

Exercice – Todo List en console

Un exercice de synthèse pour pratiquer les variables, les boucles, les conditions et les tableaux dynamiques – en construisant une vraie mini-application interactive.

Objectif

Créer une **application console interactive** de gestion de tâches (Todo List).

L'utilisateur navigue dans un menu, peut ajouter des tâches, les afficher, les marquer comme faites, et en supprimer.

Aucune classe n'est utilisée – uniquement des `List<string>`, des boucles et des conditions.

Résultat attendu

Écran d'accueil



1. Voir les tâches
2. Ajouter une tâche
3. Marquer comme faite
4. Supprimer une tâche
0. Quitter

Votre choix :

Option 1 – Voir les tâches (liste vide)

--- Mes tâches ---
(aucune tâche pour l'instant)

Option 1 – Voir les tâches (avec tâches)

--- Mes tâches ---
1. [] Faire les courses
2. [] Réviser C#
3. [✓] Promener le chien

[] = à faire

[✓] = terminée

Option 2 – Ajouter une tâche

Nouvelle tâche : Faire les courses

✓ Tâche ajoutée !

Option 3 – Marquer comme faite

--- Mes tâches ---

1. Faire les courses
2. Réviser C#

Numéro de la tâche à cocher : 1

Tâche marquée comme faite !

Option 4 – Supprimer une tâche

--- Mes tâches ---

1. Faire les courses
2. Réviser C#

Numéro de la tâche à supprimer : 1

Tâche supprimée.

Option 0 – Quitter



À bientôt !

Erreur de saisie

Votre choix : 9

 Choix invalide. Réessayez.

Contraintes

- Le menu se **réaffiche automatiquement** après chaque action
 - Si la liste est vide et que l'utilisateur choisit l'option 3 ou 4, afficher :
 Aucune tâche à modifier.
 - Si l'utilisateur entre un numéro de tâche inexistant (option 3 ou 4) :
 Numéro invalide.
 - Aucune classe autorisée – tout dans `Program.cs`
-

Données à utiliser

Tu auras besoin de **deux listes parallèles** :

```
1 | List<string> taches = new List<string>();  
2 | List<bool> faites = new List<bool>();
```

- `taches[i]` contient le texte de la tâche numéro `i`
 - `faites[i]` vaut `true` si la tâche est cochée, `false` sinon
-

Décomposition suggérée

Travaille dans cet ordre – teste après chaque étape :

Étape 1 – Afficher le menu en boucle (`while` infini + `break` sur 0)

Étape 2 – Lire le choix et afficher un message selon l'option choisie (juste un `Console.WriteLine` de test)

Étape 3 – Implémenter "Ajouter une tâche"

Étape 4 – Implémenter "Voir les tâches" (avec `[]` / `[✓]`)

Étape 5 – Implémenter "Marquer comme faite"

Étape 6 – Implémenter "Supprimer une tâche"

Étape 7 – Ajouter toutes les vérifications d'erreur

Notions mobilisées

Notion	Utilisée pour
<code>List<string></code>	stocker les tâches
<code>List<bool></code>	mémoriser l'état de chaque tâche
<code>while</code>	boucle principale du menu
<code>switch</code>	aiguiller selon le choix
<code>for</code>	afficher et parcourir les tâches
<code>int.Parse</code>	convertir le choix et le numéro
interpolation <code>\$""</code>	afficher <code>[]</code> ou <code>[✓]</code> dynamiquement

- **Étape bonus possible** : "sauvegarder dans un fichier `.txt`" si tu veux introduire `File.WriteAllLines` plus tard