

# Mise en œuvre du Web Storage au travers d'exemples

Dans la mesure où les méthodes associées à `sessionStorage` et à `localStorage` sont identiques, les deux exemples utilisent tous les deux `localStorage`.

## 1. Exemple 1 : Stockage par `localStorage` de chaînes de caractères

Dans ce premier exemple, l'objectif va être d'étudier les possibilités offertes par `localStorage` dans la gestion (création, accès, suppression) d'informations de type chaînes de caractères.

### Script complet

Le code source du script est reproduit intégralement ci-après :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<!--
NOM DU SCRIPT : LOCAL_STORAGE_01.htm
REALISATION INFORMATIQUE : Christian VIGOUROUX
DATE DE CREATION : 01/01/2014
DATE DE DERNIERE MODIFICATION : 01/01/2014
OBJET : Stockage avec utilisation de localStorage
-->

<!-- Début script HTML -->
<html>

    <!-- Début en-tête script HTML -->
    <head>

        <!-- Balise meta -->
        <meta http-equiv="Content-Type"
        content="text/html; charset=utf-8" />

        <!-- Titre du script HTML -->
        <title>LOCAL_STORAGE_01</title>

        <!-- Début script JavaScript -->
        <script type='text/javascript'>

            /* Déclarations de variables communes
            à toutes les fonctions */
            var intitule;
            var nomPersonne;
            var prenomPersonne;
            var mailPersonne;

            /* Fonction ecrireLocalStorage */
            function ecrireLocalStorage()
            {
```

```

/* Test possibilité d'utiliser localStorage */
if (typeof localStorage != "undefined")
{
    /* Stockage du nom (nomPersonne)
    dans le stockage localStorage */
    nomPersonne = "VIGOUROUX";
    localStorage.setItem("nomPersonne",
    nomPersonne);
    /* Stockage du prénom (prenomPersonne)
    dans le stockage localStorage */
    prenomPersonne = "Christian";
    localStorage.setItem("prenomPersonne",
    prenomPersonne);
    /* Stockage du mail (mailPersonne)
    dans le stockage localStorage */
    mailPersonne = "christian.vigouroux@online.fr";
    localStorage.setItem("mailPersonne",
    mailPersonne);
    /* Affichage de contrôle */
    intitule
    = "Contenu du localStorage (après écriture)";
    document.getElementById("intitule").innerHTML
    = intitule;
    document.getElementById("nomPersonne").innerHTML
    = nomPersonne;
    document.getElementById("prenomPersonne")
    .innerHTML = prenomPersonne;
    document.getElementById("mailPersonne").innerHTML
    = mailPersonne;
}
else
{
    /* Message d'erreur (pas de possibilité
    de stockage localStorage) */
    alert("localStorage n'est pas supporté");
}
};

/* Fonction lireLocalStorage */
function lireLocalStorage()
{
    /* Test possibilité d'utiliser localStorage */
    if (typeof localStorage != "undefined")
    {
        /* Récupération du nom (nomPersonne)
        depuis le stockage localStorage */
        nomPersonne = localStorage.getItem("nomPersonne");
        /* Récupération du prénom (prenomPersonne)
        depuis le stockage localStorage */
        prenomPersonne
        = localStorage.getItem("prenomPersonne");
        /* Récupération du mail (mailPersonne)
        depuis le stockage localStorage */
        mailPersonne

```

```

        = localStorage.getItem("mailPersonne");
        /* Affichage de contrôle */
        intitule
        = "Contenu du localStorage (après relecture)";
        document.getElementById("intitule").innerHTML
        = intitule;
        document.getElementById("nomPersonne").innerHTML
        = nomPersonne;
        document.getElementById("prenomPersonne")
        .innerHTML = prenomPersonne;
        document.getElementById("mailPersonne").innerHTML
        = mailPersonne;
    }
    else
    {
        /* Message d'erreur (pas de possibilité
        de stockage localStorage) */
        alert("localStorage n'est pas supporté");
    }
};

