






# To-Do List

👤 Créé par	👤 Lynn F
🕒 Heure de création	@8 janvier 2025 09:51
☰ 07/01/25	Rechercher un design pour la To Do List
☰ 07/01/25	Commencer le à coder le projet
☰	<b>CSS :</b> <ul style="list-style-type: none"> <li>• Le CSS définit un style épuré et convivial. Les "post-its" sont stylisés pour ressembler à de vrais post-its jaunes. Des effets de survol et des transitions sont utilisés pour rendre l'interface plus interactive.</li> <li>• Il existe également des règles pour rendre l'application responsive sur différentes tailles d'écran.</li> </ul>
☰ .	<b>JavaScript :</b> <ul style="list-style-type: none"> <li>• Le fichier JavaScript contient toute la logique de l'application : l'ajout de tâches, la modification de leur état, leur suppression, et l'interaction avec le stockage local pour persister les données.</li> <li>• Un gestionnaire d'événements permet de détecter les</li> </ul>

	actions de l'utilisateur (clics sur les boutons) et de mettre à jour l'interface en conséquence.
≡ Affichage des tâches :	<p>Les tâches sont affichées sous forme de "post-its", avec trois catégories distinctes :</p> <ul style="list-style-type: none"> <li>• En cours</li> <li>• Terminées</li> <li>• Supprimées. Chaque "post-it" contient le texte de la tâche et des boutons pour marquer la tâche comme terminée ou la supprimer.</li> </ul>
≡ Améliorations possibles :	<p>Ajouter une fonctionnalité de <b>synchronisation en ligne</b>, pour que les tâches soient stockées sur un serveur et accessibles depuis différents appareils. Permettre aux utilisateurs de <b>classer les tâches</b> par date d'échéance ou par priorité. Ajouter des notifications pour rappeler à l'utilisateur des tâches non terminées.</p>
≡ Code JavaScript :	<p>Le JavaScript permettra de gérer l'interaction avec les boutons et l'affichage.</p> <p><b>Fonctionnement du JavaScript :</b></p> <ol style="list-style-type: none"> <li>1. <b>Sélectionner les éléments HTML :</b> On sélectionne les boutons et l'affichage en utilisant getElementById ou querySelector.</li> <li>2. <b>Gestion des clics sur les boutons :</b> <ul style="list-style-type: none"> <li>◦ Lorsqu'un bouton est cliqué, la valeur du bouton (qu'il s'agisse d'un chiffre ou d'un opérateur) est ajoutée à l'affichage.</li> <li>◦ Le bouton C efface l'affichage.</li> <li>◦ Le bouton = évalue l'expression et affiche le résultat dans l'affichage.</li> </ul> </li> <li>3. <b>Gestion des erreurs :</b> Le script peut vérifier la validité de l'expression (par exemple, éviter les calculs sur une chaîne vide ou malformée).</li> </ol>

<p>≡ Composants de la calculette :</p>	<p><b>Forme de la Calculette :</b></p> <ul style="list-style-type: none"> <li>• La calculette sera intégrée sous forme d'un petit bloc avec une forme de carotte.</li> <li>• L'interface de la calculette sera simple avec un affichage de l'entrée et des boutons pour les calculs.</li> </ul> <p><b>Structure HTML :</b></p> <p>La calculette sera composée :</p> <ul style="list-style-type: none"> <li>• Un <b>affichage</b> pour afficher les résultats des calculs.</li> <li>• Des <b>boutons</b> représentant les chiffres (0-9), les opérateurs (+, -, *, /), un bouton égal (=) pour obtenir le résultat, un bouton pour effacer (C), et un bouton pour réinitialiser (AC).</li> </ul>
<p>≡ Conclusion :</p>	<p>Le projet est une simple <b>to-do list</b> avec une interface agréable et intuitive. Grâce à son utilisation du stockage local et sa structure responsive, l'application est prête à offrir une expérience fluide sur tous les appareils.</p>
<p>≡ Conclusion:</p>	<p>Cette calculette simple mais originale en forme de carotte apporte une touche ludique et pratique au projet de gestion de tâches (To-Do List). Elle est facile à utiliser, et son design s'intègre bien avec le reste de l'interface. L'ajout de cette fonctionnalité permet non seulement de maintenir l'interface interactive, mais aussi d'ajouter un élément amusant et utile à la fois.</p>
<p>≡ Description du code :</p>	<p><b>Gestion des entrées :</b> Les chiffres et opérateurs sont ajoutés à <code>currentInput</code>. <b>Calcul :</b> Quand on appuie sur =, on utilise <code>eval()</code> pour calculer l'expression et mettre à jour l'affichage avec le résultat. <b>Réinitialisation :</b> Lorsque C est pressé, l'affichage est réinitialisé.</p>
<p>≡ Effets visuels :</p>	<p>Chaque "post-it" a un effet de survol (agrandissement et rotation du trombone) pour rendre l'application plus interactive. L'interface comprend également des transitions fluides pour les changements d'état des tâches.</p>

