

Table des matières

1 Utilisation	2	2 Créer ou modifier un modèle	3
1.1 Généralités	2	2.1 Insérer un nouveau champ	3
1.2 Configuration	2	2.2 Ajouter des valeurs à un champ	3
1.3 Structure des fichiers modèles	2	2.3 Paramétrer la création de fichiers auxiliaires	3

Lien de téléchargement du logiciel : [Repo de BFCours](#)

Vocabulaire utilisé

- | | | |
|--|--|---|
| — pré-compléter un modèle (p. 2) | — création de nouveaux modèles (p. 3) | — chaîne de caractères (p. 3) |
| — création ou non d'un dossier (p. 2) | — modification des modèles existants (p. 3) | — liste de chaînes de caractères (p. 3) |
| — renseigner le chemin vers l'exécutable de votre distribution (p. 2) | — implémenter les champs pré remplis (p. 3) | — paramétrer l'inclusion de documents secondaires (p. 3) |
| — % instructions (p. 2) | — python (p. 3) | |
| — bfcours (p. 3) | — document_cfg/commands.txt (p. 3) | |

1. Utilisation

1.1 Généralités

L'assistant de création de document est un outil permettant de faciliter la **création des documents** nécessaires aux activités d'enseignement.

Il propose une interface simple permettant de **pré-compléter un modèle LaTeX**.

Il permet la **création ou non d'un dossier** contenant les documents nécessaires au fonctionnement du modèle sélectionné.

The screenshot shows the CTK application window. At the top, there's a 'Sélectionnez un modèle :' dropdown menu with 'Cours' selected. Below it, 'Titre du document :' has 'Doc_gestionnaire_modelis' entered. A checkbox 'Créer un dossier pour le document :' is checked. A blue button 'Ouvrir les Préférences' is visible. The main area contains several input fields: 'niveau :' (Prof), 'theme :' (Gestionnaire de modèles de BFcours), 'type_etablissement :' (Boum), 'nom_etablissement :' (Factory), 'type_document_principal :' (Documentation), 'contenu_principal :' (\input{enonce}), 'type_document_secondaire :' (Pour impression), and 'contenu_secondaire :'. A 'Soumettre' button is at the bottom right.

1.2 Configuration

L'utilisateur est libre de choisir le logiciel nécessaire à l'ouverture du dossier produit par l'assistant. Pour cela, il faut ouvrir l'interface de configuration et **renseigner le chemin vers l'exécutable de votre distribution**.

 Il est possible que des dysfonctionnements apparaissent si l'IDE n'est pas VSCode.

The screenshot shows the 'Préférences' window. It has a toggle switch 'Ouvrir dans VSCode' which is turned on. Below it is a text field 'Chemin vers VSCode' containing the path 'C:\Users\Utilisateur\AppData\Local\Programs\Microsoft VS Code\Code.exe'. There's a color selection bar with 'Sombre' selected. At the bottom is a blue 'Enregistrer' button.

1.3 Structure des fichiers modèles

De manière tout à fait décevante, le logiciel fonctionne en interprétant les marqueurs **% instructions** contenu dans un document.

Il est donc possible de fournir n'importe quel document \LaTeX pour peu qu'on ait au préalable retiré les marqueurs « % » inutiles.

Ensuite, on inscrit des instructions dans ces commentaires et le logiciel s'adaptera naturellement à la structure de votre document.

Les interactions possible sont décrites dans la partie suivante.

Les modèles pré existants nécessitent l'utilisation du package **bfcours**, mais il est tout à fait possible de l'utiliser sans ce package en **modifiant les modèles**

2. Créer ou modifier un modèle

Le logiciel facilite la **création de nouveaux modèles** et la **modification des modèles existants**, de la façon la plus simple possible (que j'ai imaginé, naturellement...).

L'utilisateur familier avec \LaTeX pourra inspecter les modèles existants pour modifier les interactions basiques avec le gestionnaire.

Les modèles se trouvent dans le dossier **document_model/<nom_du_modèle.tex**

2.1 Insérer un nouveau champ

On peut **insérer un nouveau champ** à compléter de deux manières :

- Par un % **<nom du champ> : <valeurs par défaut séparées par des virgules>**
Cette syntaxe créera un champ dans l'interface avec un champ de texte précomplété avec les valeurs choisies.
- Par un % **<nom du champ>**
Lorsqu'un champ est **déjà créé**, on peut y faire référence sans mettre les double points, et cette partie sera remplacée par le nom du champ en question qui aura été généré.

2.2 Ajouter des valeurs à un champ

Il y a deux façons d' **implémenter les champs pré remplis** par le logiciel :

- Par une liste de valeurs séparées par des **virgules**.
Le logiciel proposera un champ remplissage qui développe toutes les propositions dans une liste déroulante.
- Par la syntaxe **commande(nom_de_la_commande(<paramètres>))**
Pour le lecteur familier avec **python** principalement.
Si la commande est définie dans le fichier **document_cfg/commandes.txt**, le logiciel l'exécutera et remplira le champ avec la ou les valeurs retournées par la commande.
Se référer aux commandes déjà implémentées pour comprendre les standards : Le plus souvent, il s'agit de retourner une **chaîne de caractères** ou une **liste de chaines de caractères**.

2.3 Paramétrer la création de fichiers auxilliaires

Il est possible de **paramétrer l'inclusion de documents secondaires** via l'utilisation de :
% **contenu_du_champ** : **\input{nom_donne_au_document}, \input{autre_nom_possible}**

Le logiciel interprétera ces déclarations et créera automatiquement les fichiers.