/* Fonction supprimerLocalStorage */
function supprimerLocalStorage()
{
    /* Test possibilité d'utiliser localStorage */
    if (typeof localStorage != "undefined")
    {
        /* Suppression des entrées nomPersonne,
        prenomPersonne et mailPersonne
        dans le localStorage */
        /* NB : sessionStorage.clear(); supprimerait
        également l'ensemble des entrées
        du localStorage */
        sessionStorage.removeItem("nomPersonne");
        sessionStorage.removeItem("prenomPersonne");
        sessionStorage.removeItem("mailPersonne");
        /* Affichage de contrôle */
        intitule
        = "Contenu du localStorage (après suppression)";
        document.getElementById("intitule")
        .innerHTML = intitule;
        document.getElementById("nomPersonne")
        .innerHTML = "";
        document.getElementById("prenomPersonne")
        .innerHTML = "";
        document.getElementById("mailPersonne")
        .innerHTML = "";
    }
    else
    {
        /* Message d'erreur (pas de possibilité
        de stockage localStorage) */
        alert("localStorage n'est pas supporté");
    }
}

```

```

        }

    };

</script>

</head>

<!-- Début section body du script HTML -->
<body>

    <!-- Titre du traitement -->
    <h1>Editions ENI - JavaScript - LOCAL_STORAGE_01</h1>

    <!-- Début script JavaScript -->
    <script type='text/javascript'>

        /* Affichage du nom du script */
        alert("LOCAL_STORAGE_01");

    </script>

    <!-- Formulaire avec un bouton d'écriture,
    un bouton de lecture et un bouton de suppression -->
    <form>
        <input
            type="button"
            id="boutonEcriture"
            value="Ecriture dans localStorage"
            onclick="ecrireLocalStorage();"
        />
        <br />
        <input
            type="button"
            id="boutonLecture"
            value="Lecture depuis localStorage"
            onclick="lireLocalStorage();"
        />
        <br />
        <input
            type="button"
            id="boutonSuppression"
            value="Suppression de l'entrée dans localStorage"
            onclick="supprimerLocalStorage();"
        />
    </form>

    <!-- Affichage du contenu du système de stockage -->
    <br />
    <span id="intitule"></span><br />
    <span id="nomPersonne"></span><br />
    <span id="prenomPersonne"></span><br />
    <span id="mailPersonne"></span>

    <!-- Affichage du code source -->

```

```

        <br /><br /><br />
        <center>
        <a href="JavaScript:window.location='view-source:' +
        window.location">
            Code source
        </a>
        </center>

    </body>

</html>

```

### **Formulaire à trois boutons**

Commençons par les trois boutons `<input type="button" ...>` présentés dans le formulaire HTML :

```

<!-- Formulaire avec un bouton d'écriture, un bouton de lecture et
un bouton de suppression -->
<form>
    <input type="button" id="boutonEcriture" value="Ecriture dans
    localStorage" onclick="ecrireLocalStorage();">
    <br />
    <input type="button" id="boutonLecture" value="Lecture depuis
    localStorage" onclick="lireLocalStorage();">
    <br />
    <input type="button" id="boutonSuppression" value="Suppression de
    l'entrée dans localStorage" onclick="supprimerLocalStorage();">
</form>

```

Le premier d'entre eux (`id="boutonEcriture"`) déclenche une fonction de nom `ecrireLocalStorage` qui sert à entreposer trois informations (le nom, le prénom et l'adresse mail d'une personne dans un système de stockage `localStorage`).

Le second bouton (`id="boutonLecture"`) a un rôle inverse, il déclenche une fonction `lireLocalStorage` qui ira lire dans le système de stockage `localStorage` les trois informations suscitées.

Enfin le dernier bouton (`id="boutonSuppression"`) par l'intermédiaire de la fonction `supprimerLocalStorage` effacera les données entreposées dans le système de stockage.

### **Affichage du contenu du localStorage**

Un compte rendu de contenu du système de stockage est aussi affiché à tout moment sous le formulaire :

```

<!-- Affichage du contenu du système de stockage -->
Contenu du localStorage :<br />
<span id="nomPersonne"></span><br />
<span id="prenomPersonne"></span><br />
<span id="mailPersonne"></span>

```

## **Fonction JavaScript écrireLocalStorage**

Passons maintenant à l'étude détaillée des trois fonctions évoquées.

Dans la fonction `ecrireLocalStorage` un test est effectué pour savoir si le navigateur supporte la gestion d'un `localStorage` :