 Filtres :	<p>L'utilisateur peut choisir d'afficher une seule catégorie de tâches (par exemple, uniquement les tâches en cours, terminées, ou supprimées) pour mieux organiser l'affichage.</p>
 Fonctionnalités :	<p><b>Affichage du résultat :</b></p> <ul style="list-style-type: none"> <li>• La calculatrice possède un champ d'affichage (&lt;input id="display"&gt;) qui est mis à jour avec chaque entrée de chiffres ou opérateurs.</li> <li>• Le champ d'affichage est <b>non modifiable</b> par l'utilisateur directement. Il est mis à jour uniquement par les boutons de la calculatrice.</li> </ul> <p><b>Gestion des boutons :</b></p> <ul style="list-style-type: none"> <li>• Les <b>boutons de chiffres</b> (0-9) permettent d'ajouter les chiffres à l'écran.</li> <li>• Les <b>boutons d'opérateurs</b> (+, -, *, /) ajoutent l'opérateur correspondant dans l'affichage.</li> <li>• Le bouton <b>C</b> permet de <b>réinitialiser</b> l'affichage (effacer les chiffres et opérateurs).</li> <li>• Le bouton <b>=</b> effectue le calcul et affiche le résultat.</li> <li>• La <b>fonctionnalité de calcul</b> est réalisée en utilisant JavaScript pour interpréter les entrées et effectuer l'opération correspondante.</li> </ul> <p><b>Calcul :</b></p> <ul style="list-style-type: none"> <li>• Lors de l'appui sur le bouton <b>=</b>, le calcul est effectué en évaluant la chaîne d'opération.</li> <li>• On peut utiliser <code>eval()</code> (avec prudence) ou d'autres techniques pour gérer l'évaluation des expressions.</li> </ul>
 Fonctionnalités principales :	<p><b>Ajout de tâches :</b></p> <ul style="list-style-type: none"> <li>• L'utilisateur peut ajouter des tâches via un champ de texte et un bouton "Ajouter".</li> <li>• Les tâches sont stockées dans le <b>localStorage</b> pour qu'elles soient conservées même après un rafraîchissement de la page.</li> </ul>

≡ Fonctionnalités spécifiques :	Le projet offre également une fonctionnalité permettant d'afficher un seul post-it à la fois en fonction de la catégorie choisie par l'utilisateur.
≡ Gestion des tâches :	<p><b>Marquer comme terminée</b> : L'utilisateur peut cliquer sur un bouton pour marquer une tâche comme terminée.</p> <p><b>Supprimer une tâche</b> : L'utilisateur peut cliquer sur un bouton pour déplacer la tâche vers la liste des tâches supprimées. <b>Supprimer définitivement</b> : Dans la liste des tâches supprimées, l'utilisateur peut supprimer définitivement la tâche.</p>
≡ Interface responsive :	Le design est responsive et s'adapte à tous les écrans (ordinateurs, tablettes, smartphones). L'interface devient plus simple et les éléments prennent plus de place sur les petits écrans pour améliorer l'expérience utilisateur.
≡ Mini calculatrice Carotte	<p><b>Objectif :</b></p> <p>Créer une calculatrice simple mais fonctionnelle en forme de carotte pour un projet de liste de tâches (To-Do List). Elle sera placée en dessous des tâches en cours, et son objectif est de fournir une fonctionnalité de calcul tout en s'intégrant visuellement dans le projet.</p>
≡ Objectif du projet :	<p>Le but de ce projet est de créer une application de gestion de tâches, aussi appelée "To-Do List", où les utilisateurs peuvent ajouter, marquer comme terminées, supprimer ou supprimer définitivement des tâches. Chaque tâche peut être placée dans l'une des trois catégories :</p> <ul style="list-style-type: none"> <li>• <b>En cours</b></li> <li>• <b>Terminées</b></li> <li>• <b>Supprimées</b></li> </ul>
≡ Stockage local :	Les données des tâches (y compris leur état : terminée, supprimée) sont sauvegardées dans <b>localStorage</b> pour permettre la persistance des tâches même après un rafraîchissement ou une fermeture du navigateur.

<p>≡ Structure du projet :</p>	<p><b>HTML :</b></p> <ul style="list-style-type: none"> <li>• La structure HTML comprend une zone d'entrée pour ajouter des tâches, un bouton pour ajouter, et une section pour afficher les tâches sous forme de "post-its" classés dans différentes catégories.</li> </ul>
<p>≡ Styles (CSS) :</p>	<p><b>Couleurs et forme :</b></p> <ul style="list-style-type: none"> <li>• La calculette utilise les couleurs et formes associées au thème de la carotte : <ul style="list-style-type: none"> <li>◦ <b>Orange vif</b> pour la calculette.</li> <li>◦ <b>Vert clair</b> pour les boutons d'opérations.</li> <li>◦ <b>Rouge vif</b> pour le bouton de suppression.</li> </ul> </li> <li>• L'emoji de la carotte 🥕 sera ajouté au-dessus de la calculette pour rappeler l'idée de la forme de la carotte.</li> </ul> <p><b>Responsive Design :</b></p> <ul style="list-style-type: none"> <li>• La calculette est responsive et peut s'adapter à différentes tailles d'écran.</li> <li>• Sur les petits écrans (mobiles et tablettes), les boutons sont disposés dans une grille compacte.</li> </ul>