

TP3
Javascript

Exercice 1 :

créer une interface web permettant de calculer le prix TTC d'un produit en utilisant JavaScript. L'interface inclut un formulaire pour entrer le prix HT et le taux de TVA.

<pre> <!DOCTYPE html> <html lang="fr"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Calcul de Prix TTC</title> </head> <body> <div class="container"> <h1>Calcul TTC</h1> <form id="ttcForm"> <label for="prixHt">Prix HT (€) :</label> <input type="number" id="prixHt" placeholder="Entrez le prix HT" required step="0.01"> <label for="tauxTva">Taux de TVA (%) :</label> <input type="number" id="tauxTva" placeholder="Entrez le taux de TVA" required step="0.01"> <button type="submit">Calculer TTC</button> </form> <div class="result" id="resultat"></div> </div> </pre>	<pre> <script> // Gestionnaire d'événement pour le formulaire document.getElementById('ttcForm').addEventListener('submit', function(event) { event.preventDefault(); // Empêche le rechargement de la page // Récupération des valeurs const prixHt = parseFloat(document.getElementById('prixHt').value); const tauxTva = parseFloat(document.getElementById('tauxTva').value); // Vérification des données if (isNaN(prixHt) isNaN(tauxTva) prixHt < 0 tauxTva < 0) { document.getElementById('resultat').textContent = "Veuillez entrer des valeurs valides."; return; } // Calcul du prix TTC const prixTtc = prixHt + (prixHt * tauxTva / 100); // Affichage du résultat document.getElementById('resultat').textContent = `Le prix TTC est de : \${prixTtc.toFixed(2)} €`; }); </script> </body> </html> </pre>
---	--

Exercice 2 :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Gestion des Tâches</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
      background-color: #f5f5f5;
    }
    .container {
      max-width: 500px;
      margin: 0 auto;
      background-color: white;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    }
    h1 {
      text-align: center;
      color: #333;
    }
    form {
      display: flex;
      gap: 10px;
      margin-bottom: 20px;
    }
    input[type="text"] {
      flex: 1;
      padding: 10px;
      font-size: 16px;
      border: 1px solid #ddd;
      border-radius: 4px;
    }
    button {
      padding: 10px 15px;
      font-size: 16px;
      color: white;
      background-color: #007bff;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    button:hover {
      background-color: #0056b3;
    }
    .task-list {
      list-style: none;
      padding: 0;
    }
    .task {
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding: 10px;
      margin-bottom: 10px;
      background-color: #f9f9f9;
      border: 1px solid #ddd;
      border-radius: 4px;
    }
    .task.completed {
      text-decoration: line-through;
      color: #888;
    }
    .task button {
      background-color: #dc3545;
      font-size: 14px;
      padding: 5px 10px;
      border: none;
      border-radius: 4px;
      color: white;
      cursor: pointer;
    }
```

```
.task button:hover {
  background-color: #b02a37;
}
</style>
</head>
<body>
  <div class="container">
    <h1>Gestion des Tâches</h1>
    <form id="taskForm">
      <input type="text" id="taskInput" placeholder="Ajouter une
nouvelle tâche..." required>
      <button type="submit">Ajouter</button>
    </form>
    <ul class="task-list" id="taskList"></ul>
  </div>

  <script>
    // Sélection des éléments HTML
    const taskForm = document.getElementById('taskForm');
    const taskInput = document.getElementById('taskInput');
    const taskList = document.getElementById('taskList');

    // Gestionnaire d'événement pour ajouter une tâche
    taskForm.addEventListener('submit', function(event) {
      event.preventDefault(); // Empêche le rechargement de la page

      // Récupère la valeur de l'entrée
      const taskText = taskInput.value.trim();

      // Vérifie que le champ n'est pas vide
      if (taskText === "") return;

      // Crée un élément de tâche
      const taskItem = document.createElement('li');
      taskItem.classList.add('task');
      taskItem.innerHTML = `
        <span>${taskText}</span>
        <button>Supprimer</button>
      `;