```
/* Test possibilité d'utiliser localStorage */
if (typeof localStorage != "undefined")
{
    /* Traitement possible ... */
}
else
{
    /* Message d'erreur (pas de possibilité de stockage localStorage) */
    alert("localStorage n'est pas supporté");
}
};
```

Dans le cas favorable, nous avons consécutivement le placement des informations dans le `localStorage` (cas du `nomPersonne` ci-après) :

```
/* Stockage du nom (nomPersonne) dans le stockage localStorage */
nomPersonne = "VIGOUROUX";
localStorage.setItem("nomPersonne", nomPersonne);
```

Rappelons que la variable mémoire `nomPersonne` à laquelle est affectée arbitrairement "VIGOUROUX" a été déclarée en début de script :

```
/* Déclarations de variables communes à toutes les fonctions */
var nomPersonne;
```

Bien évidemment dans une application plus réaliste, les données ne seraient pas uniquement des constantes, mais par exemple des données de formulaires ou encore des résultats de calcul.

Comme annoncé en début de présentation de cet exemple, nous avons aussi une actualisation de trois champs `<span> ... </span>` sous le formulaire à trois boutons pour afficher le contenu de notre `localStorage` :

```
/* Affichage de contrôle */
document.getElementById("nomPersonne").innerHTML = nomPersonne;
```

## **Fonction JavaScript lireLocalStorage**

Comme dans la fonction `ecrireLocalStorage`, un test est prévu sur la capacité du navigateur à gérer un `localStorage`.

Ensuite la récupération des trois informations stockées dans le `localStorage` par l'intermédiaire de la fonction

écritureLocalStorage est assurée par la méthode getItem de l'objet localStorage. Pour le nom de la personne (nomPersonne), la récupération se fait comme suit :

```
/* Récupération du nom (nomPersonne) depuis le stockage localStorage */  
nomPersonne = localStorage.getItem("nomPersonne");
```

Une fois le contenu du stockage localStorage lu, comme pour la fonction écritureLocalStorage, les trois champs `<span> ... </span>` sont actualisés.

### **Fonction JavaScript supprimerLocalStorage**

Cette dernière fonction a pour objectif de supprimer intégralement le contenu du système de stockage localStorage.

Un test est également prévu sur la capacité du navigateur à gérer un localStorage.

Dans le script présenté, la méthode removeItem de l'objet localStorage est utilisée pour chacune des informations à supprimer.

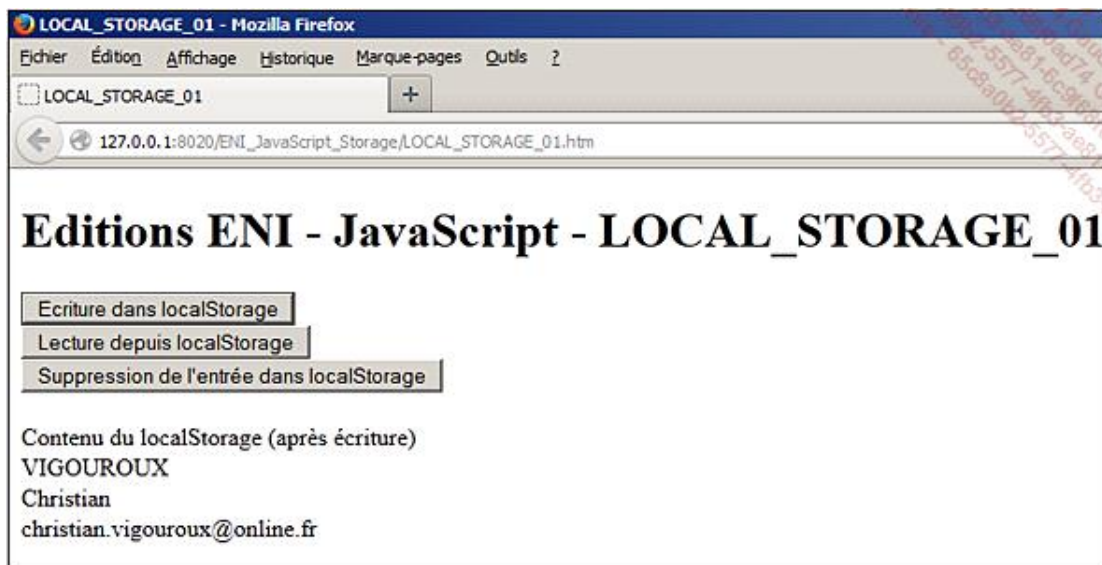
```
/* Suppression des entrées nomPersonne, prenomPersonne et mailPersonne  
dans le localStorage */  
sessionStorage.removeItem("nomPersonne");  
/* ... */
```



Il est également possible de procéder à la suppression complète du système de stockage localStorage par la méthode `session Storage.clear()` ;.

### **Compte rendu d'exécution**

Un clic sur le bouton 'Ecriture dans localStorage' provoque l'affichage suivant :



Après relecture du contenu du `localStorage`, l'affichage devient :



Enfin, après suppression du `localStorage`, l'affichage obtenu est :



## 2. Exemple 2 : Stockage dans le `localStorage` d'un objet JavaScript

Dans ce deuxième exemple, l'objectif est encore d'assurer par des fonctions JavaScript la mémorisation d'informations (nom, prénom et mail d'une personne) dans un système de stockage `localStorage`. Par contre cette fois-ci les trois informations seront placées dans un objet JavaScript. Il nous faudra avoir recours, pour assurer le stockage, à une sérialisation JSON et à l'inverse à une désérialisation pour récupérer les valeurs.

### Script complet

Le code source du script est reproduit intégralement ci-après :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<!--
```



