

Introduction to Web Technology

HTML5: Client-Side Storage

Joseph Tonien
School of Computing and Information Technology
University of Wollongong

Client-Side Web Storage

- Store data on the client side, instead of the server
- Make the web application available offline
- The storage is per origin (protocol + domain + port)
- Simple storage: data is stored in name/value pair

2 types of storage:

- **localStorage**: a single persistent object which stores data with no expiration date;
- **sessionStorage**: stores data for one session only, data is cleared when the browser tab is closed.

Client-Side Web Storage

Checking if the browser supports web storage or not:

```
if (typeof(Storage) !== "undefined") {  
    // code for web storage ...  
} else {  
    console.log("Web Storage not supported");  
}
```

Client-Side Web Storage

Storing and retrieving data from Web Storage:

```
// storing data to the localStorage  
localStorage.setItem("the-key", "the-value");  
  
// get data from localStorage  
var the-value = localStorage.getItem("the-key");
```

Removing data from Web Storage:

```
// removing data to the localStorage  
localStorage.removeItem("the-key");
```

Example: To-Do-List (1)

Task: Urgency:

We want to create a web application where the user can create a to-do-list and save it to the local storage.

We will store the JSON of the task list into the local storage:

Key	Value
toDoListJSON	[{"task":"Grocery shopping","urgency":"Low"},...]

```
[
  {
    "task":"Grocery shopping",
    "urgency":"Low"
  },
  {
    "task":"Wash the dishes",
    "urgency":"Medium"
  },
  {
    "task":"Do web tech assignment 2",
    "urgency":"High"
  },
  {
    "task":"Do laundry",
    "urgency":"Low"
  }
]
```

Example: To-Do-List (1)

When the page load, we need to do the database initialization:

- (i) get the to-do list JSON from the local storage,
- (ii) parse the JSON.

```
<body onLoad="initApp() ; ">
```

```
...
```

```
// initialize the application
```

```
function initApp() {
```

```
    if (typeof(Storage) !== "undefined") {
```

```
        // get the to-do list JSON from local storage
```

```
        // parse the JSON to the toDoList
```

```
    } else {
```

```
        console.log("Web Storage not supported");
```

```
    }
```

```
}
```

Example: To-Do-List (1)

```
// to-do list which is saved to web storage
var toDoList = [];

function initApp() {
    if (typeof(Storage) !== "undefined") {

        // get the to-do list JSON from local storage
        var toDoListJSON = localStorage.getItem("toDoListJSON");

        // parse the JSON to the toDoList
        if(toDoListJSON !== null){
            toDoList = JSON.parse(toDoListJSON);
        }

    } else {
        console.log("Web Storage not supported");
    }
}
```

Example: To-Do-List (1)

Adding a task to the to-do-list:

Task: Urgency: Low ▼ Add

Show Task

```
<body onLoad="initApp();">
```

Task:

```
<input id="task" type="text" />
```

Urgency:

```
<select id="urgency">
  <option value="High">High</option>
  <option value="Medium">Medium</option>
  <option value="Low" selected="selected">Low</option>
</select>
```

```
<button onClick="addTask()">
```

Add

```
</button>
```


Example: To-Do-List (1)

Adding a task to the to-do-list:

Task: Urgency:

```
// add a task
function addTask() {
    // get task description from user input
    var toDoObj = {};
    toDoObj.task = document.getElementById("task").value;
    toDoObj.urgency = document.getElementById("urgency").value;

    // add the task to toDoList
    toDoList.push(toDoObj);

    // if Web Storage supported then update the JSON
    if (typeof(Storage) !== "undefined") {
        localStorage.setItem("toDoListJSON", JSON.stringify(toDoList) );
    }
}
```

Example: To-Do-List (1)

Showing all the tasks:

```
<button onClick="showTask()">  
Show Task  
</button>
```

```
<br /><br />
```

```
<div id="taskDisplay">  
</div>
```

Task: Urgency: Low ▾ Add

Show Task

Task: Grocery shopping, Urgency: Low

Task: Wash the dishes, Urgency: Medium

Task: Do web tech assignment 2, Urgency: High

Task: Do laundry, Urgency: Low

Example: To-Do-List (1)

Showing all the tasks:

```
<div id="taskDisplay">  
</div>
```

```
// show all the tasks  
function showTask() {  
    var html = "";  
  
    for (var i=0; i<todoList.length; i++) {  
        var todo = todoList[i];  
  
        html += "Task: " + todo.task + ", Urgency: " + todo.urgency + "<br /><br />";  
    }  
  
    document.getElementById("taskDisplay").innerHTML = html;  
}
```

Task: Urgency:

Task: Grocery shopping, Urgency: Low

Task: Wash the dishes, Urgency: Medium

Task: Do web tech assignment 2, Urgency: High

Task: Do laundry, Urgency: Low

Example: To-Do-List (2)

To-Do-List(2) example is the same as the previous To-Do-List(1) example, except that each task is displayed with a color corresponding to its urgency level.

