Affectation de codage du module 1

## Instructions d'affectation

### Idée générale

L'idée de cette mission est de créer une application frontale qui présente à l'utilisateur une zone de texte dans laquelle il peut répertorier les éléments séparés par des virgules qu'il mange habituellement pour le déjeuner. Une fois que cela est entré, l'utilisateur doit cliquer sur le bouton "Vérifier si trop".

Si le nombre d'éléments dans la zone de texte est inférieur ou égal à 3 (par exemple, 1, 2 ou 3), un message doit s'afficher sous la zone de texte indiquant "Enjoy!". Si le nombre d'éléments est supérieur à 3 (4, 5 et plus), le message "Trop !" devrait apparaître sous la zone de texte. (\*Astuce :\* Pour implémenter ce comportement, vous pouvez utiliser la méthode `split`. [Voir la documentation de cette méthode] (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/ Corde/split))

Si la zone de texte est vide et que l'utilisateur clique sur le bouton "Vérifier si trop", le message "Veuillez d'abord saisir les données" devrait s'afficher. 'Vide' ici signifie soit `""` (chaîne vide) ou une chaîne contenant uniquement des espaces. (Astuce : AngularJS `ng-model` effectue déjà le découpage pour vous, vous ne devriez donc rien avoir à faire.)

Un seul message doit être affiché à la fois. En d'autres termes, si vous avez les deux messages "Enjoy!" et "Trop!" apparaître en même temps, c'est une erreur.

Vous n'êtes pas obligé de traiter un cas où il n'y a pas d'élément entre quelques virgules. Par exemple, vous êtes libre de considérer ce cas `item 1, item2,,item3` ou ce cas `item 1, item2, ,item3` comme 4 éléments dans la liste. Cependant, vous pouvez implémenter la vérification d'un élément "vide" en tant que fonctionnalité bonus (non noté). Assurez-vous de mettre un commentaire quelque part à côté de la zone de texte d'entrée indiquant que vous ne considérez PAS un élément vide, c'est-à-dire `, ,` comme un élément vers le décompte.

### Des règles

Si vous enfreignez l'une de ces règles, vous échouerez au devoir :

\* Vous n'êtes pas autorisé à utiliser l'attribut HTML "onclick" pour lier le comportement au bouton. Vous \*\* devez \*\* utiliser le comportement de liaison AngularJS.

\* À aucun moment, votre code Javascript ne doit rechercher \* quoi que ce soit \* dans le DOM du navigateur.

### Pas

Voici ce que vous devrez faire pour terminer le devoir :

1. (Si vous ne l'avez pas déjà fait) Créez un compte GitHub.com et un référentiel que vous utiliserez pour cette classe.

2. (Si vous ne l'avez pas déjà fait) Suivez les instructions de la vidéo de configuration du développement (début du module 1) sur la façon de créer un référentiel et de le configurer de manière à pouvoir héberger et afficher vos pages Web finies sur les pages GitHub, c'est-à-dire GitHub nom de domaine .io. Vous devrez fournir cette URL pour votre évaluation par les pairs.

3. Créez un dossier dans votre référentiel qui servira de dossier conteneur pour votre solution à cette affectation. Tu peux appeler ça comme tu le veux. Par exemple, `module1-solution` ou `mod1\_solution`, etc.

\* Vous pouvez le faire en "clonant" votre référentiel avec "git clone https://www.github.com/your\_repo\_url" sur votre machine locale, en créant le dossier "module1-solution" à la racine du répertoire du référentiel avec un README .txt à l'intérieur du répertoire `module1-solution`. Ensuite, vous feriez `git add .`, suivi de `git commit -m 'New folder'`, suivi de `git push` pour télécharger le nouveau dossier avec le fichier README.txt dans le référentiel GitHub.

4. HTML/CSS pour le devoir

\* Option 1 : Copiez le \*\*contenu\*\* du dossier "assignment1-starter-code" dans le dossier nouvellement créé à l'étape précédente. Si vous avez cloné ce référentiel, le dossier d'affectation 1 se trouve dans `root\_dir\_of\_your\_local\_repo/assignments/assignment1/assignment1-starter-code`

\* Option 2 : Créez vous-même le HTML/CSS. Assurez-vous de nommer le fichier HTML `index.html`. La seule exigence est que votre code HTML ait une zone de texte, un bouton et une région où vous allez afficher le message. Le reste dépend de toi.

5. Importez AngularJS dans votre projet en plaçant une balise `<script>` juste avant la balise `</body>`.

6. Déclarez `ng-app` sur l'élément `html` ou `body`. Nommez votre application "LunchCheck".

7. Créez `app.js` dans votre projet et déclarez un module Angular correspondant à votre déclaration `ng-app`.

8. Retournez à `index.