NOM DU SCRIPT : LOCAL\_STORAGE\_02.htm  
REALISATION INFORMATIQUE : Christian VIGOUROUX  
DATE DE CREATION : 01/01/2014  
DATE DE DERNIERE MODIFICATION : 01/01/2014  
OBJET : Stockage avec utilisation de localStorage et JSON  
-->

<!-- Début script HTML -->  
<html>

<!-- Début en-tête script HTML -->  
<head>

<!-- Balise meta -->  
<meta http-equiv="Content-Type"  
content="text/html; charset=utf-8" />

<!-- Titre du script HTML -->  
<title>LOCAL\_STORAGE\_02</title>

<!-- Début script JavaScript -->  
<script type='text/javascript'>

```
/* Fonction ecrireLocalStorage */
function ecrireLocalStorage()
{
    /* Test possibilité d'utiliser localStorage et JSON */
    if (typeof localStorage != "undefined" && JSON)
    {
        /* Définition d'un objet JavaScript
        coordonneesPersonne */
        var coordonneesPersonne =
        {
            nom:document.getElementById("nom").value,
            prenom:document
                .getElementById("prenom").value,
            mail:document.getElementById("mail").value
        };
        /* Sérialisation en un objet JSON
        de nom identite */
        localStorage.setItem("identite",
            JSON.stringify(coordonneesPersonne));
        /* Affichage de contrôle */
        alert("Enregistrement dans localStorage effectué");
    }
    else
    {
        /* Message d'erreur (pas de possibilité
        de stockage localStorage) */
        alert("localStorage n'est pas supporté");
    }
};

/* Fonction supprimerLocalStorage */
function supprimerLocalStorage()
```

```

{
    /* Test possibilité d'utiliser localStorage et JSON */
    if (typeof localStorage != "undefined" && JSON)
    {
        /* Suppression de l'entrée identite
        dans le localStorage */
        /* NB : sessionStorage.clear(); supprimerait
        également l'ensemble des entrées
        du localStorage */
        sessionStorage.removeItem("identite");
        /* Mise à jour des champs d'affichage
        (<input type="texte" ... />) */
        document.getElementById("nom").value = "";
        document.getElementById("prenom").value = "";
        document.getElementById("mail").value = "";
        /* Affichage de contrôle */
        alert("Suppression de l'entrée identite
        dans le localStorage effectuée");
    }
    else
    {
        /* Message d'erreur (pas de possibilité
        de stockage localStorage) */
        alert("localStorage n'est pas supporté");
    }
};

</script>

</head>

<!-- Début section body du script HTML -->
<body>

    <!-- Titre du traitement -->
    <h1>Editions ENI - JavaScript - LOCAL_STORAGE_02</h1>

    <!-- Début script JavaScript -->
    <script type='text/javascript'>

        /* Affichage du nom du script */
        alert("LOCAL_STORAGE_02");

    </script>

    <!-- Tableau de saisie et d'affichage des champs nom,
    prénom et mail -->
    <table>
        <tr>
            <td>Nom</td>
            <td>
                <input
                    type="text"
                    id="nom"
                    value=""

