L3 informatique, 2021-2022, Projet de LFC

Traduction de Markdown en html

Description générale du projet

Le but du projet est d'analyser un fichier Markdown et de le traduire en html.

1 Modalités

Vous devrez rendre le projet par étapes (travaux à rendre toutes les 1, 2 ou 3 semaines selon les étapes). Un sujet spécifique à chaque étape sera mis en ligne sur plubel avec indication de la date limite pour rendre celle-ci.
Lorsque vous rendez une étape, pensez à mentionner, dans le document principal, le nom des personnes ayant participé au travail.
Pour les étapes où il faut rendre un programme, vous devez fournir les instructions nécessaires à la création de l'exécutable et ne pas oublier de remettre tous les modules nécessaires à la compilation et l'exécution de votre programme (.h, .c, etc.). Inutile de fournir un exécutable ; il ne fonctionnera pas. Si vous fournissez un makefile, celui-ci ne doit pas inclure l'exécution de votre programme sur un de vos fichiers tests ni la suppression des fichiers C intermédiaires.
Après chaque étape (sauf la dernière), il vous sera communiqué un corrigé de celle-ci afin que vous puissiez réaliser l'étape suivante sur une base correcte.
Afin que nous puissions tester vos programmes, merci de faire en sorte que ceux-ci puissent être compilés et exécutés sur les machines des salles d'enseignement.

2 Fichiers à analyser

Le langage à analyser pour ce projet est en fait une version simplifiée du langage Markdown (décrite ci-dessous). Vous pouvez consulter les articles concernant ce langage pour avoir davantage d'informations mais restreignez-vous toujours aux options de mise en forme décrits dans cet énoncé pour la réalisation du projet.

Un fichier Markdown contient du texte et des caractères spéciaux qui vont définir la mise en forme à appliquer à certaines parties de ce texte.

Ces caractères spéciaux sont # * et _ (pour ce projet).
En fonction de la façon dont ils sont placés, ils peuvent avoir différentes fonctions.

est utilisé pour définir les titres.
Pour cela, il doit être placé en début de ligne, avec éventuellement entre 1 et 3 espaces avant, et suivi d'au moins un espace.
Pour définir un titre, on écrit un ou plusieurs # (jusqu'à 6 et sans espace entre deux #), en fonction du niveau de titre (# : titre de niveau 1, ## : titre de niveau 2, etc.)
La fin du titre annoncée par les # est déterminée par un retour à la ligne.
En dehors de cette utilisation, tous les autres # seront considérés comme des erreurs lexicales.

Exemples :

Code Markdown	Résultat
texte1 # Premier titre de niveau 1 texte2 ## Premier titre de niveau 2 texte3 ## Second titre de niveau 2 texte4 # Second titre de niveau 1 texte5	texte1 **Premier titre de niveau 1** texte2 **Premier titre de niveau 2** texte3 **Second titre de niveau 2** texte4 **Second titre de niveau 1** texte5

* Et _ sont utilisés pour mettre en évidence certains mots du texte (gras et italique). Dans ce cas, ils sont utilisés par paire.
Une seule étoile ou un seul underscore de part et d'autre d'un texte met celui-ci en italique.
Deux étoiles ou deux underscores (sans espaces entre les deux) de part et d'autre d'un texte met celui-ci en gras.
Trois étoiles ou trois underscores (sans espaces entre deux) de part et d'autre d'un texte met celui-ci en gras et italique.
Remarque : le texte se trouvant entre les – et * de début et les * et _ de fin ne doit pas comporter de ligne vide.

Exemples :

Code Markdown	Résultat
Voici **un texte en gras**, *un texte en italique*, et ***un texte en gras et italique***	Voici **un texte en gras**, *un texte en italique*, et ***un texte en gras et italique***
La même chose avec des underscores : __un texte en gras__, _un texte en italique_, et ___un texte en gras et italique___	La même chose avec des underscores : **un texte en gras**, *un texte en italique*, et ***un texte en gras et italique***
On peut combiner des étoiles et des underscores : premier texte en *__italique et gras__*, second texte en __*gras et italique*__, troisième texte en _**italique et gras**_, quatrième texte en **_gras et italique_**	On peut combiner des étoiles et des underscores : premier texte en ***italique et gras***, second texte en ***gras et italique***, troisième texte en ***italique et gras***, quatrième texte en ***gras et italique***
Et appliquer une mise en évidence progressive : *d'abord en italique ** puis ajout du gras*** ou d'abord en **gras *puis ajout de l'italique***.	Et appliquer une mise en évidence progressive : *d'abord en italique* ***puis ajout du gras*** ou d'abord en **gras *puis ajout de l'italique***

Les retours à la ligne jouent également un rôle important dans les fichiers Markdown.
Un retour à la ligne à la fin d'une ligne de titre annonce la fin du titre.
Une ligne vide indique un changement de paragraphe. Est considérée comme ligne vide une suite de retours à la ligne, éventuellement séparés par des espaces.

Exemples :

Code Markdown	Résultat

`Un paragraphe`
`avec trois`
`retours à la ligne`
`qui n'ont aucun effet sur le résultat.`
 	
`Un nouveau paragraphe.`
	Un paragraphe avec trois retours à la ligne qui n'ont aucun effet sur le résultat.
	
	Un nouveau paragraphe.

* Peut également être utilisé pour introduire les items d'une liste à puce. Dans ce cas, il doit obligatoirement se trouver en début de ligne et être suivi d'au moins 1 espace. Le texte qui suit constitue un item de la liste. Celui-ci s'arrête au prochain item (prochaine * se trouvant en début de ligne) ou à la prochaine ligne vide (pour le dernier item de la liste).

Exemples :

Code Markdown	Résultat

`ci-dessous une liste à puces`
`* 1er item`
`* 2ème item`
`sur 2 lignes`
`* 3ème item`
`* dernier item.`
 	
`Un nouveau paragraphe.`
	ci-dessous une liste à puces
	
	• 1er item
	• 2ème item sur 2 lignes
	• 3ème item
	• dernier item.
	
	Un nouveau paragraphe.

Remarque : Markdown permet également de créer des sous-listes et des listes à numéros mais vous vous restreindrez aux listes à puces simples (tous les items au même niveau).

Tous les * et _ reconnus et mal placés dans le fichier source (qui ne peuvent pas être reconnus comme balise de mise en forme ou de liste à puce du fait de leur emplacement) seront considérés comme faisant partie du texte et traduits tels quels dans le fichier html.

3 Traduction en html (pour la dernière étape du projet)

Définition d'un titre de niveau 1 : <h1> titre</h1>
Définition d'un titre de niveau 2 : <h2> titre</h2>
Etc.
Mise en gras d'un texte : texte
Mise en italique d'un texte : texte
Changement de paragraphe
Définition d'une liste à puces :

 item
 item
…

4 Etapes du projet

4.1 Analyse lexicale

La première étape consistera à établir la liste des unités lexicales du langage et donner la description de celles-ci.
Cette liste sera utilisée lors de la seconde étape pour l'écriture du programme lex qui permettra de générer l'analyseur lexical de votre compilateur.

4.2 Analyse syntaxique

Les étapes suivantes consisteront à décrire de façon formelle la syntaxe d'un fichier Markdown et écrire le programme yacc à partir duquel sera généré l'analyseur syntaxique du compilateur. Ces étapes permettront essentiellement de vérifier si les balises de mise en forme sont utilisées correctement.

4.3 Analyse sémantique/traduction

Les dernières étapes consisteront à insérer dans les programmes lex et yacc les instructions permettant de traduire le fichier Markdown en html.