Deleting Cookies

```
let cookies = document.cookie;
```

- This returns all cookies in one string.
- Change a cookie in the same way that you created it:

```
document.cookie = "username=John Smith; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";
```

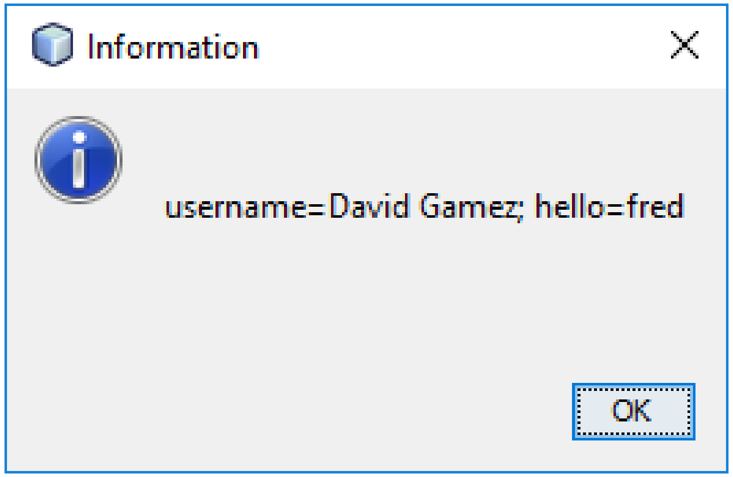
- Delete a cookie by setting the expires parameter to a date in the past.

```
document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC";
```

Cookies are Appended to Each Other

- If you set a cookie with a new key, the older cookies are not overwritten.
- New cookie is added to document.cookie.

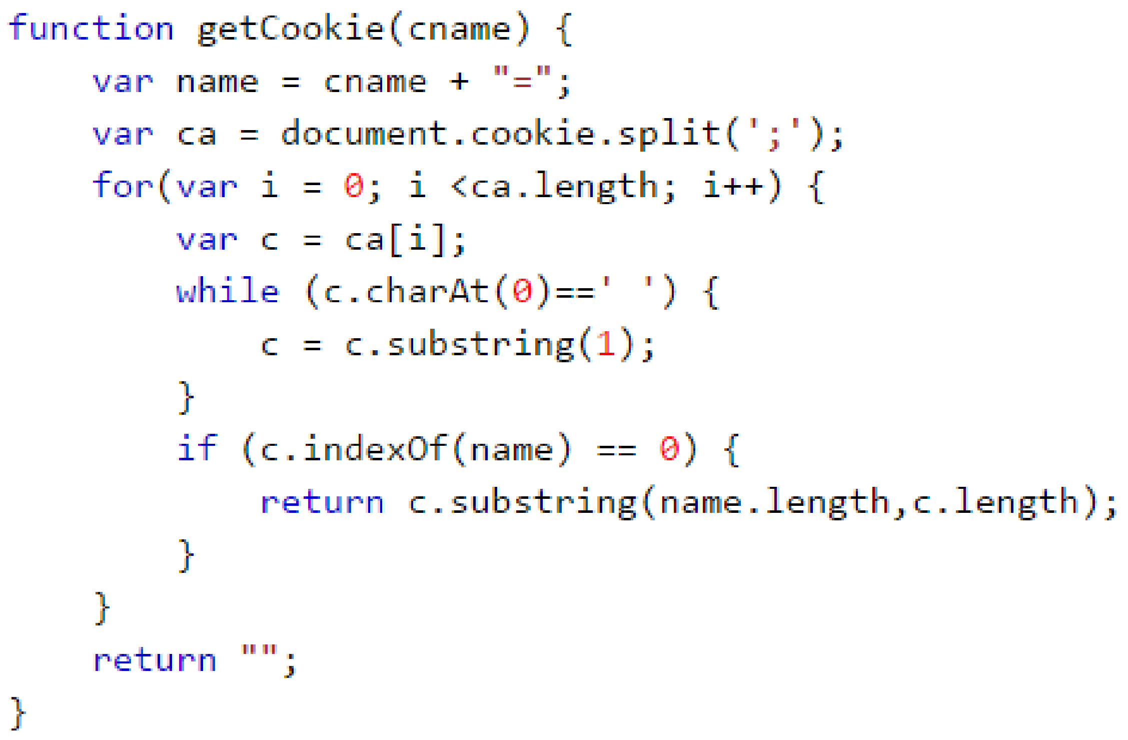
```
document.cookie = "username=David Gamez; expires=Thu, 01 Jan 2017 00:00:00 UTC";
document.cookie = "hello=fred";
alert(document.cookie);
```



username=David Gamez; hello=fred

Disadvantages of Cookies

- Storage limited to 4KB.
- Messy string processing to extract the value of a particular variable.



```
function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for(var i = 0; i <ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') {
            c = c.substring(1);
        }
        if (c.indexOf(name) == 0) {
            return c.substring(name.length,c.length);
        }
    }
    return "";
}
```

HTML5 Local Storage

HTML5 Local Storage

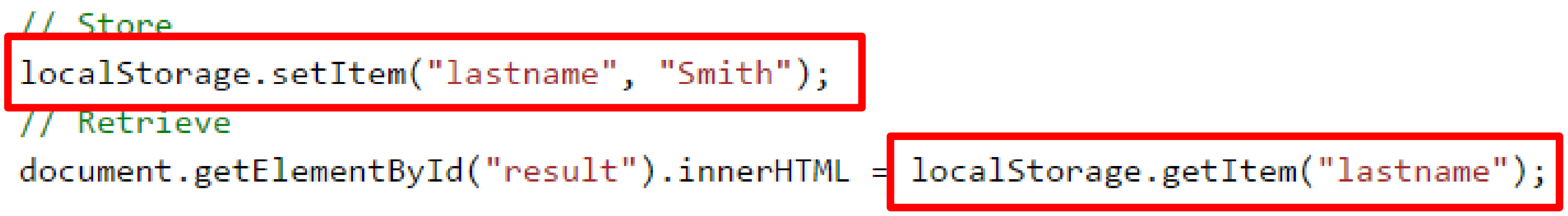
- Simple way of storing key-value pairs in browser using JavaScript.
- Storage is specific to each domain.
- Storage can be persistent – remains after you quit the browser.
- 5-10MB of storage available.

Storage Types

- Local storage:
  - Stores data with no expiration date.
- Session storage:
  - Stores data for one session.
  - Data is deleted when browser tab is closed.

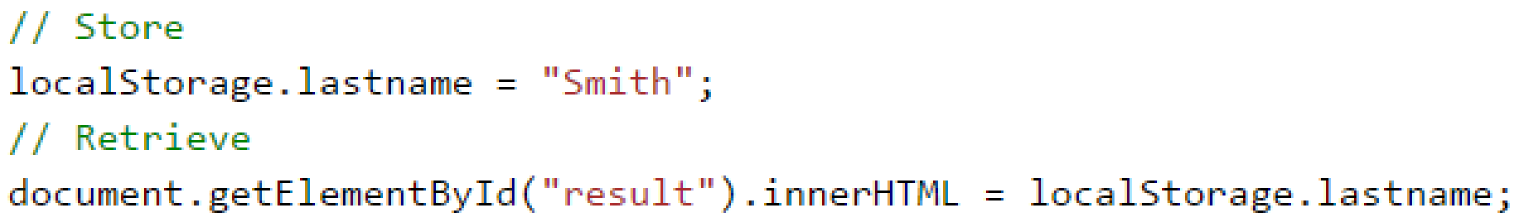
Using HTML Local Storage

- With setItem(…) and getItem(…) functions:



```
// Store
localStorage.setItem("lastname", "Smith");
// Retrieve
document.getElementById("result").innerHTML = localStorage.getItem("lastname");
```

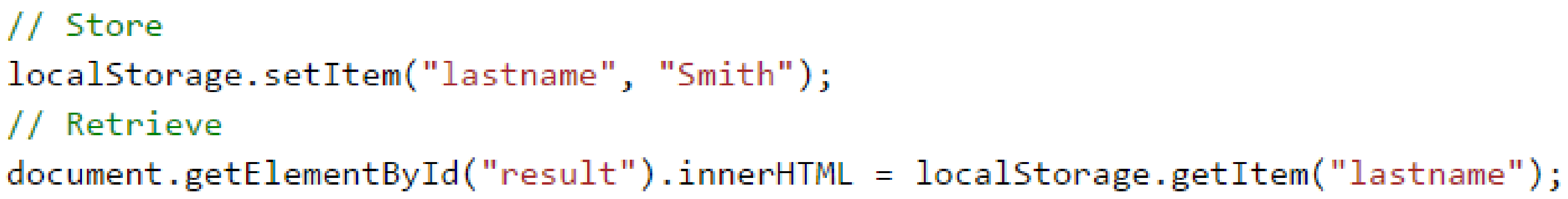
- Dot notation:



```
// Store
localStorage.lastname = "Smith";
// Retrieve
document.getElementById("result").innerHTML = localStorage.lastname;
```

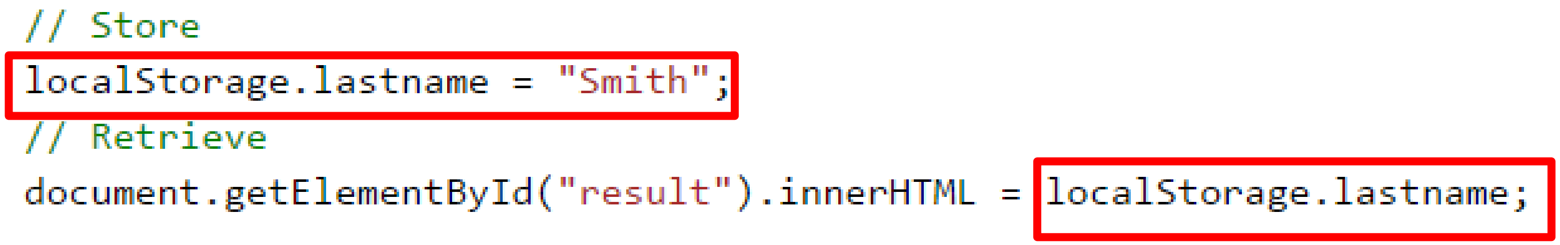
Using HTML Local Storage

- With setItem(…) and getItem(…) functions:



```
// Store
localStorage.setItem("lastname", "Smith");
// Retrieve
document.getElementById("result").innerHTML = localStorage.getItem("lastname");
```

- Dot notation:



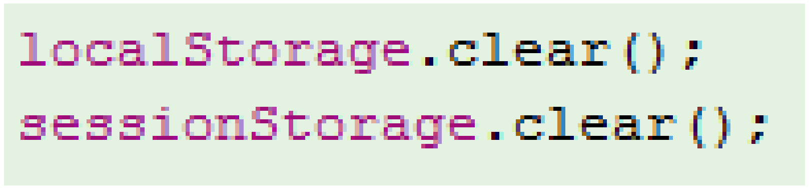
```
// Store
localStorage.lastname = "Smith";
// Retrieve
document.getElementById("result").innerHTML = localStorage.lastname;
```

Session Storage

- sessionStorage stores data for one session.
- Data is lost when the browser tab is closed.
- For example:
  - sessionStorage.setItem(“key”, “value”);
  - sessionStorage.key = “value”;

Clearing Local Storage

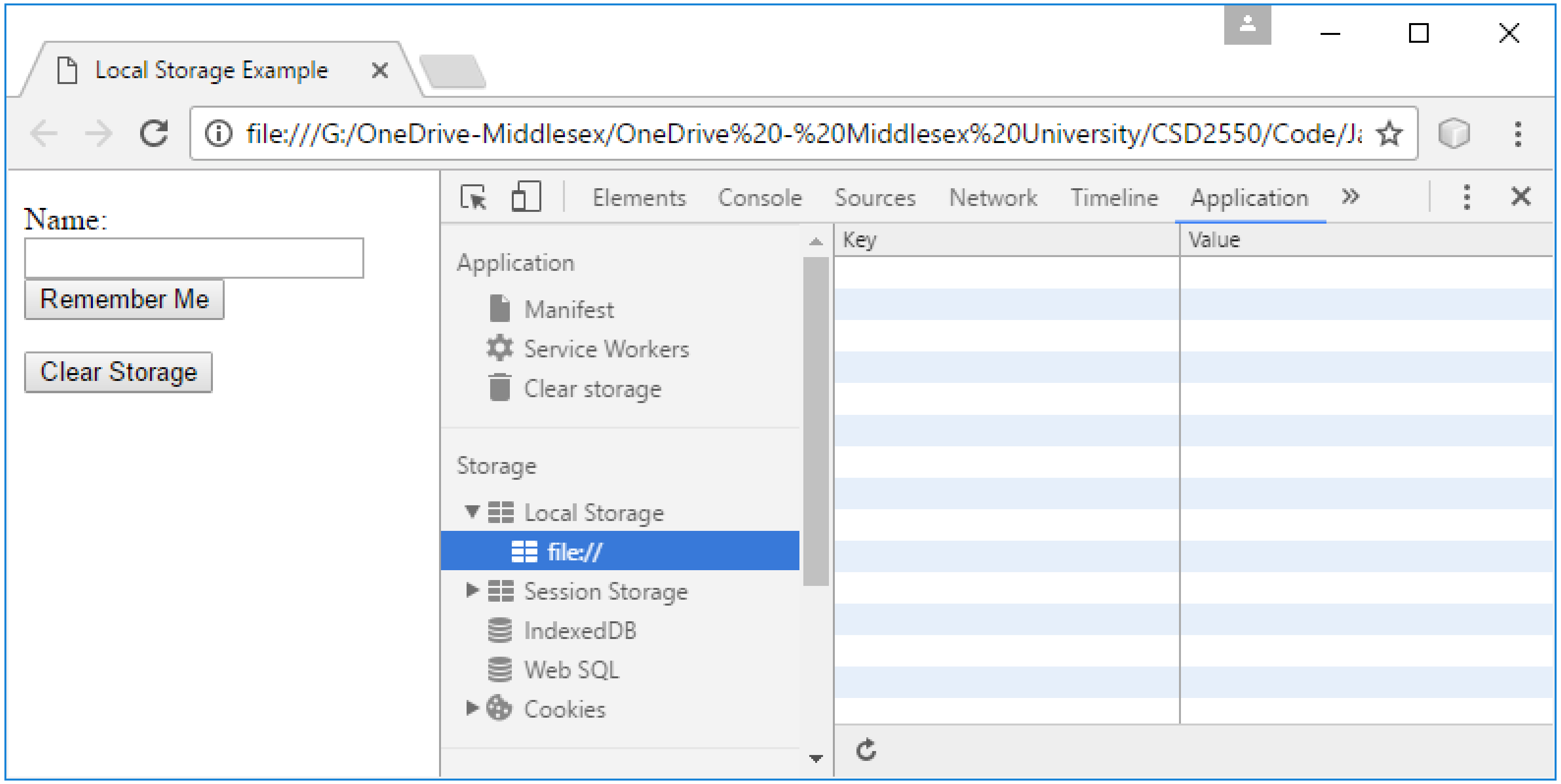
- Local and session storage have a clear function that deletes all of the data for the domain.