```

```

        size="40"
    />
</td>
</tr>
<tr>
    <td>Prénom</td>
    <td>
        <input
            type="text"
            id="prenom"
            value=""
            size="40"
        />
    </td>
</tr>
<tr>
    <td>Mail</td>
    <td>
        <input
            type="text"
            id="mail"
            value=""
            size="40"
        />
    </td>
</tr>
</table>
<!-- Formulaire avec boutons d'écriture,
de lecture et de suppression -->
<br />
<form>
    <input
        type="button"
        id="boutonEcriture"
        value="Ecriture dans localStorage"
        onclick="ecrireLocalStorage();"
    />
    <br />
    <input
        type="button"
        id="boutonLecture"
        value="Lecture depuis localStorage"
    />
    <br />
    <input
        type="button"
        id="boutonSuppression"
        value="Suppression de l'entrée dans localStorage"
        onclick="supprimerLocalStorage();"
    />
</form>

<!-- Début script JavaScript -->
<script type='text/javascript'>

```

```

/* Lecture depuis le localStorage */
document.getElementById('boutonLecture').onclick
= function() {
    /* Test possibilité d'utiliser
    localStorage et JSON */
    if(typeof localStorage != "undefined" && JSON)
    {
        /* Désérialisation de l'objet JSON lu */
        var coordonneesPersonne
        = JSON
        .parse(localStorage
        .getItem("identite"));
        /* Mise à jour des champs d'affichage
        (<input type="texte" ... />) */
        document.getElementById("nom").value
        = coordonneesPersonne.nom;
        document.getElementById("prenom").value
        = coordonneesPersonne.prenom;
        document.getElementById("mail").value
        = coordonneesPersonne.mail;
        /* Affichage de contrôle */
        alert("Lecture depuis localStorage effectuée");
    }
    else
    {
        /* Message d'erreur (pas de possibilité
        de stockage localStorage) */
        alert("localStorage n'est pas supporté");
    }
};
</script>

<!-- Affichage du code source -->
<br /><br /><br />
<center>
<a href="JavaScript:window.location='view-source:' + window.location">
    Code source
</a>
</center>

</body>

</html>

```

### **Saisie des données à placer dans le localStorage**

Dans ce script plus évolué que celui de l'exemple n°1, une saisie est proposée à l'utilisateur afin qu'il puisse renseigner les trois informations (nom, prénom et mail d'une personne) et les mémoriser dans le système de stockage localStorage :

```

<!-- Tableau de saisie et d'affichage des champs nom, prénom et mail -->
<table>

```

```

<tr>
  <td>Nom</td>
  <td><input type="text" id="nom" value="" size="40"></td>
</tr>
<tr>
  <td>Prénom</td>
  <td><input type="text" id="prenom" value="" size="40"></td>
</tr>
<tr>
  <td>Mail</td>
  <td><input type="text" id="mail" value="" size="40"></td>
</tr>
</table>

```

## **Formulaire d'appel des fonctions JavaScript**

Le déclenchement des trois fonctions est assuré par un formulaire à trois boutons :

```

<!-- Formulaire avec boutons d'écriture, de lecture et de suppression -->
<br />
<form>
  <input type="button" id="boutonEcriture" value="Ecriture dans
  localStorage" onclick="ecrireLocalStorage();" />
  <br />
  <input type="button" id="boutonLecture" value="Lecture depuis
  localStorage" />
  <br />
  <input type="button" id="boutonSuppression" value="Suppression de
  l'entrée dans localStorage" onclick="supprimerLocalStorage();" />
</form>

```

Petite particularité dans ce script, les fonctions `ecrireLocalStorage` et `supprimerLocalStorage` sont placées dans la section HTML `<head> ... </head>` alors que la fonction de lecture n'est pas explicitement nommée et est placée dans la section HTML `<body> ... </body>` sous le formulaire lui-même.

