

# Documentation de l'Application Todo List

## Table des matières

1. [Introduction](#)
2. [Structure du projet](#)
3. [Interface utilisateur](#)
4. [Styles CSS](#)
5. [Fonctionnalités JavaScript](#)
6. [Stockage des données](#)
7. [Guide d'utilisation](#)
8. [Personnalisation](#)
9. [Améliorations possibles](#)

## Introduction

Cette application Todo List est une application web légère et intuitive permettant de gérer vos tâches quotidiennes. Développée en HTML, CSS et JavaScript pur (vanilla), elle offre une interface utilisateur moderne avec un design en "glassmorphism" et dispose des fonctionnalités essentielles d'une liste de tâches, tout en restant simple et performante.

### Caractéristiques principales :

- Interface utilisateur moderne avec effets de verre (glassmorphism)
- Ajout, suppression et marquage de tâches comme terminées
- Filtrage des tâches par statut (tout, actif, terminé)
- Persistance des données via le stockage local (localStorage)
- Design responsive adapté à tous les appareils

## Structure du projet

L'application est contenue dans un seul fichier HTML qui intègre :

- Le balisage HTML pour la structure
- Les styles CSS intégrés dans une balise `<style>`
- Le code JavaScript intégré dans une balise `<script>`

Cette approche "tout-en-un" facilite le déploiement et l'utilisation de l'application sans dépendances externes.

# Interface utilisateur

L'interface utilisateur est composée des éléments suivants :

1. **En-tête** : Titre principal "Todo List" avec effet de dégradé
2. **Zone de saisie** : Champ de texte et bouton pour ajouter de nouvelles tâches
3. **Filtres** : Boutons pour filtrer les tâches (Tout, Actifs, Terminés)
4. **Liste des tâches** : Affichage dynamique des tâches avec:
  - o Checkbox pour marquer une tâche comme terminée
  - o Texte de la tâche
  - o Bouton de suppression

## Styles CSS

### Système de variables CSS

L'application utilise des variables CSS pour une gestion cohérente des couleurs et des styles :

```
:root {
  --primary-color: #ff6b6b;
  --secondary-color: #845ef7;
  --accent-color: #ff922b;
  --dark-bg: #0B1B33;
  --card-bg: rgba(11, 27, 51, 0.7);
  --text-color: #e2e8f0;
  --gradient: linear-gradient(135deg, #ff6b6b, #ff922b);
  --glass-bg: rgba(11, 27, 51, 0.6);
  --glass-border: rgba(255, 255, 255, 0.1);
}
```

## Design Glassmorphism

Le design "Glassmorphism" est implémenté à travers ces propriétés clés :

- Fond semi-transparent avec `background: var(--glass-bg)`
- Effet de flou avec `backdrop-filter: blur(15px)`
- Bordures légères avec `border: 1px solid var(--glass-border)`

## Effets et transitions

Des animations et effets subtils améliorent l'expérience utilisateur :

- Transitions douces sur les boutons avec `transition: transform 0.3s ease`
- Effet de surélévation au survol avec `transform: translateY(-2px)`
- Changement d'opacité pour les tâches terminées

# Fonctionnalités JavaScript

## Gestion des données

La structure de données principale est un tableau d'objets `todos`, où chaque tâche est représentée par un objet avec :

- `id` : identifiant unique basé sur le timestamp
- `text` : texte de la tâche
- `completed` : état d'achèvement (booléen)

```
let todos = JSON.parse(localStorage.getItem('todos')) || [];
```

## Principales fonctions

`addTodo()`

Ajoute une nouvelle tâche à la liste en récupérant le texte saisi et en créant un nouvel objet tâche.

```
function addTodo() {  
  const input = document.getElementById('todoInput');  
  const text = input.value.trim();  
  
  if (text) {  
    todos.push({  
      id: Date.now(),  
      text: text,  
      completed: false  
    });  
    input.value = '';  
    saveTodos();  
    renderTodos();  
  }  
}
```

`toggleTodo(id)`

Change l'état d'achèvement d'une tâche lorsque l'utilisateur clique sur sa checkbox.

```
function toggleTodo(id) {
  todos = todos.map(todo => {
    if (todo.id === id) {
      return { ...todo, completed: !todo.completed };
    }
    return todo;
  });
  saveTodos();
  renderTodos();
}
```

`deleteTodo(id)`

Supprime une tâche de la liste lorsque l'utilisateur clique sur le bouton de suppression.

```
function deleteTodo(id) {
  todos = todos.filter(todo => todo.id !== id);
  saveTodos();
  renderTodos();
}
```

`filterTodos(filter)`

Change le filtre actuel et met à jour l'affichage des tâches.

```
function filterTodos(filter) {
  currentFilter = filter;
  document.querySelectorAll('.filter-btn').forEach(btn => {
    btn.classList.remove('active');
  });
  event.target.classList.add('active');
  renderTodos();
}
```

`renderTodos()`

Affiche les tâches en fonction du filtre actuel.

```
function renderTodos() {
  const todoList = document.getElementById('todoList');
  todoList.innerHTML = '';

  let filteredTodos = todos;
  if (currentFilter === 'active') {
    filteredTodos = todos.filter(todo => !todo.completed);
  } else if (currentFilter === 'completed') {
    filteredTodos = todos.filter(todo => todo.completed);
  }

  filteredTodos.forEach(todo => {
    const li = document.createElement('li');
    li.className = `todo-item ${todo.completed ? 'completed' : ''}`;
    li.innerHTML = `
      <input type="checkbox" ${todo.completed ? 'checked' : ''}
        onclick="toggleTodo(${todo.id})">
      <span>${todo.text}</span>
      <button class="delete-btn" onclick="deleteTodo(${todo.id})">x</button>
    `;
    todoList.appendChild(li);
  });
}
```

## Événements

L'application utilise les événements suivants :

- Clic sur le bouton "Ajouter" pour ajouter une tâche
- Pression de la touche "Entrée" dans le champ de saisie pour ajouter une tâche
- Clic sur une checkbox pour marquer une tâche comme terminée
- Clic sur un bouton de filtre pour changer le mode d'affichage
- Clic sur le bouton de suppression pour supprimer une tâche

## Stockage des données

L'application utilise l'API localStorage du navigateur pour persister les données entre les sessions :

```
function saveTodos() {
  localStorage.setItem('todos', JSON.stringify(todos));
}
```

Les tâches sont chargées au démarrage de l'application :

```
let todos = JSON.parse(localStorage.getItem('todos')) || [];
```

# Guide d'utilisation

## Ajouter une tâche

1. Saisissez le texte de la tâche dans le champ de saisie
2. Cliquez sur le bouton "Ajouter" ou appuyez sur la touche "Entrée"

## Marquer une tâche comme terminée

- Cliquez sur la checkbox à gauche de la tâche

## Supprimer une tâche

- Cliquez sur le bouton "×" à droite de la tâche

## Filtrer les tâches

- Cliquez sur l'un des boutons de filtre :
  - "Tout" : affiche toutes les tâches
  - "Actifs" : affiche uniquement les tâches non terminées
  - "Terminés" : affiche uniquement les tâches terminées