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

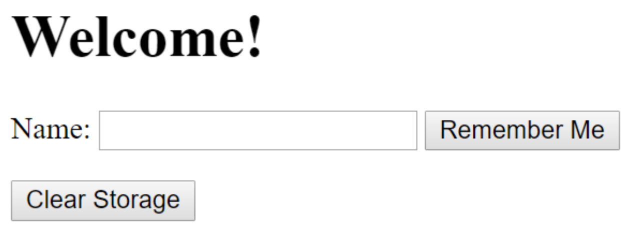
Viewing Data in Local Storage

- Can use Developer Tools to view, delete and edit values in local and session storage.



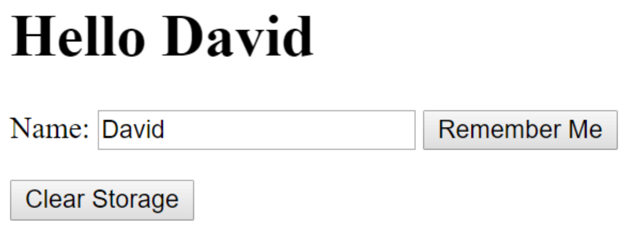
Example

```
<body onload="showName()">
    <h1 id="Header">Welcome!</h1>
    <p id="inputPara">
        Name: <input type="text" id="nameInput">
        <button onclick="storeName()">Remember Me</button>
    </p>
    <p>
        <button onclick="clearStorage()">Clear Storage</button>
    </p>
    <script>
        let header = document.getElementById("Header");

        //Displays the user's name if it has been set.
        function showName(){
            if(localStorage.usrName != undefined)
                header.innerHTML = "Hello " + localStorage.usrName;
        }
        //Stores name when user clicks button
        function storeName(){
            let name = document.getElementById("nameInput").value;
            header.innerHTML = "Hello " + name;
            localStorage.usrName = name;//Store name
       }
       //Clears local storage
       function clearStorage(){
           localStorage.clear();
           header.innerHTML = "Welcome!";
       }
    </script>
</body>
```



Welcome!

Name:



Hello David

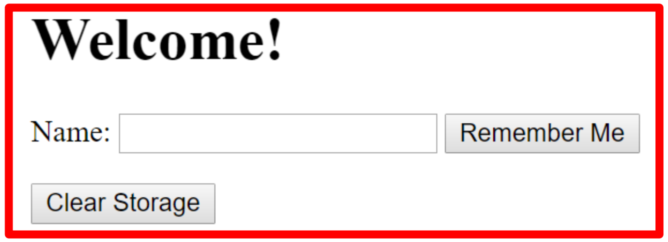
Name: David

Demo

- local-storage-example.html

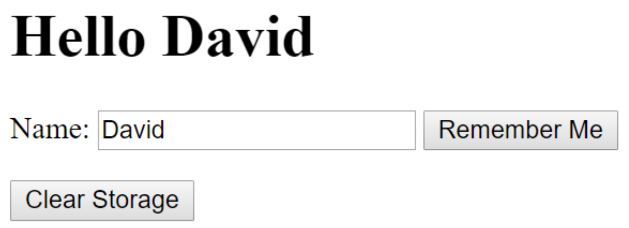
Example

```
<body onload="showName()">
    <h1 id="Header">Welcome!</h1>
    <p id="inputPara">
        Name: <input type="text" id="nameInput">
        <button onclick="storeName()">Remember Me</button>
    </p>
    <p>
        <button onclick="clearStorage()">Clear Storage</button>
    </p>
    <script>
        let header = document.getElementById("Header");

        //Displays the user's name if it has been set.
        function showName(){
            if(localStorage.usrName != undefined)
                header.innerHTML = "Hello " + localStorage.usrName;
        }
        //Stores name when user clicks button
        function storeName(){
            let name = document.getElementById("nameInput").value;
            header.innerHTML = "Hello " + name;
            localStorage.usrName = name;//Store name
        }
        //Clears local storage
        function clearStorage(){
            localStorage.clear();
            header.innerHTML = "Welcome!";
        }
    </script>
</body>
```



Welcome!

Name:

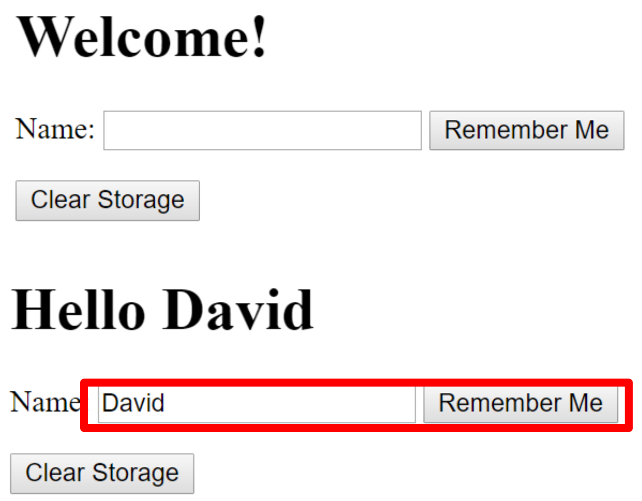


Hello David

Name: David

Example

```html
<body onload="showName()">
    <h1 id="Header">Welcome!</h1>
    <p id="inputPara">
        Name: <input type="text" id="nameInput">
        <button onclick="storeName()">Remember Me</button>
    </p>
    <p>
        <button onclick="clearStorage()">Clear Storage</button>
    </p>
    <script>
        let header = document.getElementById("Header");

        //Displays the user's name if it has been set.
        function showName(){
            if(localStorage.usrName != undefined)
                header.innerHTML = "Hello " + localStorage.usrName;
        }
        //Stores name when user clicks button
        function storeName(){
            let name = document.getElementById("nameInput").value;
            header.innerHTML = "Hello " + name;
            localStorage.usrName = name;//Store name
       }
       //Clears local storage
       function clearStorage(){
           localStorage.clear();
           header.innerHTML = "Welcome!";
       }
    </script>
</body>
```



Welcome!

Name: [ ] Remember Me

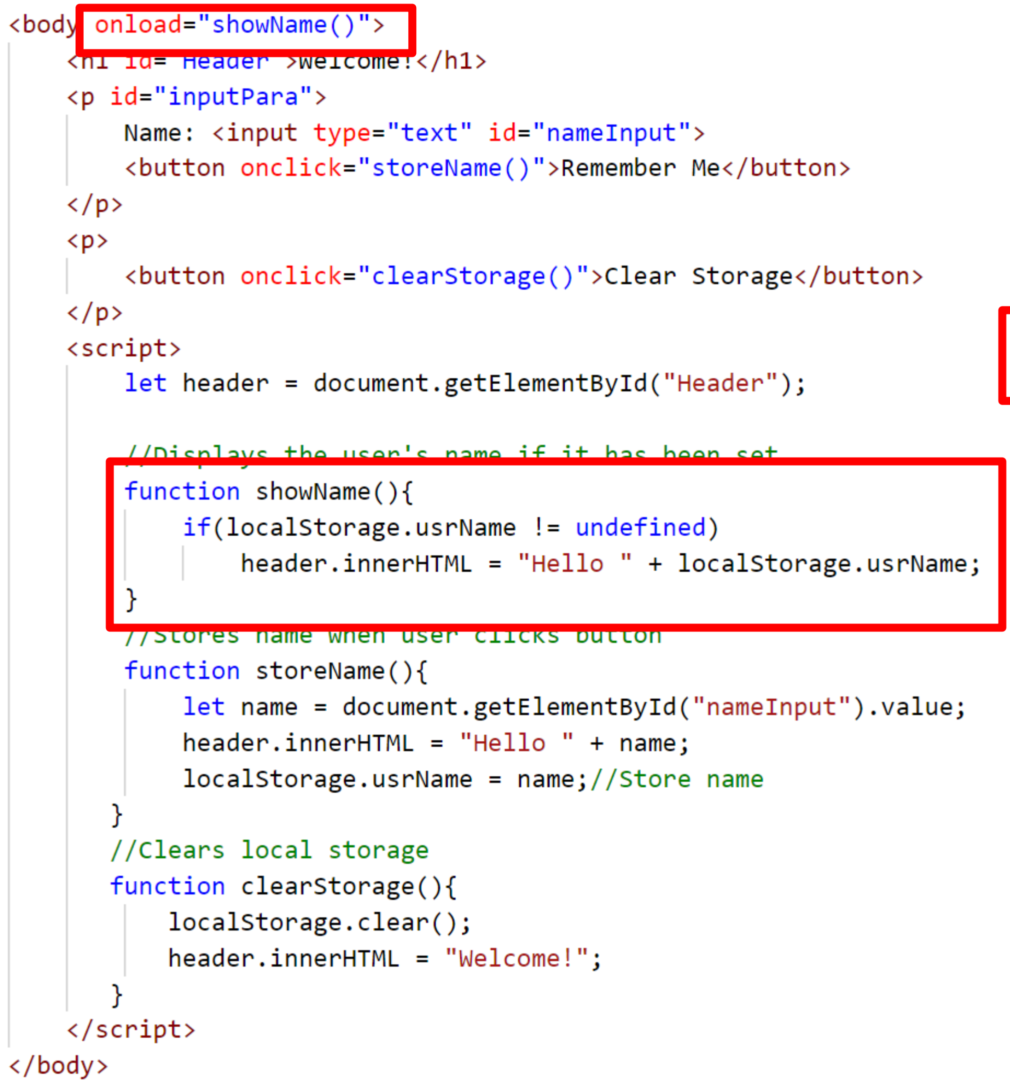
Clear Storage

Hello David

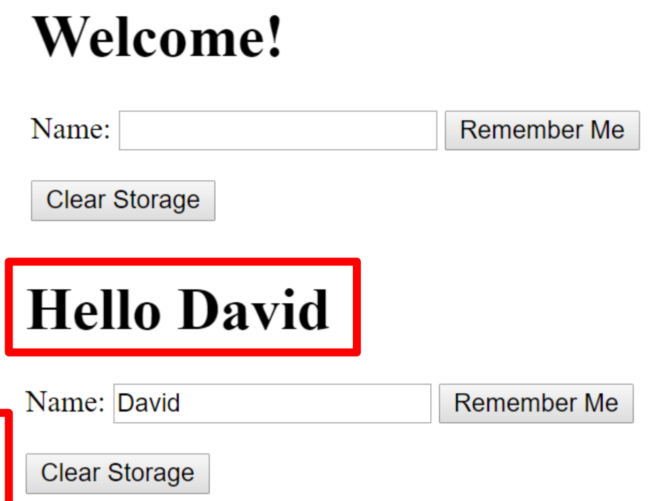
Name: David Remember Me

Clear Storage

Example



```
<body onload="showName()">
    <h1 id="Header">Welcome!</h1>
    <p id="inputPara">
        Name: <input type="text" id="nameInput">
        <button onclick="storeName()">Remember Me</button>
    </p>
    <p>
        <button onclick="clearStorage()">Clear Storage</button>
    </p>
    <script>
        let header = document.getElementById("Header");

        //Displays the user's name if it has been set
        function showName(){
            if(localStorage.usrName != undefined)
                header.innerHTML = "Hello " + localStorage.usrName;
        }
        //Stores name when user clicks button
        function storeName(){
            let name = document.getElementById("nameInput").value;
            header.innerHTML = "Hello " + name;
            localStorage.usrName = name;//Store name
        }
       //Clears local storage
       function clearStorage(){
           localStorage.clear();
           header.innerHTML = "Welcome!";
       }
    </script>
</body>
```



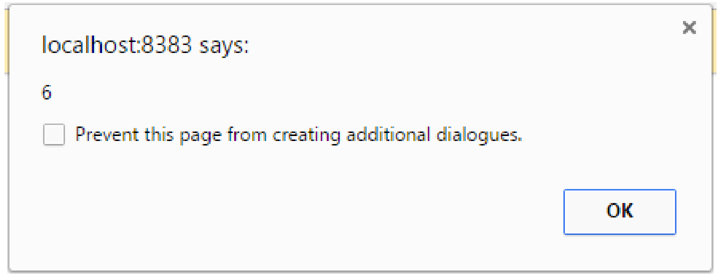
Difference between Cookies and Local Storage

- Cookies are exchanged every time the client interacts with the server.
- The data in HTML local storage can only be accessed locally by JavaScript.
- The server only receives locally stored data if it is explicitly sent to it – for example, using AJAX.
- Different amounts of data can be stored:
  - Cookie: 4KB
  - HTML local storage: 5MB.

Data in Local and Session Storage

- **Keys and values are strings.**
- If you store numbers, the browser converts to strings when they are stored and converts them back to numbers etc. depending on context.

```
localStorage.number = 2;
alert(localStorage.number * 3);
```



localhost:8383 says:

6

Prevent this page from creating additional dialogues.