Étudions maintenant dans le détail chacune de ces fonctions.

### **Fonction `ecrireLocalStorage`**

Dans la fonction `ecrireLocalStorage`, le code débute par un test sur la possibilité d'utiliser le `localStorage` dans le navigateur :

```

/* Test possibilité d'utiliser localStorage et JSON */
if (typeof localStorage != "undefined" && JSON)

```

Vous le noterez, le code est légèrement différent de celui déjà vu dans l'exemple n°1. Un test complémentaire sur la prise en compte de JSON est également prévu.



Vous trouverez une définition précise de JSON sur le site officiel : <http://www.json.org/>

Voyons ensuite la séquence de code de création d'un objet JavaScript `coordonneesPersonne` construit à partir des trois champs de saisie (`<input type="text" ... >`):

```
/* Définition d'un objet JavaScript coordonneesPersonne */
var coordonneesPersonne =
{
    nom:document.getElementById("nom").value,
    prenom:document.getElementById("prenom").value,
    mail:document.getElementById("mail").value
};
```

Les objets JavaScript ne sont pas sauvegardables en l'état dans un système de stockage `localStorage` car seules des chaînes de caractères peuvent être prises en compte. Il convient de transformer notre objet `coordonneesPersonne` en une chaîne de caractères avant le stockage. Cette opération s'appelle une **sérialisation JSON**.

Dans les langages Web, comme JavaScript ou PHP, la **sérialisation** consiste à transformer une variable composite (comme un tableau ou un objet) en une chaîne de caractères.

Le code pour réaliser cette opération est le suivant :

```
/* Sérialisation en un objet JSON de nom identite */
localStorage.setItem("identite", JSON.stringify(coordonneesPersonne));
```

La méthode `setItem` est employée une seule fois, les trois informations sont placées dans une chaîne de caractères unique (`identite`). La **sérialisation** est assurée par la méthode `stringify`.

Les navigateurs récents (y compris Microsoft Internet Explorer depuis sa version 8) supportent la fonction `JSON.stringify`.

À titre indicatif pour la saisie de données suivantes :

- Nom : VIGOUROUX
- Prénom : Christian
- Mail : christian.vigouroux@online.fr

la **sérialisation** donnerait :

```
{
    "nom": "VIGOUROUX",
    "prenom": "Christian",
    "mail": "christian.vigouroux@online.fr"
}
```

## **Fonction supprimerLocalStorage**

Dans cette fonction, le code débute aussi par un test sur la possibilité d'utiliser le `localStorage` et `JSON` dans le navigateur.

La suppression de l'item `identite` sauvegardé par l'intermédiaire de la fonction `ecrireLocalStorage` s'effectue comme suit :

```
/* Suppression de l'entrée identite dans le localStorage */
/* NB : sessionStorage.clear(); supprimerait également l'ensemble des
entrées du localStorage */
sessionStorage.removeItem("identite");
```

Un effacement des champs d'affichage est aussi prévu :

```
/* Mise à jour des champs d'affichage (<input type="texte" ... />) */
document.getElementById("nom").value = "";
/* ... */
```

### **Lecture des données depuis le localStorage**

Pour ce traitement, il n'a pas été prévu de fonction dédiée dans la section HTML `<head> ... </head>`.

Le code source de cette fonctionnalité a été placé à la fin de la section HTML `<body> .. </body>`.

La syntaxe :

```
document.getElementById('boutonLecture').onclick = function() {
    /* ... */
}
```

associe au bouton identifié par `boutonLecture` une fonction (non nommée dans ce cas) qui se déclenchera sur clic.

Dans le code de la fonction, vous retrouvez le test habituel portant sur la possibilité d'utiliser le `localStorage` et `JSON` dans le navigateur.

