

# TP JavaScript — Système de gestion d'examens avec LocalStorage

**Durée :** 90 minutes

**Niveau :** SMI S4

**Objectifs :**

- Créer et manipuler des objets JavaScript.
- Utiliser le DOM pour ajouter des éléments dynamiquement.
- Travailler avec le localStorage pour stocker des données persistantes.

**Consignes de travail :**

- Travailler en binôme ou trinôme (selon la disponibilité des ordinateurs)
- Utiliser votre compte GitHub
- Avant de terminer, préparer une vidéo courte d'une minute au maximum pour montrer le résultat.
- A la fin du TP, partager avec le professeur votre repository et vidéo via ce formulaire : <https://forms.gle/erWr2jijvwWRefba6>

---

## Partie 1 — Création du formulaire d'ajout d'un examen (30 min)

### TO DO 1 : Créez le fichier index.html

Créez un fichier index.html contenant un formulaire avec les champs suivants :

- Nom de l'examen
- Durée de l'examen (en minutes)
- Description
- Nom du propriétaire

Voici un exemple de structure de base :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Ajouter un Examen</title>
</head>
<body>
  <h1>Créer un nouvel examen</h1>
  <form id="form-examen">
    <input type="text" id="nom" placeholder="Nom de l'examen" required><br>
    <input type="number" id="duree" placeholder="Durée (minutes)"
required><br>
    <textarea id="description" placeholder="Description"></textarea><br>
```

```
<input type="text" id="proprietaire" placeholder="Nom du propriétaire"
required><br>
<button type="submit">Ajouter l'examen</button>
</form>

<script src="script.js"></script>
</body>
</html>
```

## TO DO 2 : Écrire le script pour enregistrer l'examen dans le localStorage

Créez un fichier script.js et ajoutez le code suivant pour gérer le formulaire et enregistrer l'examen dans le localStorage :

```
// TODO : Ajouter un écouteur d'événement sur le formulaire
document.getElementById('form-examen').addEventListener('submit', function(e)
{
    e.preventDefault();

    // TODO : Récupérer les valeurs du formulaire
    const examen = {
        nom: document.getElementById('nom').value,
        duree: parseInt(document.getElementById('duree').value),
        description: document.getElementById('description').value,
        proprietaire: document.getElementById('proprietaire').value,
        questions: []
    };

    // TODO : Sauvegarder l'examen dans le localStorage sous une clé basée sur
    le nom du propriétaire
    const examsKey = 'examens_' + examen.proprietaire;
    const exams = JSON.parse(localStorage.getItem(examsKey)) || [];
    exams.push(examen);
    localStorage.setItem(examsKey, JSON.stringify(exams));

    alert('Examen ajouté avec succès !');
    this.reset();
});
```

---

### Réflexion :

- Comment gérer plusieurs examens pour un même propriétaire ?
  - Que se passe-t-il si deux utilisateurs ont le même nom ? Comment éviter un conflit ?
-

## Partie 2 — Ajouter des questions et des propositions de réponse (40 min)

### TO DO 1 : Créer le fichier ajouter-question.html

Créez un fichier ajouter-question.html avec un formulaire permettant d'ajouter une question à un examen. Ce formulaire doit inclure :

- Un champ pour l'énoncé de la question
- Un champ pour la durée de la question (en secondes)
- Un champ pour les points
- Un bouton pour ajouter dynamiquement des propositions de réponse.

Voici la structure de base :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Ajouter une question</title>
</head>
<body>
  <h1>Ajouter une question à un examen</h1>

  <form id="form-question">
    <input type="text" id="proprietaire" placeholder="Nom du propriétaire"
required><br>
    <input type="text" id="nom-examen" placeholder="Nom de l'examen"
required><br>
    <textarea id="enonce" placeholder="Énoncé de la question"
required></textarea><br>
    <input type="number" id="duree" placeholder="Durée (secondes)"
required><br>
    <input type="number" id="points" placeholder="Points" required><br>

    <div id="propositions"></div>
    <button type="button" id="add-proposition">Ajouter une
proposition</button><br><br>

    <button type="submit">Ajouter la question</button>
  </form>

  <script src="ajouter-question.js"></script>
</body>
</html>
```