- Booleans seem to work less well. Safer using ===:

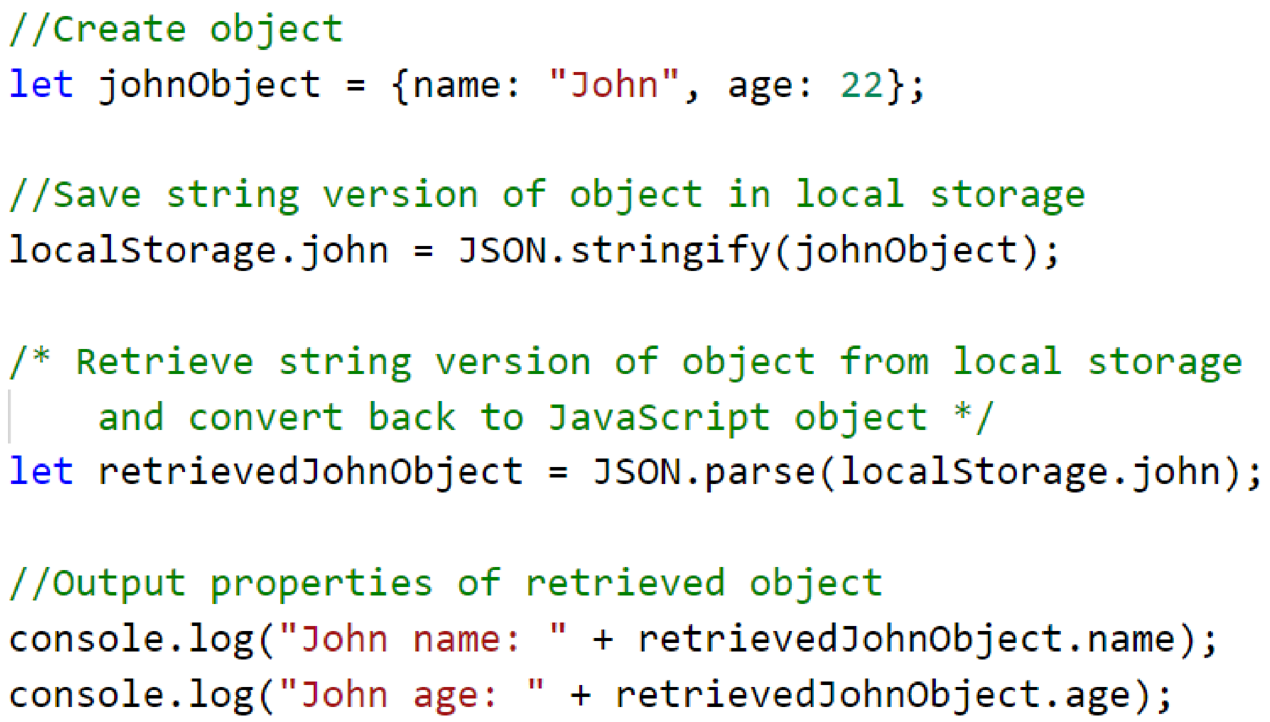
```
localStorage.bool = false;
if(localStorage.bool === "false")
    alert("Boolean is false");
```

Storing and Retrieving Arrays and Objects

- Can store arrays and objects by converting them to strings using JSON.stringify(...);
- When you retrieve them, convert them back to objects and arrays using JSON.parse(...);

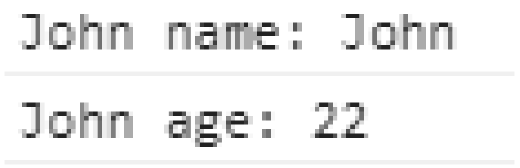
Storing and Retrieving Arrays and Objects

- Example:



```javascript
//Create object
let johnObject = {name: "John", age: 22};

//Save string version of object in local storage
localStorage.john = JSON.stringify(johnObject);

/* Retrieve string version of object from local storage
    and convert back to JavaScript object */
let retrievedJohnObject = JSON.parse(localStorage.john);

//Output properties of retrieved object
console.log("John name: " + retrievedJohnObject.name);
console.log("John age: " + retrievedJohnObject.age);
```

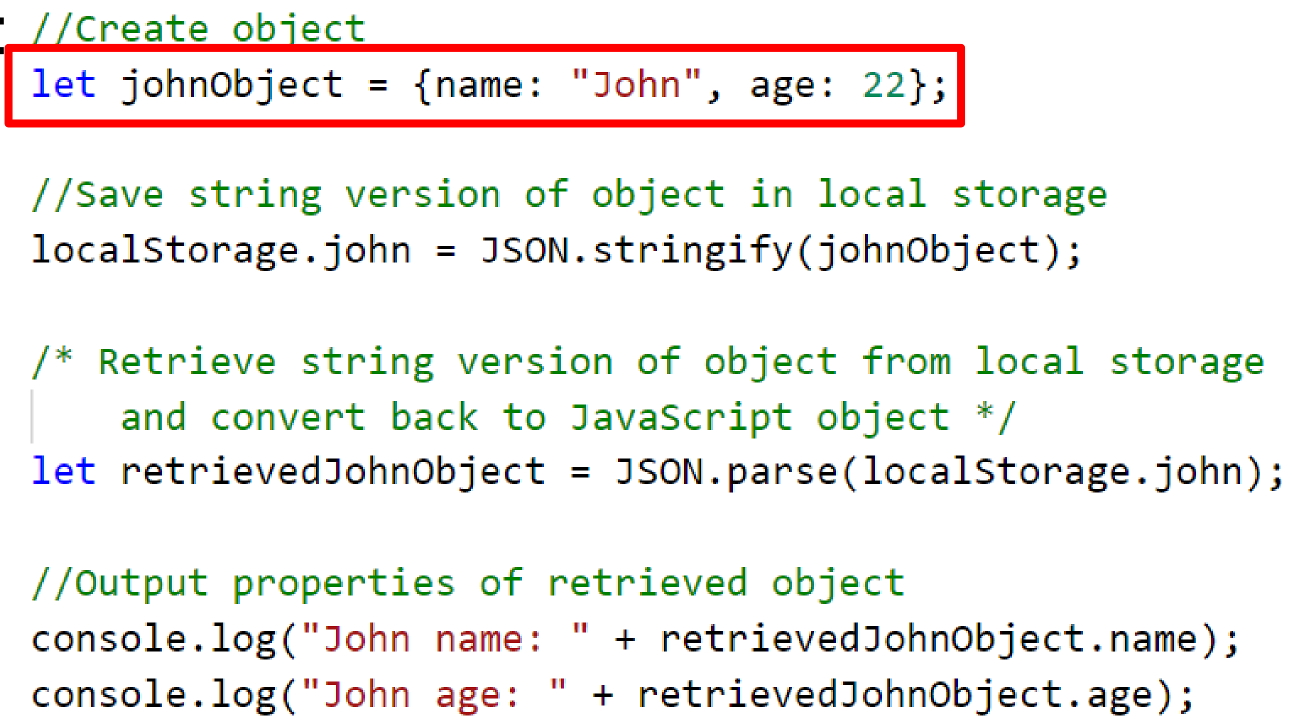
- Output:



```
John name: John
John age: 22
```

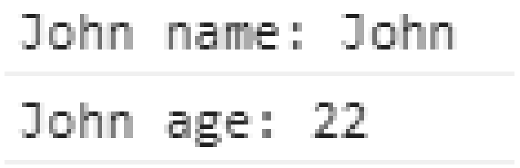
Storing and Retrieving Arrays and Objects

- Example:



```
//Create object
let johnObject = {name: "John", age: 22};

//Save string version of object in local storage
localStorage.john = JSON.stringify(johnObject);

/* Retrieve string version of object from local storage
    and convert back to JavaScript object */
let retrievedJohnObject = JSON.parse(localStorage.john);

//Output properties of retrieved object
console.log("John name: " + retrievedJohnObject.name);
console.log("John age: " + retrievedJohnObject.age);
```

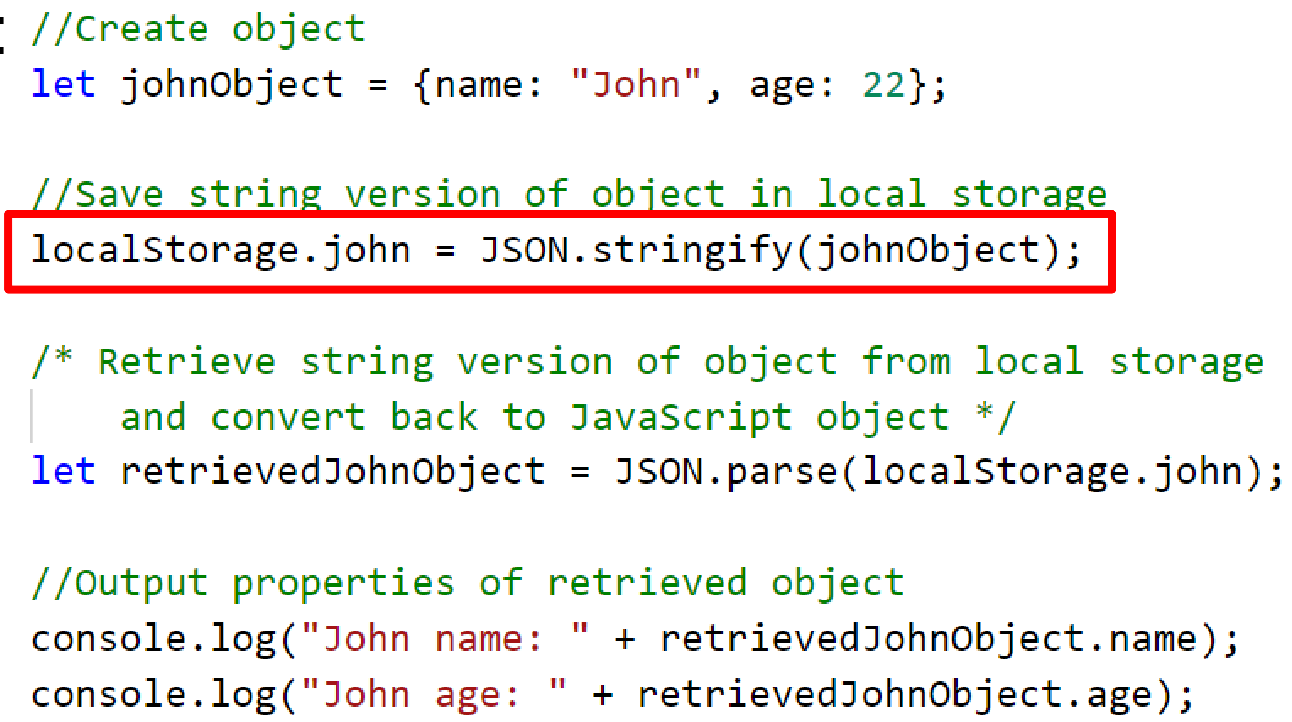
- Output:



```
John name: John
John age: 22
```

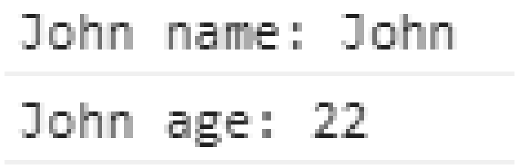
Storing and Retrieving Arrays and Objects

- Example:



```
//Create object
let johnObject = {name: "John", age: 22};

//Save string version of object in local storage
localStorage.john = JSON.stringify(johnObject);

/* Retrieve string version of object from local storage
    and convert back to JavaScript object */
let retrievedJohnObject = JSON.parse(localStorage.john);

//Output properties of retrieved object
console.log("John name: " + retrievedJohnObject.name);
console.log("John age: " + retrievedJohnObject.age);
```

- Output:



```
John name: John
John age: 22
```

Storing and Retrieving Arrays and Objects

- Example:

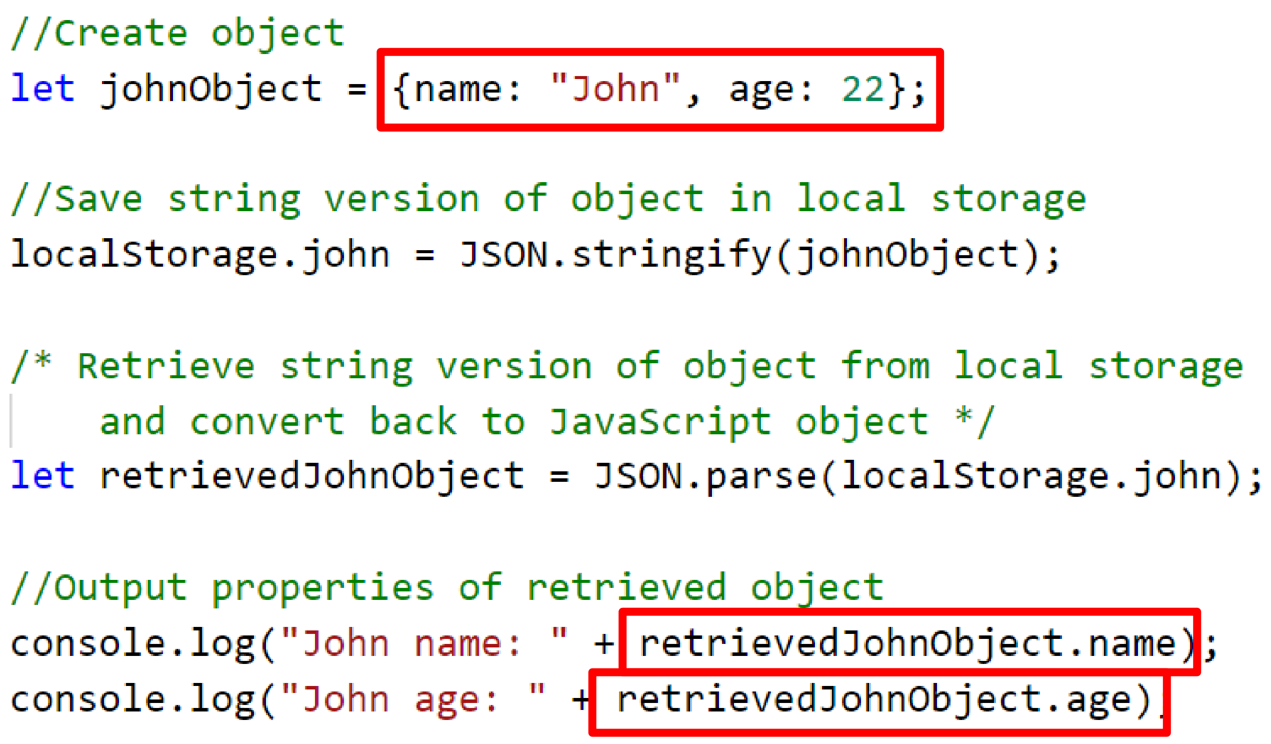
```
//Create object
let johnObject = {name: "John", age: 22};

//Save string version of object in local storage
localStorage.john = JSON.stringify(johnObject);

/* Retrieve string version of object from local storage
    and convert back to JavaScript object */
let retrievedJohnObject = JSON.parse(localStorage.john);

//Output properties of retrieved object
console.log("John name: " + retrievedJohnObject.name);
console.log("John age: " + retrievedJohnObject.age);
```

- Output:

```
John name: John
John age: 22
```

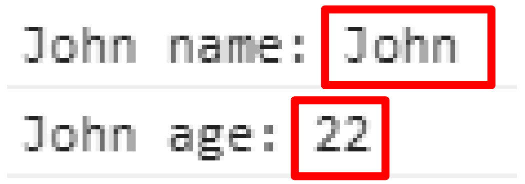
Storing and Retrieving Arrays and Objects

- Example:



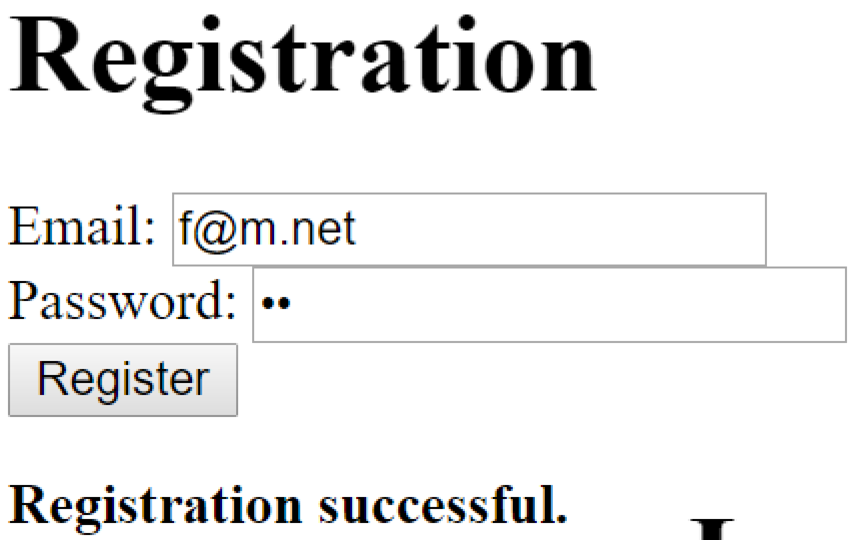
```
//Create object
let johnObject = {name: "John", age: 22};

//Save string version of object in local storage
localStorage.john = JSON.stringify(johnObject);

/* Retrieve string version of object from local storage
    and convert back to JavaScript object */
let retrievedJohnObject = JSON.parse(localStorage.john);

//Output properties of retrieved object
console.log("John name: " + retrievedJohnObject.name);
console.log("John age: " + retrievedJohnObject.age);
```

- Output:



```
John name: John
John age: 22
```

Example: Registration and Login Pages

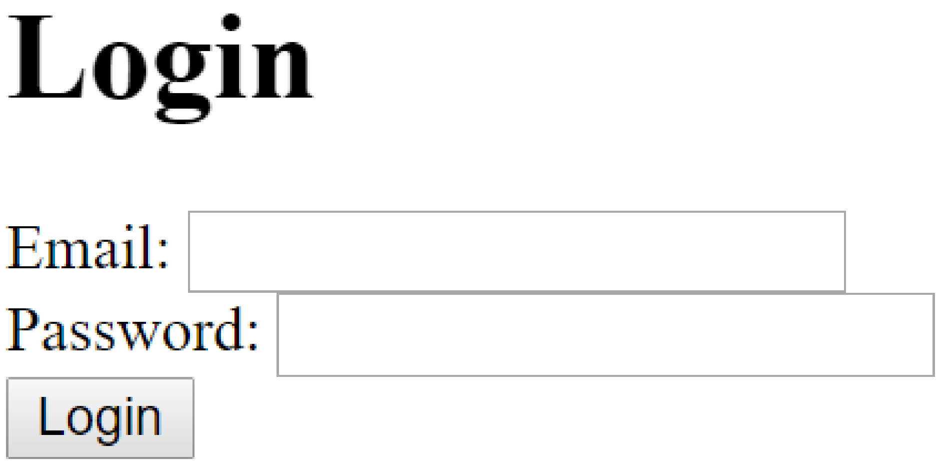


Registration

Email: f@m.net

Password: ••

Registration successful.



Login

Email:

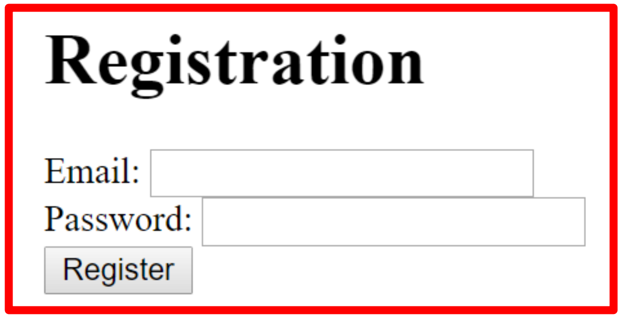
Password:

Demo

- registration.html
- login.html

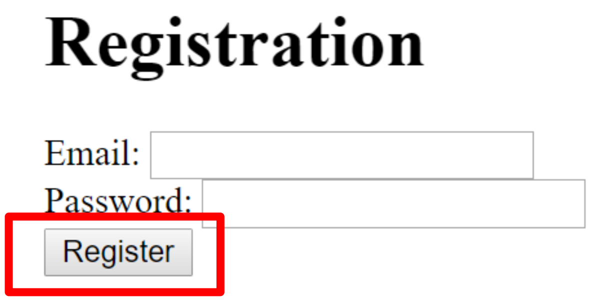
Registration

```
<h1>Registration</h1>
<p><!-- Registration input fields -->
    Email: <input type="email" id="EmailInput"><br>
    Password: <input type="password" id="PasswordInput"><br>
    <button onclick="storeUser()">Register</button>
</p>
<!-- Result of registration displayed here -->
<p id="Result"></p>

<script>
    function storeUser(){
        //Build object that we are going to store
        var usrObject = {};
        usrObject.email = document.getElementById("EmailInput").value;
        usrObject.password = document.getElementById("PasswordInput").value;

        //Store user
        localStorage[usrObject.email] = JSON.stringify(usrObject);

        //Inform user of result
        document.getElementById("Result").innerHTML = "<b>Registration successful.</b>";
    }
</script>
```

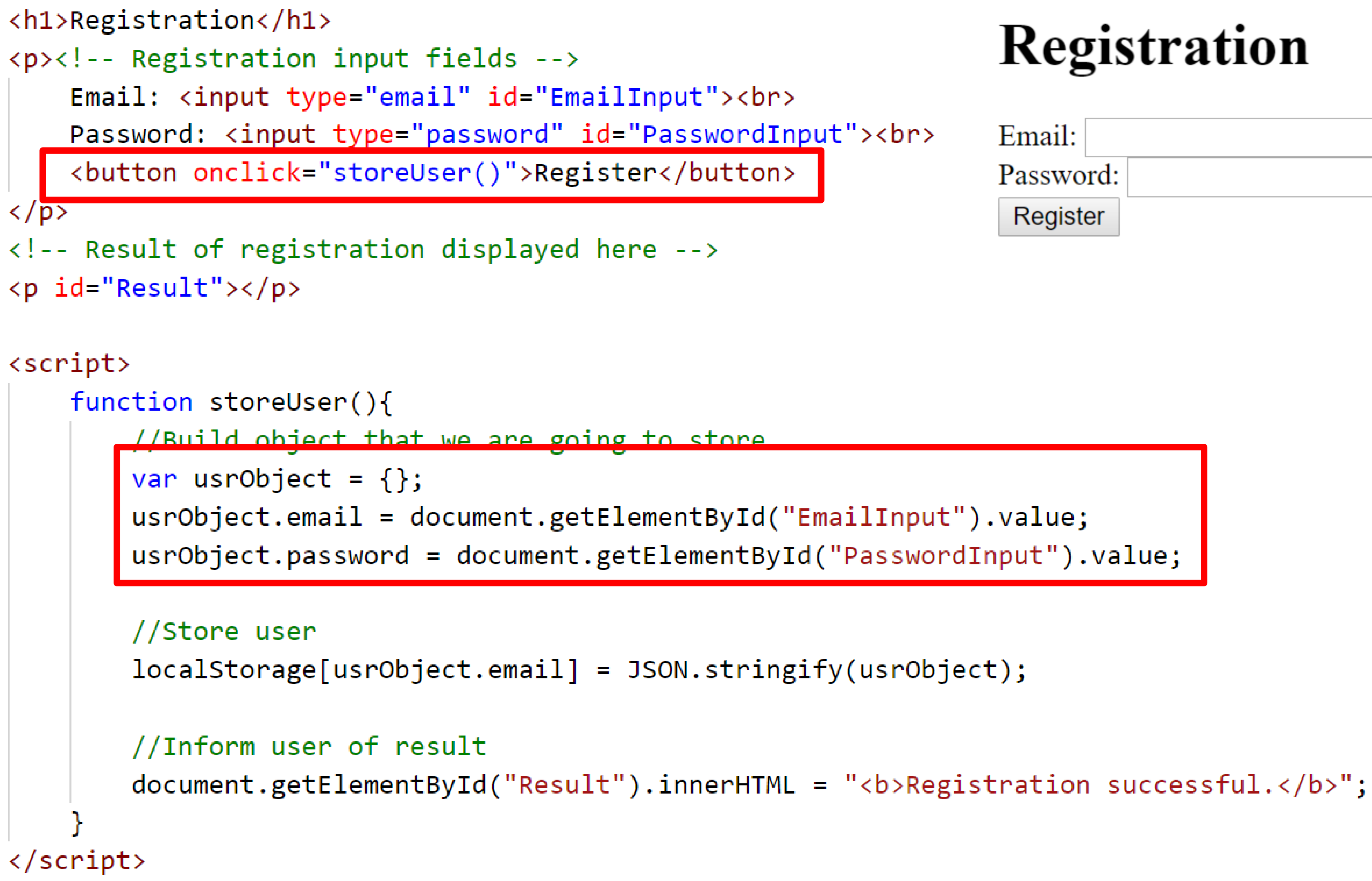


Registration

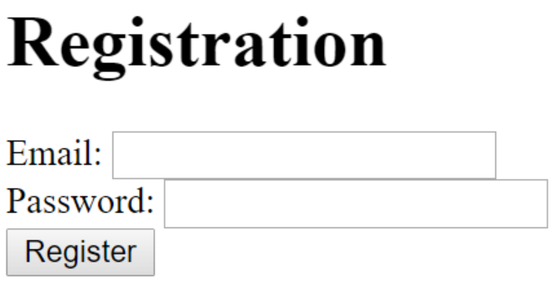
```
<h1>Registration</h1>
<p><!-- Registration input fields -->
    Email: <input type="email" id="EmailInput"><br>
    Password: <input type="password" id="PasswordInput"><br>
    <button onclick="storeUser()">Register</button>
</p>
<!-- Result of registration displayed here -->
<p id="Result"></p>

<script>
    function storeUser(){
        //Build object that we are going to store
        var usrObject = {};
        usrObject.email = document.getElementById("EmailInput").value;
        usrObject.password = document.getElementById("PasswordInput").value;

        //Store user
        localStorage[usrObject.email] = JSON.stringify(usrObject);

        //Inform user of result
        document.getElementById("Result").innerHTML = "<b>Registration successful.</b>";
    }
</script>
```



Registration



```
<h1>Registration</h1>
<p><!-- Registration input fields -->
    Email: <input type="email" id="EmailInput"><br>
    Password: <input type="password" id="PasswordInput"><br>
    <button onclick="storeUser()">Register</button>
</p>
<!-- Result of registration displayed here -->
<p id="Result"></p>

<script>
    function storeUser(){
        //Build object that we are going to store
        var usrObject = {};
        usrObject.email = document.getElementById("EmailInput").value;
        usrObject.password = document.getElementById("PasswordInput").value;

        //Store user
        localStorage[usrObject.email] = JSON.stringify(usrObject);

        //Inform user of result
        document.getElementById("Result").innerHTML = "<b>Registration successful.</b>";
    }
</script>
```



Registration

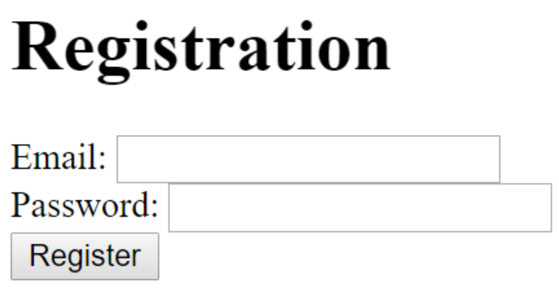
Email:

Password:

Register

Registration

```
<h1>Registration</h1>
<p><!-- Registration input fields -->
    Email: <input type="email" id="EmailInput"><br>
    Password: <input type="password" id="PasswordInput"><br>
    <button onclick="storeUser()">Register</button>
</p>
<!-- Result of registration displayed here -->
<p id="Result"></p>

<script>
    function storeUser(){
        //Build object that we are going to store
        var usrObject = {};
        usrObject.email = document.getElementById("EmailInput").value;
        usrObject.password = document.getElementById("PasswordInput").value;

        //Store user
        localStorage[usrObject.email] = JSON.stringify(usrObject);

        //Inform user of result
        document.getElementById("Result").innerHTML = "<b>Registration successful.</b>";
    }
</script>
```



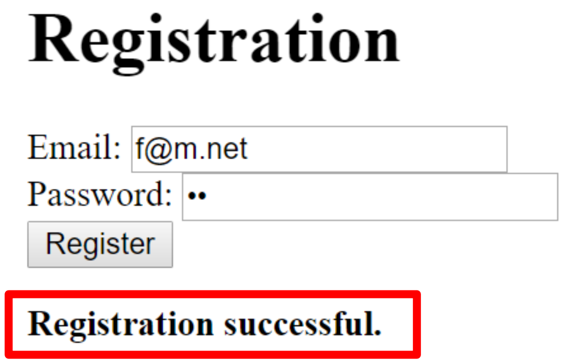
Registration

Email:

Password:

Registration

```
<h1>Registration</h1>
<p><!-- Registration input fields -->
    Email: <input type="email" id="EmailInput"><br>
    Password: <input type="password" id="PasswordInput"><br>
    <button onclick="storeUser()">Register</button>
</p>
<!-- Result of registration displayed here -->
<p id="Result"></p>

<script>
    function storeUser(){
        //Build object that we are going to store
        var usrObject = {};
        usrObject.email = document.getElementById("EmailInput").value;
        usrObject.password = document.getElementById("PasswordInput").value;

        //Store user
        localStorage[usrObject.email] = JSON.stringify(usrObject);

        //Inform user of result
        document.getElementById("Result").innerHTML = "<b>Registration successful.</b>";
    }
</script>
```