      // Ajoute un événement pour marquer comme terminé
      taskItem.addEventListener('click', function() {
        taskItem.classList.toggle('completed');
      });

      // Gestionnaire pour supprimer une tâche
      taskItem.querySelector('button').addEventListener('click',
function(event) {
  event.stopPropagation(); // Empêche le clic sur l'élément parent
  taskItem.remove();
});

      // Ajoute la tâche à la liste
      taskList.appendChild(taskItem);

      // Réinitialise le champ de saisie
      taskInput.value = "";
    });
  </script>
</body>
</html>
```

Exercice 3 :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Gestion des Tâches Avancée</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
      background-color: #eef2f7;
    }
    .container {
      max-width: 700px;
      margin: 0 auto;
      padding: 20px;
      background-color: white;
      border-radius: 8px;
      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    }
    h1 {
      text-align: center;
      color: #333;
    }
    form {
      display: flex;
      gap: 10px;
      margin-bottom: 20px;
    }
    input[type="text"] {
      flex: 1;
      padding: 10px;
      font-size: 16px;
      border: 1px solid #ddd;
      border-radius: 4px;
    }
    button {
      padding: 10px 15px;
      font-size: 16px;
      color: white;
      background-color: #007bff;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    button:hover {
      background-color: #0056b3;
    }
    table {
      width: 100%;
      border-collapse: collapse;
      margin-top: 20px;
    }
    th, td {
      padding: 10px;
      text-align: left;
      border: 1px solid #ddd;
    }
    th {
      background-color: #f7f7f7;
    }
    tr.completed td {
      text-decoration: line-through;
      color: #888;
    }
    .delete-btn {
      background-color: #dc3545;
      padding: 5px 10px;
      color: white;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    .delete-btn:hover {
      background-color: #b02a37;
    }
  </style></head><body>
```

```
<div class="container">
  <h1>Gestion des Tâches</h1>
  <form id="taskForm">
    <input type="text" id="taskInput" placeholder="Ajouter une
nouvelle tâche..." required>
    <button type="submit">Ajouter</button>
  </form>
  <table>
    <thead><tr><th>#</th><th>Tâche</th> <th>Actions</th> </tr>
    </thead>
    <tbody id="taskTableBody">
      <!-- Les tâches seront insérées ici -->
    </tbody></table></div>
<script>
  const taskForm = document.getElementById('taskForm');
  const taskInput = document.getElementById('taskInput');
  const taskTableBody =
document.getElementById('taskTableBody');

  // Charger les tâches depuis le stockage local
  let tasks = JSON.parse(localStorage.getItem('tasks')) || [];

  // Fonction pour sauvegarder dans le localStorage
  function saveTasks() {
    localStorage.setItem('tasks', JSON.stringify(tasks));
  }

  // Fonction pour afficher les tâches
  function renderTasks() {
    taskTableBody.innerHTML = '';
    tasks.forEach((task, index) => {
      const row = document.createElement('tr');
      if (task.completed) row.classList.add('completed');

      row.innerHTML = `
        <td>${index + 1}</td>
        <td>${task.name}</td>
        <td>
          <button class="toggle-btn">${task.completed ? 'Annuler' :
'Terminer'}</button>
          <button class="delete-btn">Supprimer</button>
        </td>
      `;

      // Bouton pour marquer comme terminé
      row.querySelector('.toggle-btn').addEventListener('click', () =>
{
        tasks[index].completed = !tasks[index].completed;
        saveTasks();
        renderTasks();
      });

      // Bouton pour supprimer la tâche
      row.querySelector('.delete-btn').addEventListener('click', () =>
{
        tasks.splice(index, 1);
        saveTasks();
        renderTasks();
      });

      taskTableBody.appendChild(row);
    });
  }
  // Gestionnaire d'événement pour ajouter une tâche
  taskForm.addEventListener('submit', (event) => {
    event.preventDefault();
    const taskName = taskInput.value.trim();
    if (taskName === '') return;

    tasks.push({ name: taskName, completed: false });
    saveTasks();
    renderTasks();
    taskInput.value = '';
  });

  // Afficher les tâches au chargement initial
  renderTasks();
</script></body></html>
```