Task: Urgency:

Grocery shopping

Wash the dishes

Do web tech assignment 2

Do laundry

Example: To-Do-List (2)

Task: Urgency: Low

- Grocery shopping
- Wash the dishes
- Do web tech assignment 2
- Do laundry

```
// show all the tasks
function showTask() {
    var html = "";

    for (var i=0; i<todoList.length; i++) {
        var todo = todoList[i];

        if(todo.urgency == "Low") {
            html += "<span style='color:green'>" + todo.task + "</span>";
        } else if(todo.urgency == "Medium") {
            html += "<span style='color:orange'>" + todo.task + "</span>";
        } else if(todo.urgency == "High") {
            html += "<span style='color:red'>" + todo.task + "</span>";
        }

        html += "<br /><br />";
    }

    document.getElementById("taskDisplay").innerHTML = html;
}
```

Example: To-Do-List (3)

To-Do-List(3) example is the same as the previous To-Do-List(2) example, except that each task is displayed with a delete symbol, and when the user clicks on the delete symbol the task will be deleted.

We will use an image `delete.png` for the delete symbol.

Task: Urgency:

Grocery shopping

Wash the dishes

Do web tech assignment 2


Do laundry


Example: To-Do-List (3)

We will use an image `delete.png` for the delete symbol.

Task: Urgency:

Grocery shopping 

Wash the dishes 

Do web tech assignment 2 

Do laundry 

``

Example: To-Do-List (3)

Task: Urgency: Low ▼ Add

Show Task

Grocery shopping X

Wash the dishes X

Do web tech assignment 2 X

Do laundry X

```
// show all the tasks
function showTask(){
    var html = "";

    for (var i=0; i<todoList.length; i++) {
        var todo = todoList[i];

        if(todo.urgency == "Low"){
            html += "<span style='color:green'>" + todo.task + "</span>";
        }else if(todo.urgency == "Medium"){
            html += "<span style='color:orange'>" + todo.task + "</span>";
        }else if(todo.urgency == "High"){
            html += "<span style='color:red'>" + todo.task + "</span>";
        }

        html += "<img src='delete.png' onClick=' deleteTask(\"" + todo.task + "\")' />";
        html += "<br /><br />";
    }

    document.getElementById("taskDisplay").innerHTML = html;
}
```


Example: To-Do-List (3)

Task: Urgency:

Grocery shopping

Wash the dishes

Do math assignment 2

laundry

```
// delete a task
function deleteTask(task) {

    // search for the deleted task through the list
    for (var i=0; i<todoList.length; i++) {
        var todo = todoList[i];

        if(todo.task == task) {
            todoList.splice(i, 1);
            break;
        }
    }

    // if Web Storage supported then update the JSON
    if (typeof(Storage) !== "undefined"){
        localStorage.setItem("todoListJSON", JSON.stringify(todoList) );
    }

    // show all the tasks
    showTask();
}
```

Example: To-Do-List (4)

To-Do-List(4) example is the same as the previous To-Do-List(3) example, except that the button “Show Task” is removed. Initially, all tasks will be displayed and after adding a new task, the list of updated tasks will be displayed.

Task: Urgency:

Grocery shopping

Wash the dishes

Do web tech assignment 2

Do laundry

Example: To-Do-List (4)

Task: Urgency:

Grocery shopping

Wash the dishes

Do web tech assignment 2

Do laundry

```
<body onLoad="initApp() ; ">
```

```
...
```

```
// initialize the application
```

```
function initApp() {
```

```
    if (typeof(Storage) !== "undefined") {
```

```
        // get the to-do list JSON from local storage
```

```
        // parse the JSON to the toDoList
```

```
        // show all the tasks
```

```
    } else {
```

```
        console.log("Web Storage not supported");
```

```
    }
```

```
}
```

Example: To-Do-List (4)

```
// to-do list which is saved to web storage
var toDoList = [];

function initApp() {
    if (typeof(Storage) !== "undefined") {

        // get the to-do list JSON from local storage
        var toDoListJSON = localStorage.getItem("toDoListJSON");

        // parse the JSON to the toDoList
        if(toDoListJSON !== null){
            toDoList = JSON.parse(toDoListJSON);
        }

        // show all the tasks
        showTask();

    } else {
        console.log("Web Storage not supported");
    }
}
```

Example: To-Do-List (4)

Adding a task to the to-do-list:

Task: Urgency:

```
// add a task
function addTask() {
    // get task description from user input
    var toDoObj = {};
    toDoObj.task = document.getElementById("task").value;
    toDoObj.urgency = document.getElementById("urgency").value;

    // add the task to toDoList
    toDoList.push(toDoObj);

    // if Web Storage supported then update the JSON
    if (typeof(Storage) !== "undefined"){
        localStorage.setItem("toDoListJSON", JSON.stringify(toDoList) );
    }

    // show all the tasks
    showTask();
}
```

Example: To-Do-List (4)

```
// delete a task
function deleteTask(task) {

    // search for the deleted task through the list
    for (var i=0; i<todoList.length; i++) {
        var todo = todoList[i];

        if(todo.task == task) {
            todoList.splice(i, 1);
            break;
        }
    }

    // if Web Storage supported then update the JSON
    if (typeof(Storage) !== "undefined"){
        localStorage.setItem("todoListJSON", JSON.stringify(todoList) );
    }

    // show all the tasks
    showTask();
}
```