### TO DO 2 : Ajouter des propositions de réponses dynamiquement

Dans ajouter-question.js, ajoutez le code pour permettre à l'utilisateur d'ajouter des propositions de réponse :

```
// TODO : Ajouter un écouteur d'événement pour ajouter des propositions dynamiquement
document.getElementById('add-proposition').addEventListener('click', () => {
    const container = document.createElement('div');
    const input = document.createElement('input');
    const checkbox = document.createElement('input');
    checkbox.type = 'checkbox';

    input.type = 'text';
    input.placeholder = 'Proposition';

    container.appendChild(checkbox);
    container.appendChild(input);
    document.getElementById('propositions').appendChild(container);
});
```

### TO DO 3 : Enregistrer la question dans l'examen

Toujours dans ajouter-question.js, ajoutez le code suivant pour enregistrer la question dans le localStorage :

```
// TODO : Ajouter un écouteur d'événement pour enregistrer la question
document.getElementById('form-question').addEventListener('submit',
function(e) {
    e.preventDefault();

    const enonce = document.getElementById('enonce').value;
    const duree = parseInt(document.getElementById('duree').value);
    const points = parseInt(document.getElementById('points').value);
    const proprietaire = document.getElementById('proprietaire').value;
    const nomExamen = document.getElementById('nom-examen').value;

    const propositions = [];
    document.querySelectorAll('#propositions div').forEach(div => {
        const texte = div.querySelector('input[type="text"]').value;
        const correcte = div.querySelector('input[type="checkbox"]').checked;
        propositions.push({ texte, correcte });
    });

    // TODO : Trouver l'examen existant à partir du localStorage
    const examsKey = 'examens_' + proprietaire;
    const exams = JSON.parse(localStorage.getItem(examsKey)) || [];
    const exam = exams.find(e => e.nom === nomExamen);
    if (!exam) {
        alert('Examen non trouvé !');
        return;
    }
}
```

```

// TODO : Ajouter la question à l'examen et mettre à jour le localStorage
const question = { enonce, duree, points, propositions };
exam.questions.push(question);

localStorage.setItem(examsKey, JSON.stringify(exams));

alert('Question ajoutée avec succès !');
this.reset();
document.getElementById('propositions').innerHTML = '';
});

```

### Réflexion :

- Comment vérifier que l'examen existe avant d'ajouter la question ?
- Peut-on empêcher l'ajout d'une question sans proposition correcte ?

## Partie 3 — Afficher les examens enregistrés (15 min)

### TO DO 1 : Créez examens.html pour afficher les examens

Créez un fichier examens.html qui permet de récupérer et afficher tous les examens d'un propriétaire à partir du localStorage.

```

<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Liste des examens</title>
</head>
<body>
  <h1>Examens enregistrés</h1>
  <input type="text" id="proprietaire" placeholder="Nom du propriétaire">
  <button id="afficher">Afficher</button>
  <div id="resultat"></div>

  <script>
    // TODO : Ajouter un écouteur d'événement pour afficher les examens du
    propriétaire
    document.getElementById('afficher').addEventListener('click', () => {
      const prop = document.getElementById('proprietaire').value;
      const exams = JSON.parse(localStorage.getItem('examens_' + prop)) || [];
      const container = document.getElementById('resultat');
      container.innerHTML = '';

      // TODO : Afficher les examens et leurs questions
      exams.forEach((exam, i) => {

```

```
const div = document.createElement('div');
div.innerHTML = `<h2>${exam.nom} (${exam.duree} min)</h2>
                <p>${exam.description}</p>
                <ul>${exam.questions.map(q =>
`<li>${q.enonce}</li>`).join('')}</ul>`;
                container.appendChild(div);
            });
        });
    </script>
</body>
</html>
```

---

### **Réflexion :**

- Comment afficher dynamiquement les questions de chaque examen ?
- Que faire si le propriétaire n'a pas d'examen enregistré ?