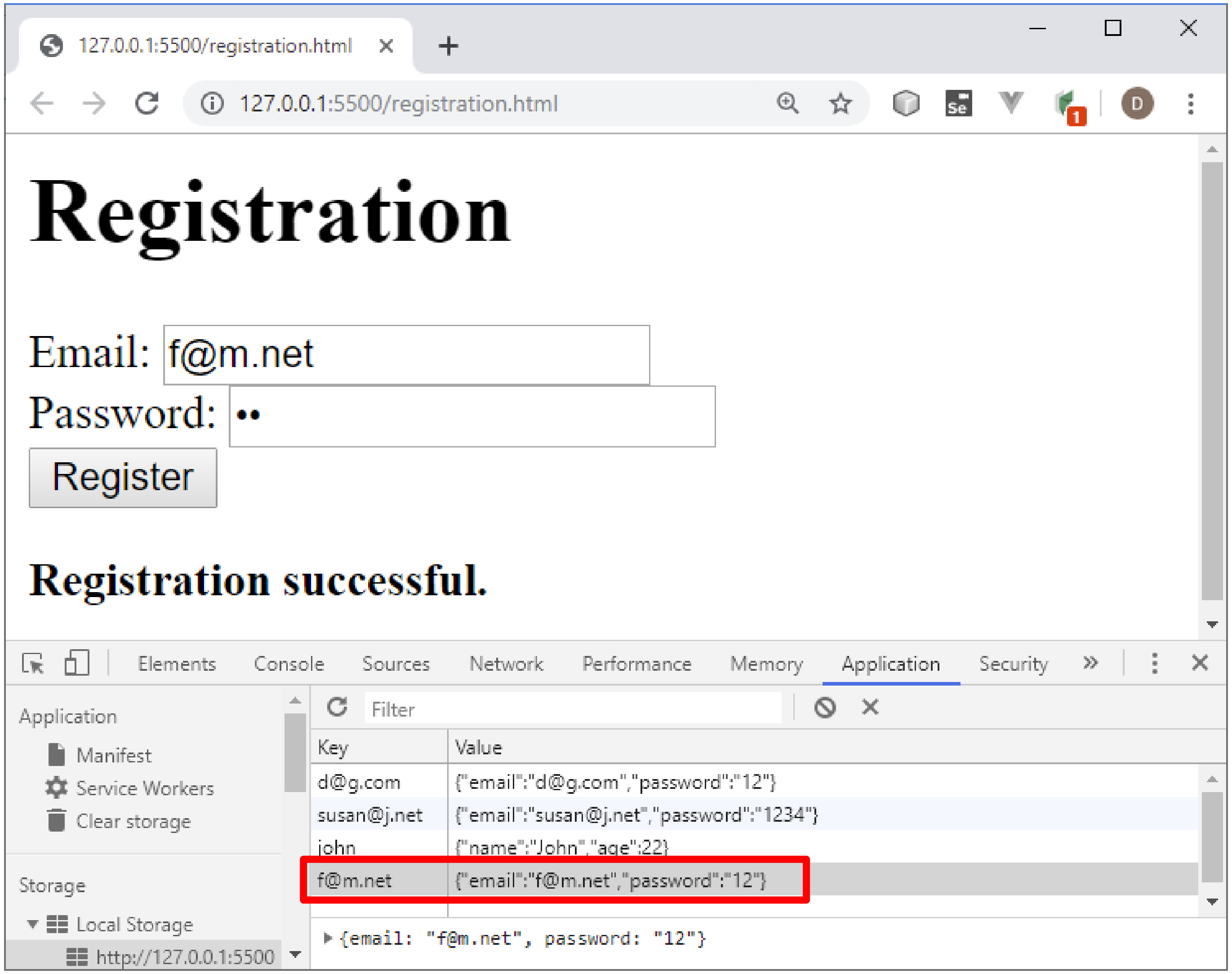
Registration

Email: f@m.net

Password: ••

Registration successful.

Registration



127.0.0.1:5500/registration.html

Registration

Email: f@m.net

Password: ••

Register

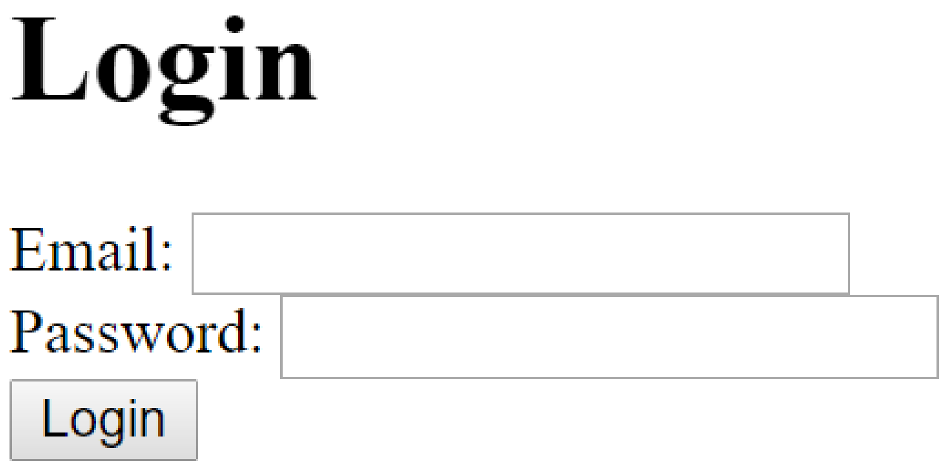
Registration successful.

Key	Value
d@g.com	{"email":"d@g.com","password":"12"}
susan@j.net	{"email":"susan@j.net","password":"1234"}
john	{"name":"John","age":22}
f@m.net	{"email":"f@m.net","password":"12"}

```
▶ {email: "f@m.net", password: "12"}
```

Login

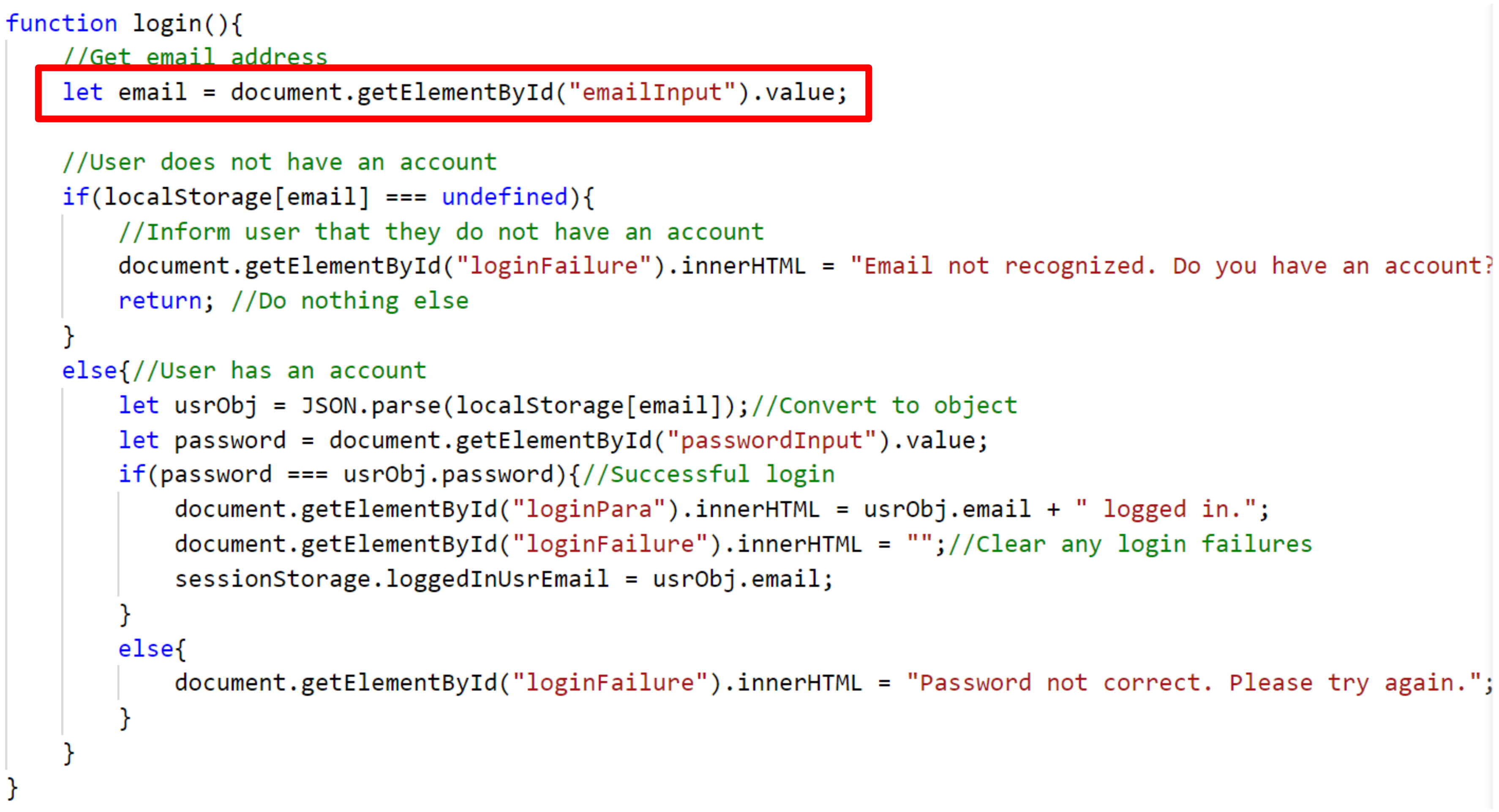
```
<h1>Login</h1>
<div id="loginPara">
        Email: <input type="email" id="emailInput"><br>
        Password: <input type="password" id="passwordInput"><br>
        <button onclick="login()">Login</button>
</div>
<p id="loginFailure" style="color: red;"></p>
```


Login

Email:

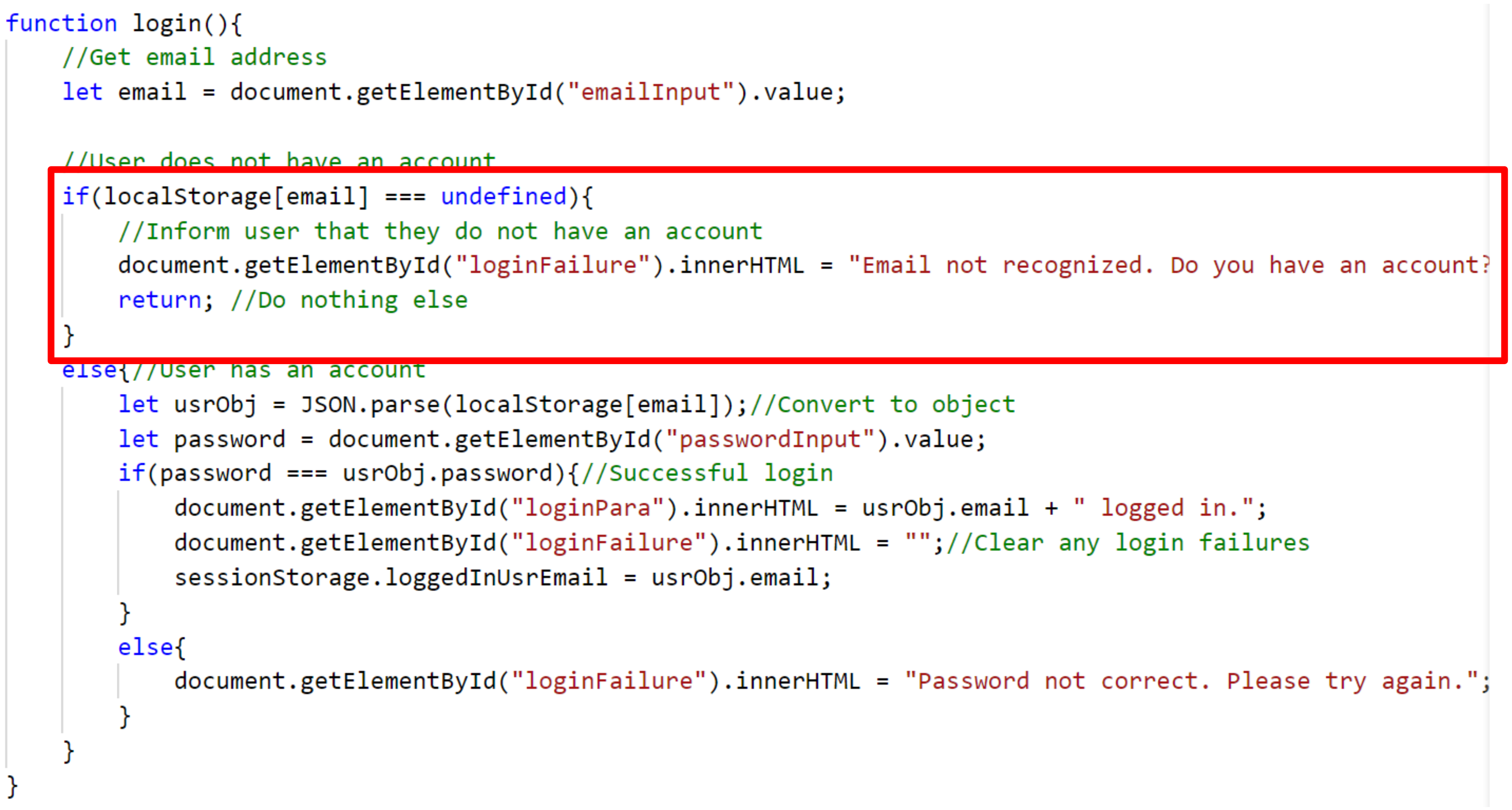
Password:

Login



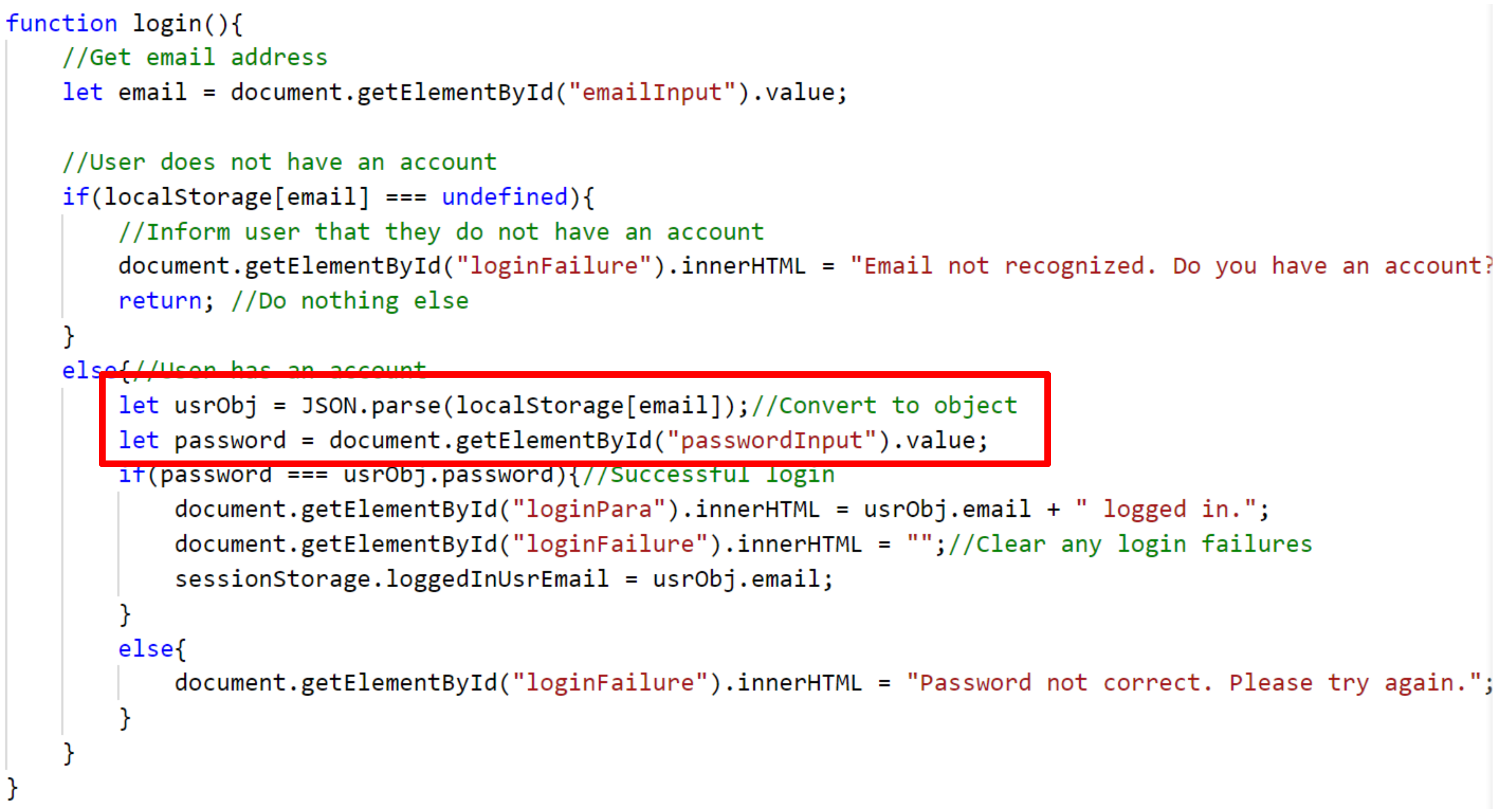
```
function login(){
    //Get email address
    let email = document.getElementById("emailInput").value;

    //User does not have an account
    if(localStorage[email] === undefined){
        //Inform user that they do not have an account
        document.getElementById("loginFailure").innerHTML = "Email not recognized. Do you have an account?
        return; //Do nothing else
    }
    else{//User has an account
        let usrObj = JSON.parse(localStorage[email]);//Convert to object
        let password = document.getElementById("passwordInput").value;
        if(password === usrObj.password){//Successful login
            document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
            document.getElementById("loginFailure").innerHTML = "";//Clear any login failures
            sessionStorage.loggedInUsrEmail = usrObj.email;
        }
        else{
            document.getElementById("loginFailure").innerHTML = "Password not correct. Please try again.";
        }
    }
}
```