Ensuite la désérialisation de l'objet `JSON` stocké dans le système de stockage `localStorage` est réalisée, comme suit :

```
/* Désérialisation de l'objet JSON lu */
var coordonneesPersonne = JSON.parse(localStorage.getItem("identite"));
```

`JSON.parse()` réalise un traitement inverse de la fonction `JSON.stringify()`. La méthode attend pour paramètre une chaîne de caractères `JSON` (`identite` dans notre cas) et la transforme en objet JavaScript. Cette transformation s'appelle une désérialisation.

Une mise à jour des champs d'affichage est ensuite aussi prévue :

```

/* Mise à jour des champs d'affichage (<input type="texte" ... />) */
document.getElementById("nom").value = coordonneesPersonne.nom;
/* ... */

```

## Compte rendu d'exécution

Commençons par renseigner les trois champs de saisie comme prévu précédemment et validons par un clic sur le bouton 'Ecriture dans localStorage' :

LOCAL\_STORAGE\_02 - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

LOCAL\_STORAGE\_02

127.0.0.1:8020/ENI\_JavaScript\_Storage/LOCAL\_STORAGE\_02.htm

## Editions ENI - JavaScript - LOCAL\_STORAGE\_02

Nom

Prénom

Mail

Ecriture dans localStorage

Lecture depuis localStorage

Suppression de l'entrée dans localStorage

Effaçons ensuite les champs de saisie :

LOCAL\_STORAGE\_02 - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

LOCAL\_STORAGE\_02

127.0.0.1:8020/ENI\_JavaScript\_Storage/LOCAL\_STORAGE\_02.htm

## Editions ENI - JavaScript - LOCAL\_STORAGE\_02

Nom

Prénom

Mail

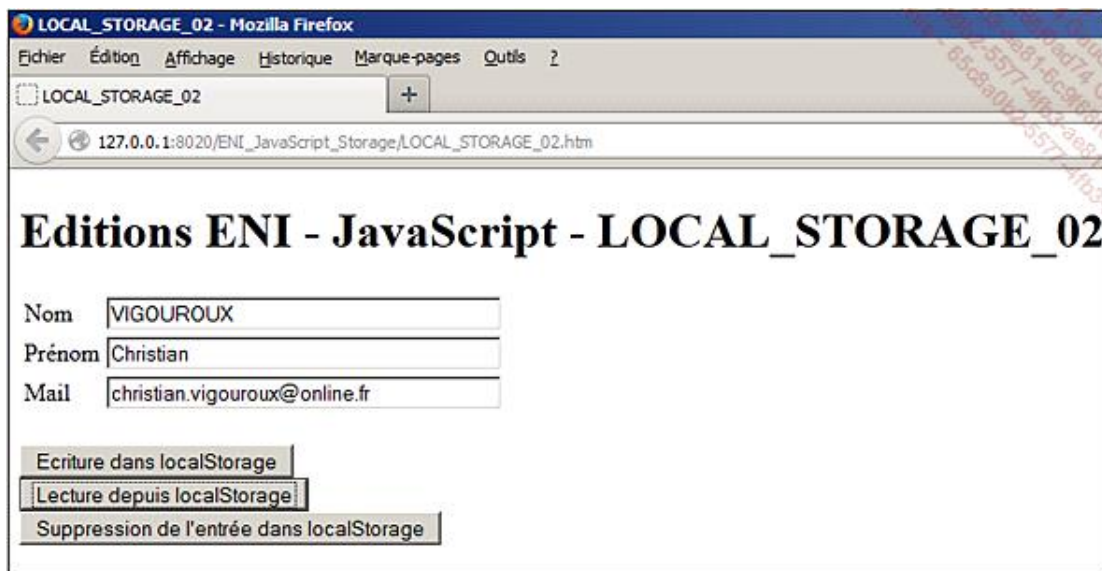
Ecriture dans localStorage

Lecture depuis localStorage

Suppression de l'entrée dans localStorage

Terminons par la relecture du localStorage par un clic sur le bouton 'Lecture depuis localStorage' :





Vous pouvez constater que les valeurs ont été restaurées dans les champs de saisie.