

## Table des matières

|  |          |
|--|----------|
| <b>I. Découvrir ezTex !</b>                                  | <b>1</b> |
| I.1. Introduction . . . . .                                  | 1        |
| I.2. L'installer ? . . . . .                                 | 1        |
| <b>II. Comment utiliser ezTex ?</b>                          | <b>2</b> |
| II.1. La création et compilation du document ezTex . . . . . | 2        |
| II.1.a La preface . . . . .                                  | 2        |
| II.1.b Le corps et ses éléments . . . . .                    | 3        |
| II.2. La compilation . . . . .                               | 4        |
| <b>III.Exemples d'utilisation :</b>                          | <b>4</b> |
| III.1.Exemples de code source . . . . .                      | 4        |
| III.2.Exemples d'intégration d'images . . . . .              | 5        |

## I. Découvrir ezTex !

### I.1. Introduction

Le  $\text{\LaTeX}$  est un peu repoussant pour les débutants, mais il est très intéressant de travailler avec car il permet de faire des documents d'une qualité inégalée. C'est pourquoi j'ai créé ce langage de "programmation" qui vous permet de generer des documents  $\text{\LaTeX}$  facilement. Vous devez juste apprendre les bases les plus simples de  $\text{\LaTeX}$ . Le logiciel s'occupe du reste !

### I.2. L'installer ?

**ezTex** est très simple à utiliser. Une commande suffit pour créer vos documents, documents d'une qualité exceptionnel (c'est du  $\text{\LaTeX}$  s'il vous plait) ! Pour ce faire, vous devrez installer **python** et  $\text{\LaTeX}$  bien sur !

Voici comment faire sur Linux-GNU :

#### Code Source bash - 1: Installation de ezTex

```
1 >: sudo apt-get install python -y
2 >: sudo apt-get install texlive-full
```

Ces deux commandes vous permettent d'utiliser **ezTex**.

## II. Comment utiliser ezTex ?

### II.1. La création et compilation du document ezTex

Utiliser **ezTex** vous oblige a savoir comment créer et compiler votre document. Dans les prochaines versions une interface graphique (GUI) sera disponible. Pour l'instant, vous allez devoir créer votre document, il suffit de créer un fichier avec un éditeur de text comme **atom**, **geany**, ou encore **nano** et **emacs** (pour les plus motivés) avec l'extension **.eztex** La structure d'un document **ezTex** est très simple :

#### Code Source bash - 2: Structure de ezTex

```
1 +- Une "preface"
2 +- Un corps
3     +- Un element de section (CHAPTER)
4         +- Du CODE
5             +- Du texte
6             +- Des IMAGES
7             +- ...
8     +- Une fin a l'element de section
```

#### II.1.a La preface

La préface c'est un peu le cerveau de **ezTex**, c'est là où vous allez donner toutes les instructions de compilation. Cette partie permet aussi de définir la titre, sous-titre, date et auteur. Vous disposez de plusieurs éléments pour la personnaliser :

- **TITLE** : vous permet de définir le titre
- **SUB\_TITLE** : vous permet de définir le sous-titre
- **AUTHOR** : vous permet de définir l'auteur
- **TABLE\_OF\_CONTENT** : active (si **TRUE**) ou désactive la table des matières
- **SAVE** : garde le fichier compilé dans le dossier **render**
- **BACK\_UP** : Fait une copie à chaque compilations (si l'ancienne compilation est plus vieille d'une minute).
- **PREVIEW** : active la prévisualisation

## II.1.b Le corps et ses éléments

Une fois la préface configurée, vous allez pouvoir créer votre document. Sachez pour commencer que **toutes** les commandes  $\text{\LaTeX}$  sont disponibles. Ainsi, si **ezTeX** ne propose pas une fonctionnalité, vous pouvez faire ce que vous voulez en utilisant  $\text{\LaTeX}$ . Les commandes que **ezTeX** propose sont :

### Code Source bash - 3: L'environnement d'organisation

```
1 CHAPTER :: TITRE
2         (text, algo, IMAGEs ...)
3         SUB_SECTION :: TITRE
4         SUB_SUB_SECTION :: TITRE
5 END
```

Vous pouvez organiser vos documents en chapitre, sous-section et sous-sous-section. Les formats par défaut de  $\text{\LaTeX}$  sont aussi supportés, mais une belle intégration dans le document **ezTeX** n'est pas garantie.

### Code Source bash - 4: L'environnement code

```
1 CODE :: LANGAGE :: TITRE
2     Du CODE ici, indentation
3         comprise
4 END
```

Les langages supportés sont : **java**, **c++**, **tex** & **bash**. Plus à venir dans le futur !

### Code Source bash - 5: L'environnement image

```
1 IMAGE :: LIEN IMAGE :: ARG TAILLE :: TITRE
```

Cet environnement permet d'afficher des images dans votre document. Elles seront centrées. La taille est personnalisable à votre guise. Si vous voulez une autre disposition, vous pouvez utiliser l'environnement que propose  $\text{\LaTeX}$ .

Un environnement "algo" va faire apparition dans les prochaines versions. Il vous permettra de créer des algorithmes en pseudo code d'une façon extrêmement simple !

## II.2. La compilation

Pour l'instant la compilation se fait par terminal. Dans les prochaines versions une interface graphique sera disponible. Utilisez la commande suivante :

### Code Source bash - 6: Commande de compilation

```
1 python genTex.py FICHER\_A\_COMPILER
```

## III. Exemples d'utilisation :

### III.1. Exemples de code source

#### Code Source bash - 7: Hello-world en C++

```
1 CODE :: C++ :: Hello world !
2         int main(){
3             cout << "Hello world!" << endl;
4         }
5 END
```

Se compilera pour donner :

#### Code Source C++ - 1: Hello world !

```
1 int main(){
2     cout << "Hello world!" << endl;
3 }
```

Nota Bene : L'indentation n'est pas supportée par **ezTex**. Vous devez indenter vous-même. **ezTex** ne détruira pas l'indentation cependant. Le logiciel la comprendra et la transcrira.

### III.2. Exemples d'intégration d'images

#### Code Source bash - 8: Intégration d'images

```
1 IMAGE :: meme.jpg :: scale = 0.4 :: COMMENT ?
```

Nota Bene : Les images doivent etre dans le même dossier que le fichier **ezTex**.  
Se compilera pour donner :

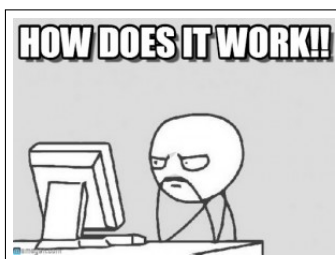


FIGURE 1 – COMMENT ?