Login



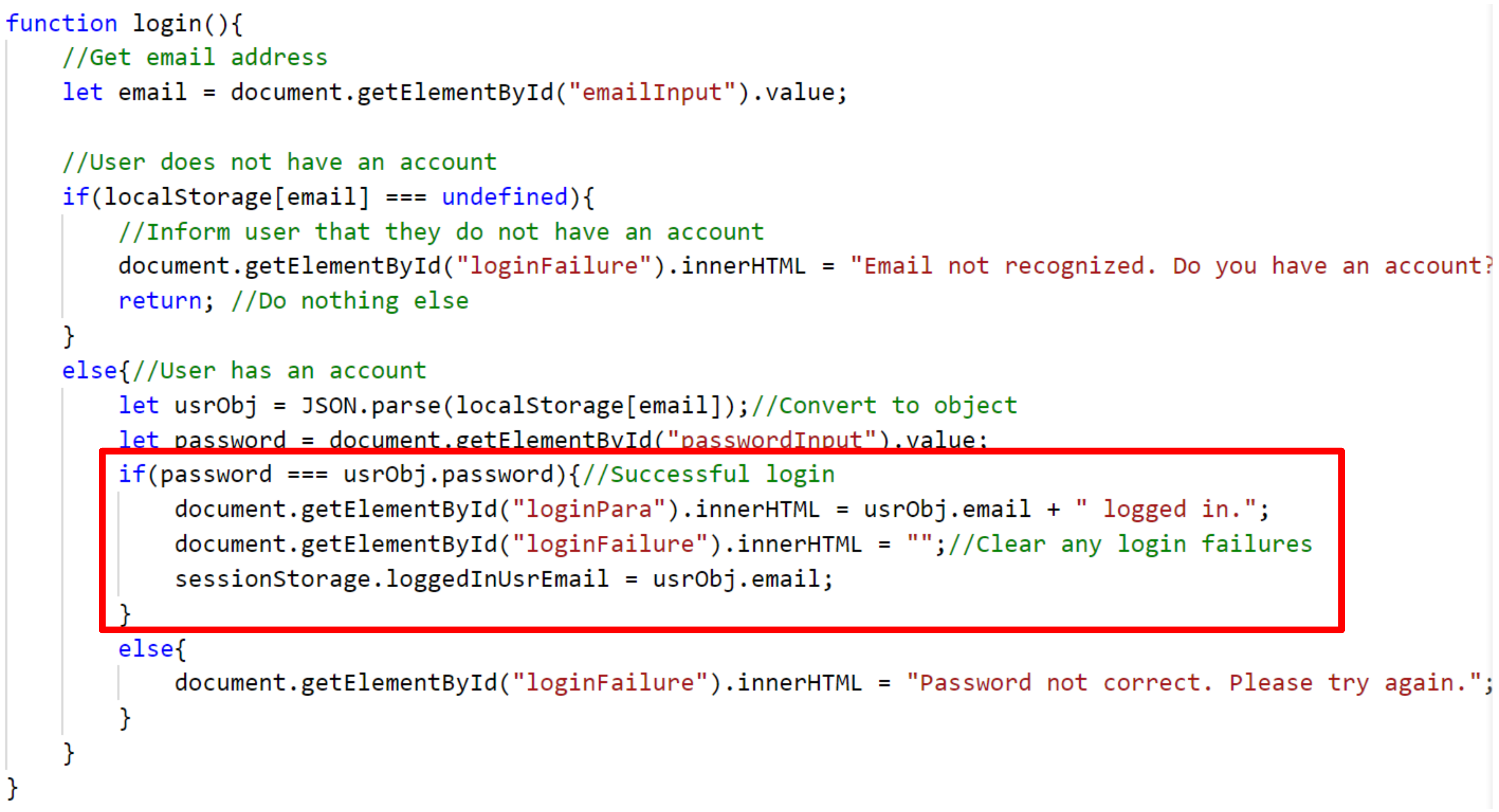
```
function login(){
    //Get email address
    let email = document.getElementById("emailInput").value;

    //User does not have an account
    if(localStorage[email] === undefined){
        //Inform user that they do not have an account
        document.getElementById("loginFailure").innerHTML = "Email not recognized. Do you have an account?
        return; //Do nothing else
    }
    else{//User has an account
        let usrObj = JSON.parse(localStorage[email]);//Convert to object
        let password = document.getElementById("passwordInput").value;
        if(password === usrObj.password){//Successful login
            document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
            document.getElementById("loginFailure").innerHTML = "";//Clear any login failures
            sessionStorage.loggedInUsrEmail = usrObj.email;
        }
        else{
            document.getElementById("loginFailure").innerHTML = "Password not correct. Please try again.";
        }
    }
}
```

Login



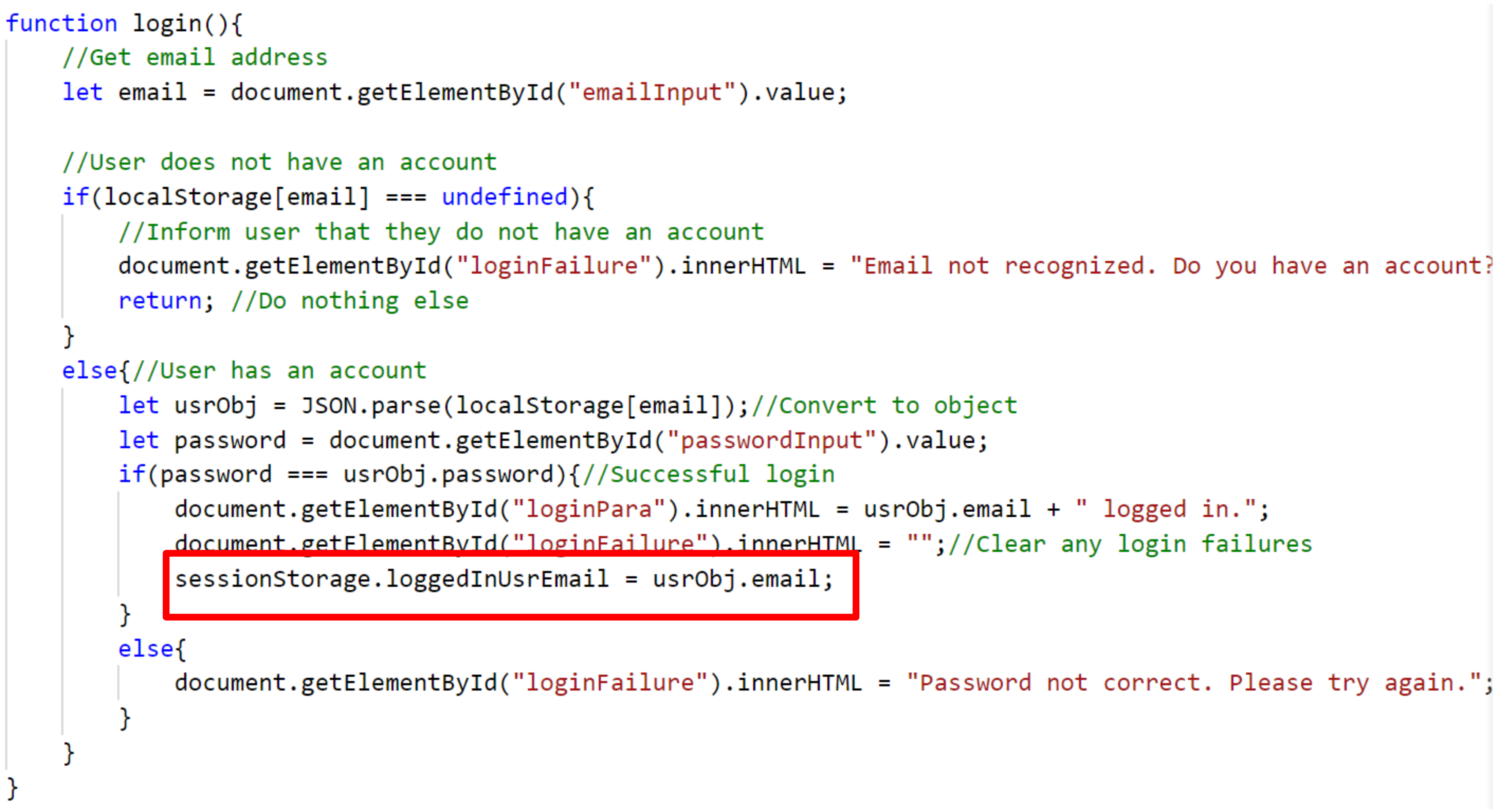
```
function login(){
    //Get email address
    let email = document.getElementById("emailInput").value;

    //User does not have an account
    if(localStorage[email] === undefined){
        //Inform user that they do not have an account
        document.getElementById("loginFailure").innerHTML = "Email not recognized. Do you have an account?
        return; //Do nothing else
    }
    else{//User has an account
        let usrObj = JSON.parse(localStorage[email]);//Convert to object
        let password = document.getElementById("passwordInput").value;
        if(password === usrObj.password){//Successful login
            document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
            document.getElementById("loginFailure").innerHTML = "";//Clear any login failures
            sessionStorage.loggedInUsrEmail = usrObj.email;
        }
        else{
            document.getElementById("loginFailure").innerHTML = "Password not correct. Please try again.";
        }
    }
}
```

Login



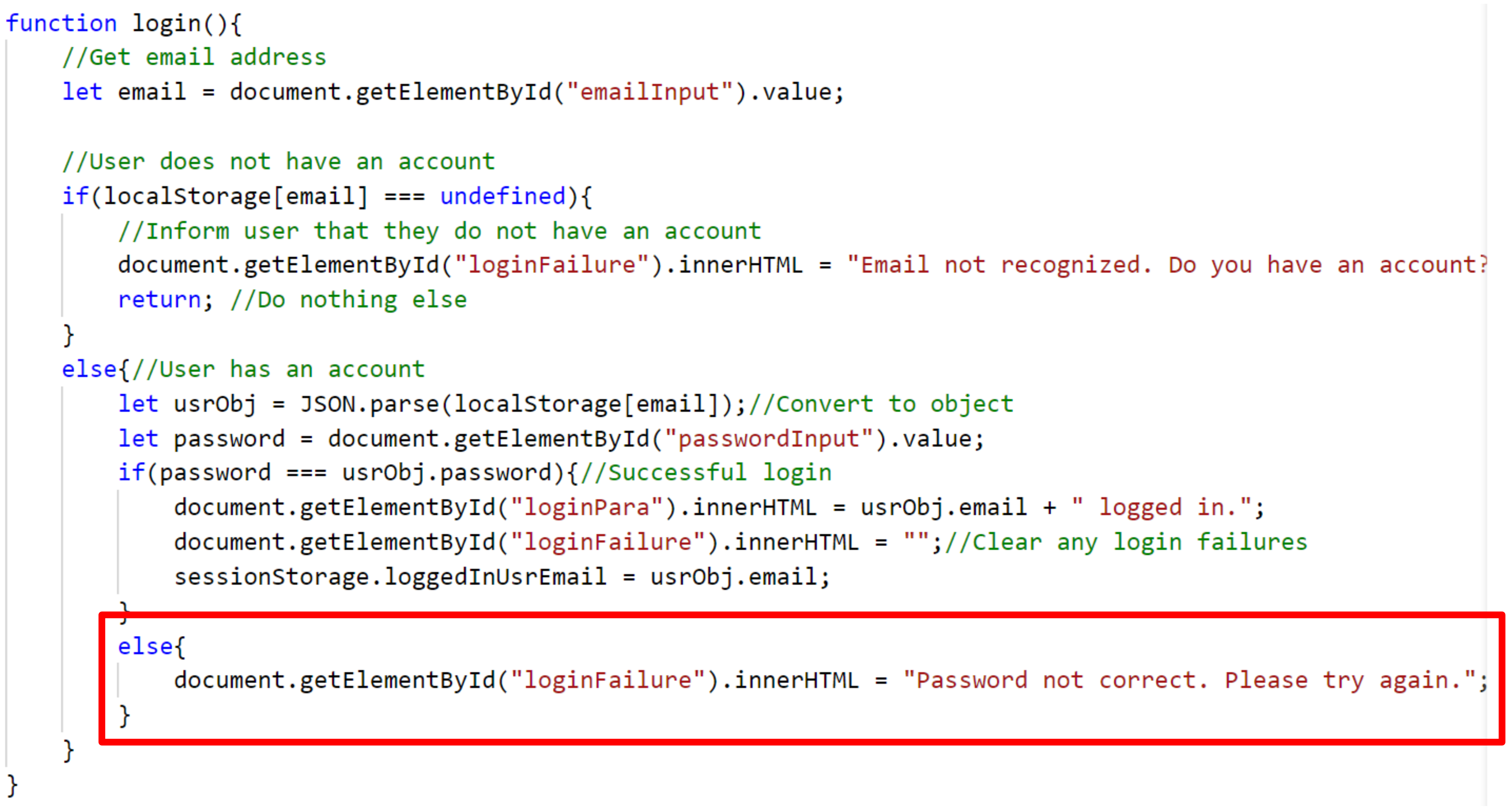
```
function login(){
    //Get email address
    let email = document.getElementById("emailInput").value;

    //User does not have an account
    if(localStorage[email] === undefined){
        //Inform user that they do not have an account
        document.getElementById("loginFailure").innerHTML = "Email not recognized. Do you have an account?
        return; //Do nothing else
    }
    else{//User has an account
        let usrObj = JSON.parse(localStorage[email]);//Convert to object
        let password = document.getElementById("passwordInput").value;
        if(password === usrObj.password){//Successful login
            document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
            document.getElementById("loginFailure").innerHTML = "";//Clear any login failures
            sessionStorage.loggedInUsrEmail = usrObj.email;
        }
        else{
            document.getElementById("loginFailure").innerHTML = "Password not correct. Please try again.";
        }
    }
}
```