Exercice 4 :

<pre> <!DOCTYPE html> <html lang="fr"><head> <title>Gestion de Bibliothèque</title> <style> body { font-family: Arial, sans-serif; margin: 20px; background-color: #f9f9f9; } .container { max-width: 800px; margin: 0 auto; padding: 20px; background-color: white; border-radius: 8px; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); } h1 { text-align: center; color: #333; } form { display: flex; gap: 10px; margin-bottom: 20px; flex-wrap: wrap; } input, select, button { flex: 1; padding: 10px; font-size: 16px; border: 1px solid #ddd; border-radius: 4px; } button { background-color: #007bff; color: white; font-weight: bold; cursor: pointer; } button:hover { background-color: #0056b3; } table { width: 100%; border-collapse: collapse; margin-top: 20px; } th, td { padding: 10px; text-align: left; border: 1px solid #ddd; } th { background-color: #f7f7f7; } .actions button { background-color: #dc3545; padding: 5px 10px; color: white; border: none; border-radius: 4px; cursor: pointer; } </pre>	<pre> .actions button:first-child { margin-right: 5px; } .actions button:hover { opacity: 0.8; } </style> </head> <body> <div class="container"> <h1>Gestion de Bibliothèque</h1> <form id="bookForm"> <input type="text" id="title" placeholder="Titre du livre" required> <input type="text" id="author" placeholder="Auteur du livre" required> <select id="status" required> <option value="Non lu">Non lu</option> <option value="Lu">Lu</option> </select> <button type="submit">Ajouter</button> </form> <table> <thead> <tr> <th>#</th> <th>Titre</th> <th>Auteur</th> <th>Statut</th> <th>Actions</th> </tr> </thead> <tbody id="bookTableBody"> <!-- Les livres seront affichés ici --> </tbody> </table> </div> </pre>	<pre> <script> const bookForm = document.getElementById('bookForm'); const bookTableBody = document.getElementById('bookTableBody'); // Charger les livres depuis le stockage local let books = JSON.parse(localStorage.getItem('books')) []; // Fonction pour sauvegarder les livres dans le localStorage function saveBooks() { localStorage.setItem('books', JSON.stringify(books)); } // Fonction pour afficher les livres function renderBooks() { bookTableBody.innerHTML = ` <td>\${index + 1}</td> <td>\${book.title}</td> <td>\${book.author}</td> <td>\${book.status}</td> <td class="actions"> <button class="toggle-btn">\${book.status === 'Lu' ? 'Non lu' : 'Lu'}</button> <button class="delete-btn">Supprimer</button> </td> `; // Bouton pour changer le statut row.querySelector('.toggle-btn').addEventListener('click', () => { books[index].status = books[index].status === 'Lu' ? 'Non lu' : 'Lu'; saveBooks(); renderBooks(); }); // Bouton pour supprimer un livre row.querySelector('.delete-btn').addEventListener('click', () => { books.splice(index, 1); saveBooks(); renderBooks(); }); bookTableBody.appendChild(row); } // Gestionnaire d'événement pour ajouter un livre bookForm.addEventListener('submit', (event) => { event.preventDefault(); const title = document.getElementById('title').value.trim(); const author = document.getElementById('author').value.trim(); const status = document.getElementById('status').value; if (title === "" author === "") return; books.push({ title, author, status }); saveBooks(); renderBooks(); // Réinitialiser le formulaire bookForm.reset(); }); // Afficher les livres au chargement initial renderBooks(); </script> </body> </html> </pre>
--	---	--