Example: Math Practice

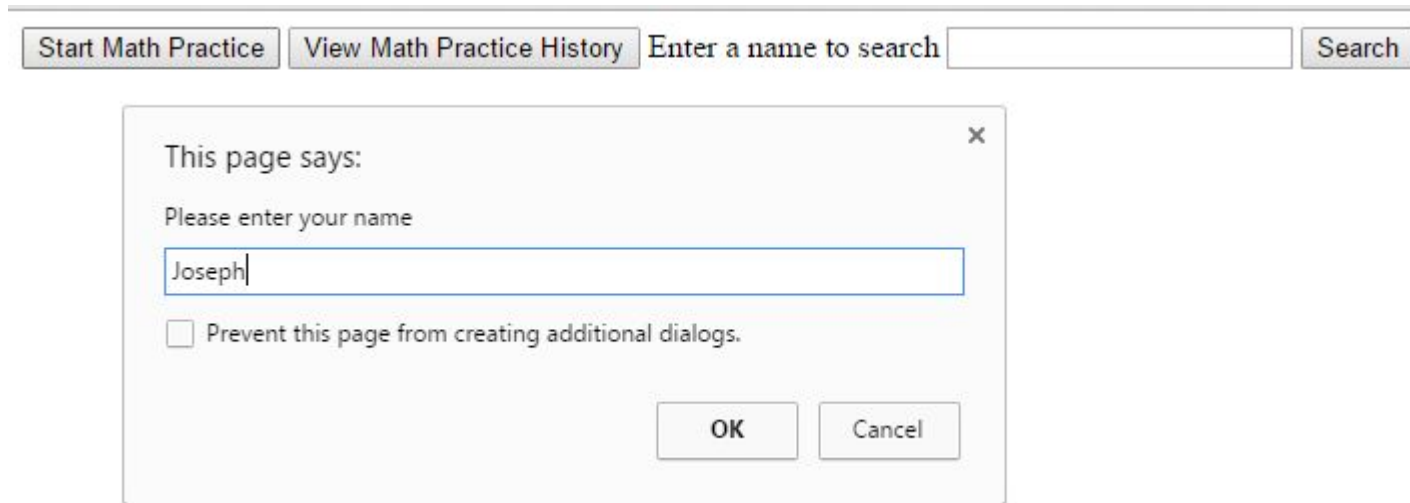
We want to create a web application where children can practice mathematics and parents can view the result of their kids practice.

Enter a name to search

Example: Math Practice

This is how the application should work.

When the user clicks on the button “Start Math Practice”, the user will be asked to enter his/her name.



The image shows a web application interface with a header bar containing three buttons: "Start Math Practice", "View Math Practice History", and "Enter a name to search" followed by a text input field and a "Search" button. Below the header, a modal dialog box is displayed. The dialog has a title bar with a close button (X). The main content of the dialog says "This page says:" followed by "Please enter your name". Below this is a text input field containing the name "Joseph". At the bottom of the dialog, there is a checkbox labeled "Prevent this page from creating additional dialogs." and two buttons: "OK" and "Cancel".

Start Math Practice View Math Practice History Enter a name to search Search

This page says: ×

Please enter your name

Joseph

☐ Prevent this page from creating additional dialogs.

OK Cancel

Example: Math Practice

Then the application prints a greetings and generate a math question.

[Start Math Practice](#) [View Math Practice History](#) Enter a name to search [Search](#)

Hi Joseph!

$1 + 1 =$ [Check Answer](#)

Example: Math Practice

User can enter an answer to the math problem and can check if it is correct.

New question will be generated.

[Start Math Practice](#)[View Math Practice History](#)

Hi Joseph!

$$1 + 1 = 2 \checkmark$$

$$1 \times 2 = 2 \checkmark$$

$$7 + 5 = 11 \times$$

You have answered 2 out of 3 questions correctly

$$7 + 5 =$$

Example: Math Practice

Parents can click on the button “View Math Practice History” to see the result of their kids practice.

[Start Math Practice](#) [View Math Practice History](#) Enter a name to search [Search](#)

Joseph

2018-10-16T03:03:08.553Z

$$9 \times 1 = 9 \checkmark$$

$$8 \times 2 = 16 \checkmark$$

John

2018-10-16T03:04:45.063Z

$$3 + 10 = 23 \times$$

$$0 \times 8 = 8 \times$$

$$6 \times 10 = 60 \checkmark$$

Example: Math Practice

Parents can enter a name to search

[Start Math Practice](#) [View Math Practice History](#) Enter a name to search

John

2018-10-16T03:04:45.063Z

$$3 + 10 = 23 \times$$

$$0 \times 8 = 8 \times$$

$$6 \times 10 = 60 \checkmark$$

John

10/16/2018, 4:07:05 PM

$$13 - 3 = 10 \checkmark$$

$$5 \times 10 = 2 \times$$

References

- <https://www.w3.org/TR/webstorage/>
- https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API