Login



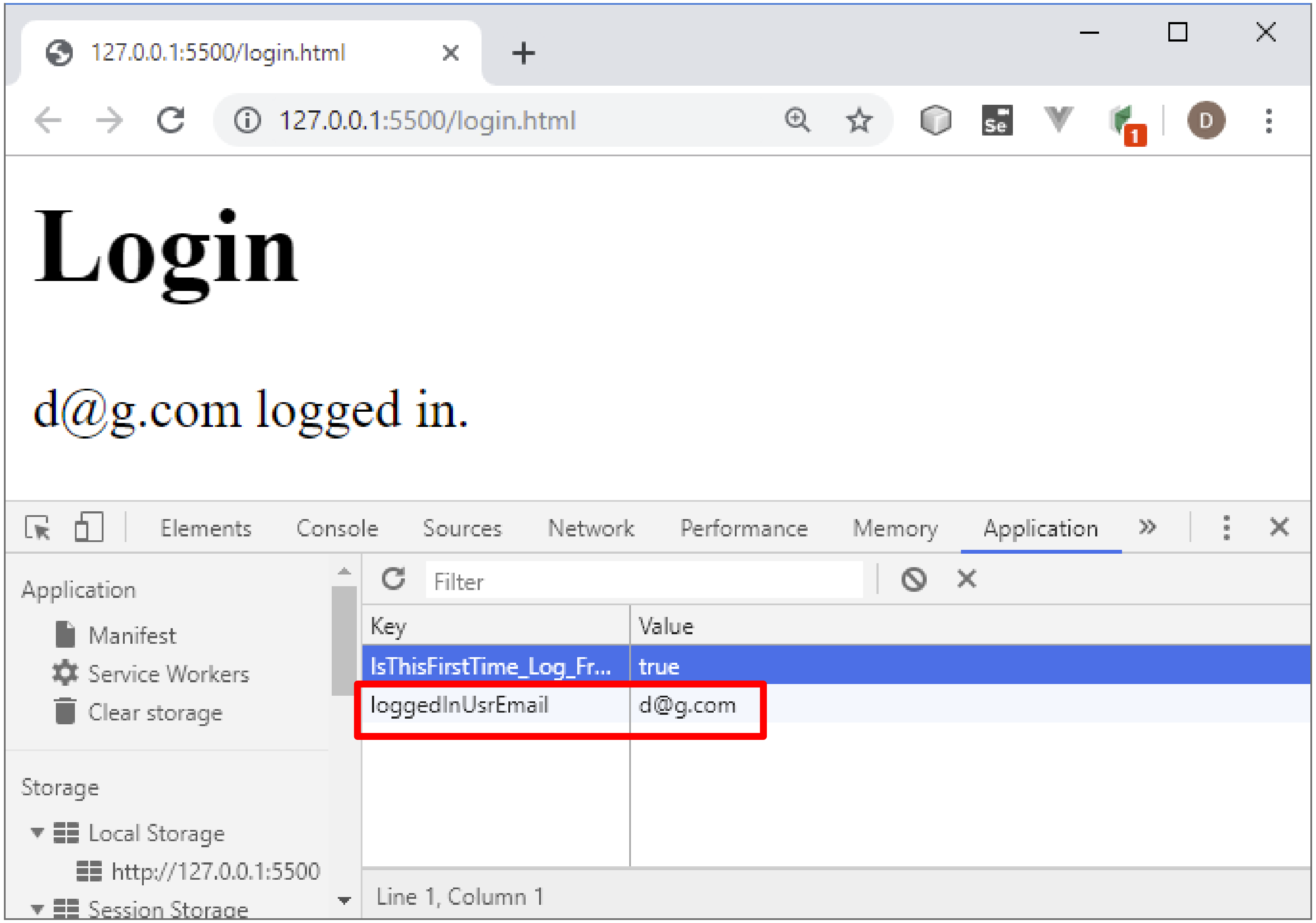
```
function login(){
    //Get email address
    let email = document.getElementById("emailInput").value;

    //User does not have an account
    if(localStorage[email] === undefined){
        //Inform user that they do not have an account
        document.getElementById("loginFailure").innerHTML = "Email not recognized. Do you have an account?
        return; //Do nothing else
    }
    else{//User has an account
        let usrObj = JSON.parse(localStorage[email]);//Convert to object
        let password = document.getElementById("passwordInput").value;
        if(password === usrObj.password){//Successful login
            document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
            document.getElementById("loginFailure").innerHTML = "";//Clear any login failures
            sessionStorage.loggedInUsrEmail = usrObj.email;
        }
        else{
            document.getElementById("loginFailure").innerHTML = "Password not correct. Please try again.";
        }
    }
}
```

Login



```
function login(){
    //Get email address
    let email = document.getElementById("emailInput").value;

    //User does not have an account
    if(localStorage[email] === undefined){
        //Inform user that they do not have an account
        document.getElementById("loginFailure").innerHTML = "Email not recognized. Do you have an account?
        return; //Do nothing else
    }
    else{//User has an account
        let usrObj = JSON.parse(localStorage[email]);//Convert to object
        let password = document.getElementById("passwordInput").value;
        if(password === usrObj.password){//Successful login
            document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
            document.getElementById("loginFailure").innerHTML = "";//Clear any login failures
            sessionStorage.loggedInUsrEmail = usrObj.email;
        }
        else{
            document.getElementById("loginFailure").innerHTML = "Password not correct. Please try again.";
        }
    }
}
```

Example: Login Page

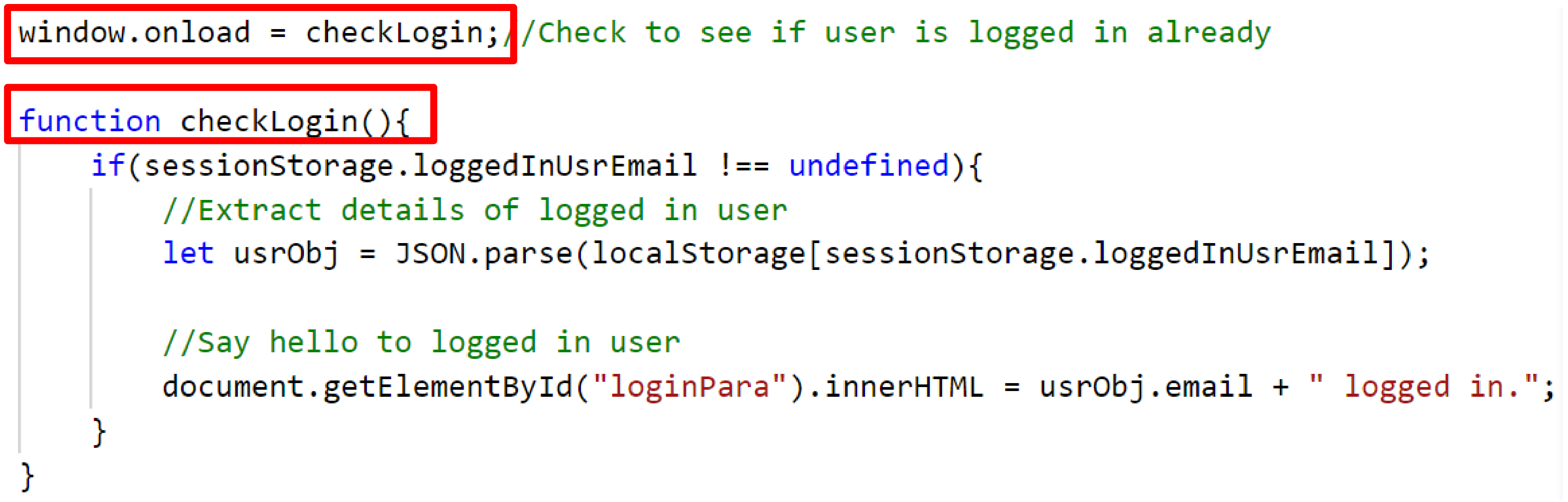


Login

d@g.com logged in.

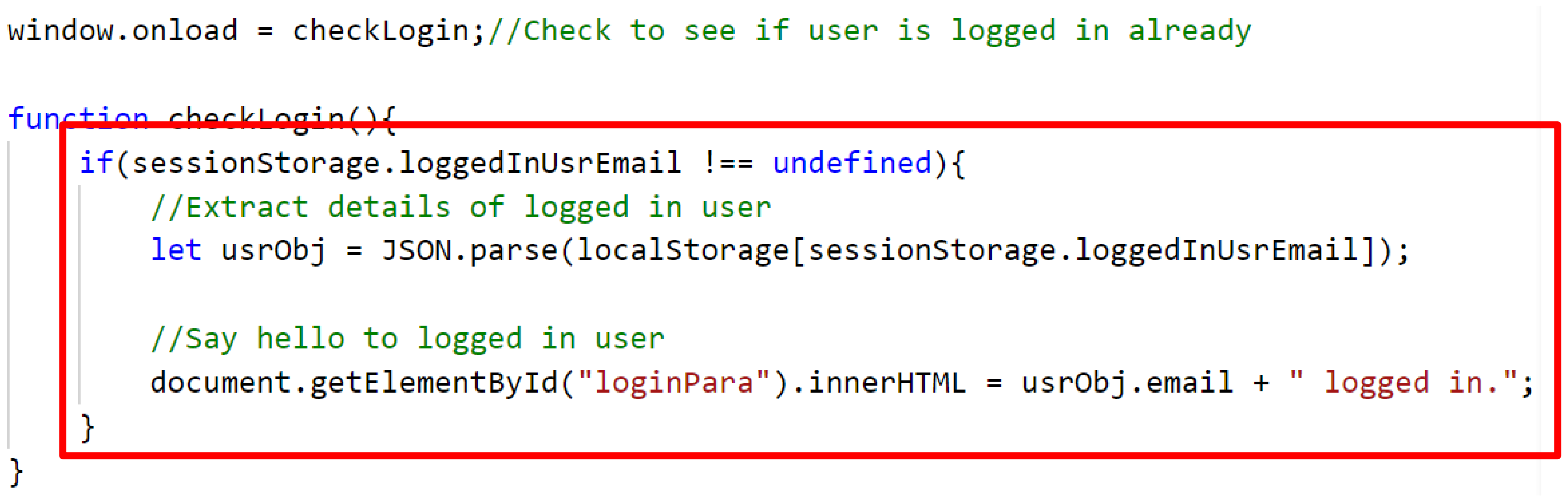
Key	Value
IsThisFirstTime_Log_Fr...	true
loggedInUsrEmail	d@g.com

Login



```
window.onload = checkLogin;//Check to see if user is logged in already

function checkLogin(){
    if(sessionStorage.loggedInUsrEmail !== undefined){
        //Extract details of logged in user
        let usrObj = JSON.parse(localStorage[sessionStorage.loggedInUsrEmail]);

        //Say hello to logged in user
        document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
    }
}
```

Login

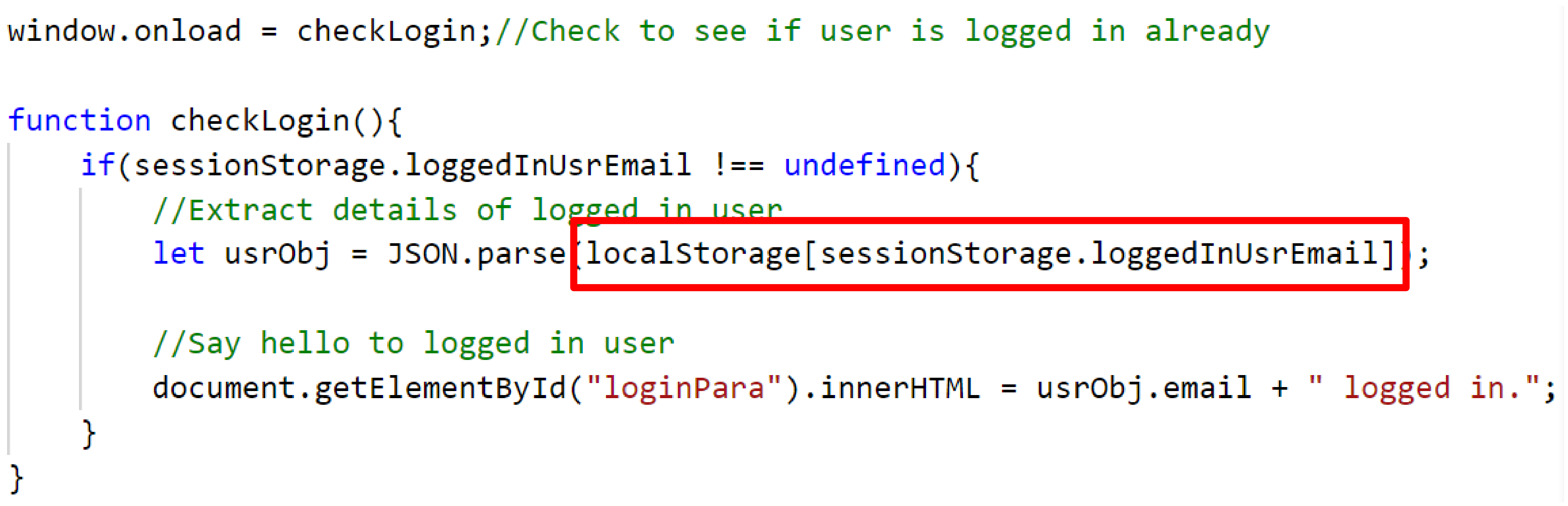


```javascript
window.onload = checkLogin;//Check to see if user is logged in already

function checkLogin(){
    if(sessionStorage.loggedInUsrEmail !== undefined){
        //Extract details of logged in user
        let usrObj = JSON.parse(localStorage[sessionStorage.loggedInUsrEmail]);

        //Say hello to logged in user
        document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
    }
}
```

Login

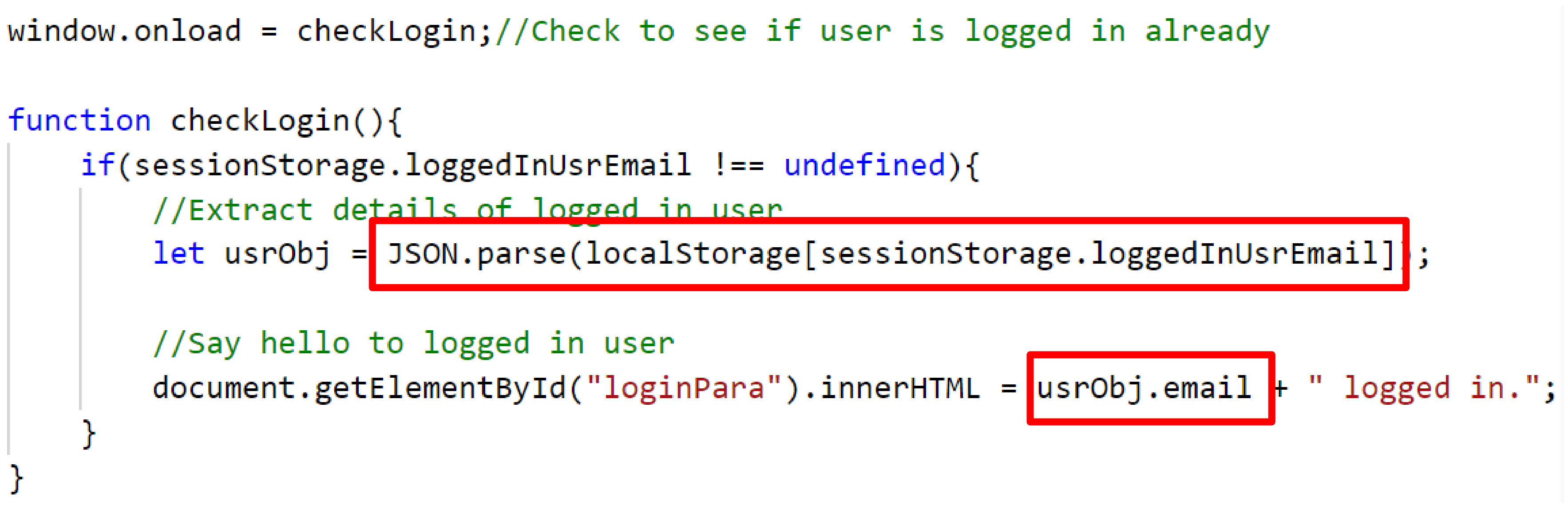
```
window.onload = checkLogin;//Check to see if user is logged in already

function checkLogin(){
    if(sessionStorage.loggedInUsrEmail !== undefined){
        //Extract details of logged in user
        let usrObj = JSON.parse(localStorage[sessionStorage.loggedInUsrEmail]);

        //Say hello to logged in user
        document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
    }
}
```

Login



```
window.onload = checkLogin;//Check to see if user is logged in already

function checkLogin(){
    if(sessionStorage.loggedInUsrEmail !== undefined){
        //Extract details of logged in user
        let usrObj = JSON.parse(localStorage[sessionStorage.loggedInUsrEmail]);

        //Say hello to logged in user
        document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
    }
}
```

Login



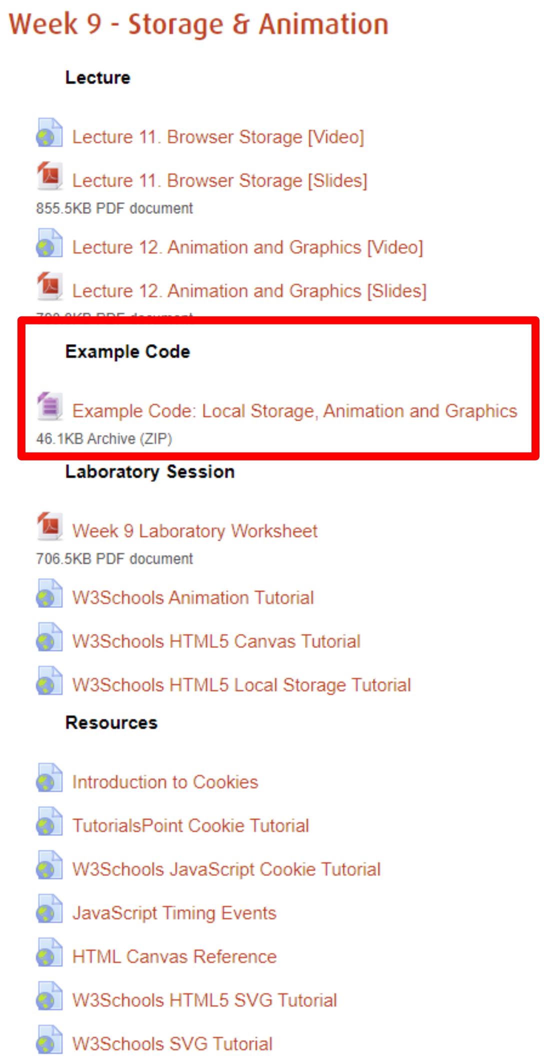
```
window.onload = checkLogin;//Check to see if user is logged in already

function checkLogin(){
    if(sessionStorage.loggedInUsrEmail !== undefined){
        //Extract details of logged in user
        let usrObj = JSON.parse(localStorage[sessionStorage.loggedInUsrEmail]);

        //Say hello to logged in user
        document.getElementById("loginPara").innerHTML = usrObj.email + " logged in.";
    }
}
```

Improvement to Registration and Login

- Store more attributes (name, address, etc.)
- Validate user input.
- Check to see if user exists already when registering.
- Add top score to user object.

Example Code

- The example code from this lecture is available on the course website.
- You are welcome to adapt it for your coursework



Week 9 - Storage & Animation

Lecture

Lecture 11. Browser Storage [Video]

Lecture 11. Browser Storage [Slides]
855.5KB PDF document

Lecture 12. Animation and Graphics [Video]

Lecture 12. Animation and Graphics [Slides]
[illegible] PDF document

Example Code

Example Code: Local Storage, Animation and Graphics
46.1KB Archive (ZIP)

Laboratory Session

Week 9 Laboratory Worksheet
706.5KB PDF document

W3Schools Animation Tutorial

W3Schools HTML5 Canvas Tutorial

W3Schools HTML5 Local Storage Tutorial

Resources

Introduction to Cookies

TutorialsPoint Cookie Tutorial

W3Schools JavaScript Cookie Tutorial

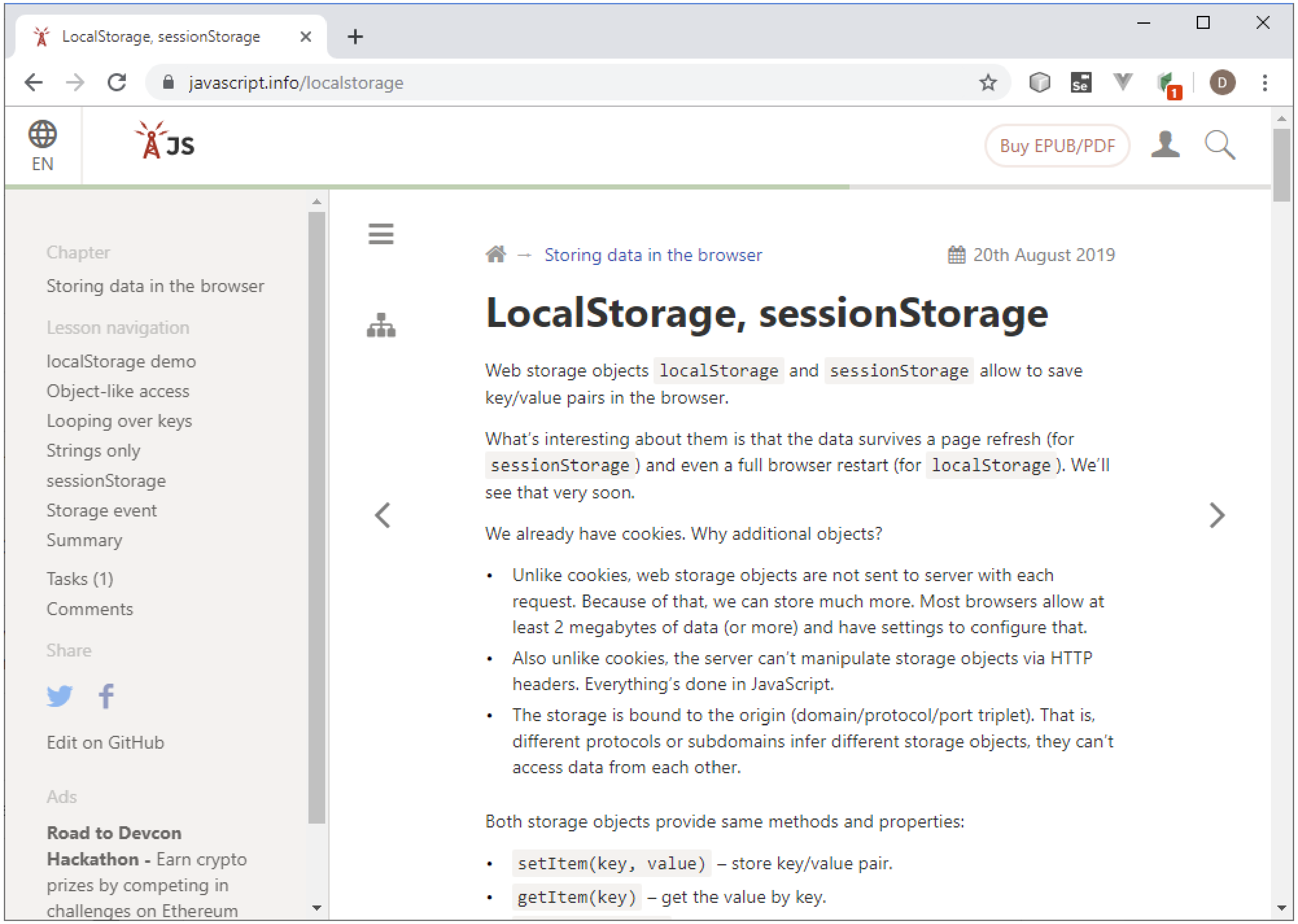
JavaScript Timing Events

HTML Canvas Reference

W3Schools HTML5 SVG Tutorial

W3Schools SVG Tutorial

Modern JavaScript Tutorial



LocalStorage, sessionStorage

javascript.info/localstorage

EN

Buy EPUB/PDF

Chapter

Storing data in the browser

Lesson navigation

localStorage demo
Object-like access
Looping over keys
Strings only
sessionStorage
Storage event
Summary

Tasks (1)
Comments

Share

Edit on GitHub

Ads

Road to Devcon Hackathon - Earn crypto prizes by competing in challenges on Ethereum

→ Storing data in the browser   20th August 2019

LocalStorage, sessionStorage

Web storage objects `localStorage` and `sessionStorage` allow to save key/value pairs in the browser.

What's interesting about them is that the data survives a page refresh (for `sessionStorage`) and even a full browser restart (for `localStorage`). We'll see that very soon.

We already have cookies. Why additional objects?

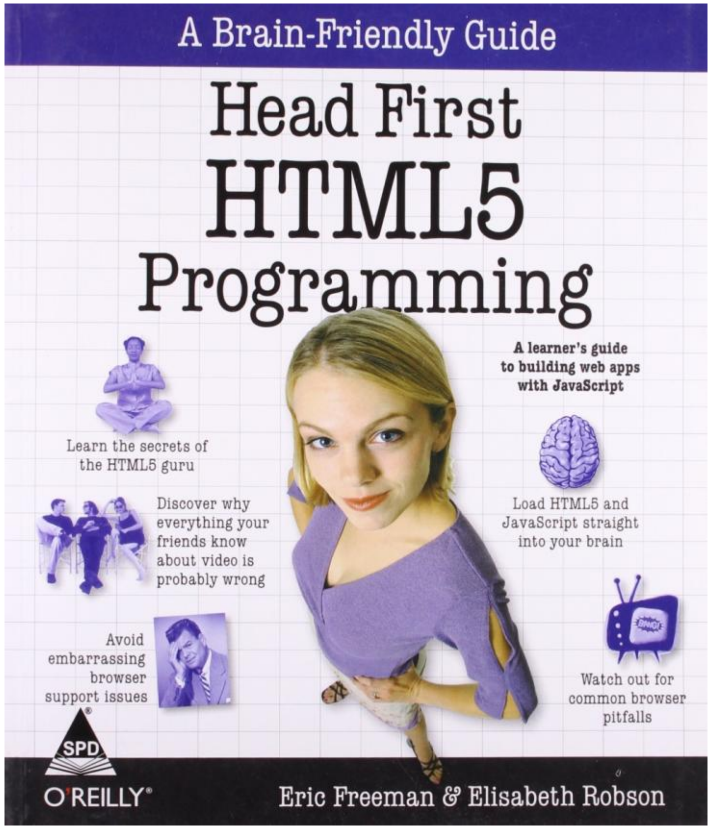
- Unlike cookies, web storage objects are not sent to server with each request. Because of that, we can store much more. Most browsers allow at least 2 megabytes of data (or more) and have settings to configure that.
- Also unlike cookies, the server can't manipulate storage objects via HTTP headers. Everything's done in JavaScript.
- The storage is bound to the origin (domain/protocol/port triplet). That is, different protocols or subdomains infer different storage objects, they can't access data from each other.

Both storage objects provide same methods and properties:

- `setItem(key, value)` – store key/value pair.
- `getItem(key)` – get the value by key.

Further Reading

- Eric Freeman and Elisabeth Robson (2011). *Head First HTML5 Programming*. Sebastopol, CA: O'Reilly.
- **Chapter 9**


A Brain-Friendly Guide
Head First HTML5 Programming
A learner's guide to building web apps with JavaScript
Learn the secrets of the HTML5 guru
Discover why everything your friends know about video is probably wrong
Load HTML5 and JavaScript straight into your brain
Avoid embarrassing browser support issues
Watch out for common browser pitfalls
SPD
O'REILLY
Eric Freeman & Elisabeth Robson

SUMMARY

Summary

- This lecture has introduced you to cookies and HTML local storage.
- You will use local storage and session storage for registration and login on your game website.