le cnam
eicnam

Projet d'initiation à la recherche

Template Typst pour la rédaction de l'étude bibliographique

Auteur
Mathieu Aucejo

FISA Mécanique – 3ème année

Année universitaire 2024 – 2025

Table des matières

Résumé

Ce package Typst est un modèle pour la rédaction du rapport de synthèse bibliographique dans le cadre du module « Projet d'initiation à la recherche » de 3ème année de la FISA Mécanique du Conservatoire national des arts et métiers.

1. Qu'est-ce que Typst ?

Typst est un nouveau langage de balise open source écrit en Rust et développé à partir de 2019 par deux étudiants allemands, Laurenz Mädje et Martin Haug, dans le cadre de leur projet de master [1], [2]. La version 0.1.0 a été publiée sur GitHub le 04 avril 2023.

Typst se présente comme un successeur de LaTeX plus moderne, rapide et simple d'utilisation. Parmi ses avantages, on peut citer :

- la compilation incrémentale ;
- des messages d'erreur clair et compréhensible ;
- un langage de programmation Turing-complet ;
- une système de style cohérent permettant de configurer aisément tous les aspects de son document (police, pagination, marges, …) ;
- une communauté active et sympathique (serveur Discord pour le support, annonce de nouveaux paquets) ;
- un système de paquets simple d'utilisation (pour rechercher ou voir la liste des paquets, n'hésitez pas à visiter Typst: Universe) ;
- des extensions pour VS Code existent, comme `Typst LSP` ou `Tinymist`, pour avoir des fonctionnalités similaires à `LaTeX Workshop`.

Pour finir, la documentation de Typst est suffisamment bien écrite et détaillée pour permettre de créer rapidement ses propres documents. Il faut compter moins d'une heure pour prendre en main la syntaxe (sans mentir 😁). Pour accéder à la documentation, suivez ce lien. Pour faciliter la transition de LaTeX vers Typst, un guide est disponible ici.

2. Utilisation

Pour utiliser le modèle, il faut l'importer dans votre fichier principal `typ`. En supposant que le template et le fichier principal sont dans le même dossier, il suffit d'insérer la commande suivante à la première ligne du fichier principal.

```
#import "report.typ": *
```

> Si vous décomposez votre document en différents fichiers, il faut insérer la commande précédente en préambule de chaque fichier.

2.1. Initialisation du modèle

Après avoir importé le modèle, celui doit être initialisé en appliquant une règle d'affichage (`show` rule) avec la commande `#report-template()` en passant les options nécessaires avec l'instruction `with` dans votre fichier principal `typ` :

```
#show report-template.with(
  ...
)
```

Le modèle `#report-template()` possède un certain nombre de paramètres permettant de personnaliser le document. Voici la liste des paramètres disponibles :

```
#report-template(
  (title): [Titre du rapport],
  (course): [Nom du module],
  (authors): "",
  (supervisors): "",
  (academic-program): none,
  (academic-year): none,
  (cover-image): none,
  (lang): "fr"
)[(body)]
```

Argument

`(title): [Titre du rapport]` — `str` | `content`

Titre du document

Argument

`(course): [Nom du module]` — `str` | `content`

Intitulé du module d'enseignement

Argument

`(authors): ""` — `array`

Liste des auteurs

exemple : `("Auteur A", "Auteur B")`

Argument

`(supervisors): ""` — `none` | `array`

Liste des encadrants du module

exemple : `("Encadrant A", "Encadrant B")`

Argument

`⟨academic-program⟩: none` `none` | `str` | `content`

Nom de la formation

Argument

`⟨academic-year⟩: none` `none` | `str` | `content`

Année universitaire en cours

exemple : `"2024 - 2025"`

Argument

`⟨cover image⟩: none` `none` | `content`

Image d'illustration du document

exemple : `image("image.png")`

2.2. Rédaction du document

Le contenu du document est à rédiger dans le fichier principal `typ` ou dans des fichiers annexes. Pour apprendre les bases de la rédaction d'un document en Typst, vous pouvez consulter ce tutoriel, qui vous permettra de vous familiariser avec les principales fonctionnalités en moins d'une heure.

2.3. Gestion de la bibliographie

Pour insérer une bibliographie, il faut insérer dans le fichier principal `typ` la commande suivante :

```
#bibliography("fichier-biblio.yml/bib")
```

> Le modèle propose deux formats de bibliographie : YAML et BibTeX.
>
> Pour le fichier YAML, celui-ci est interprété par le module `hayagriva`, dont la documentation est disponible ici.

Pour citer une référence bibliographique dans le texte, il suffit d'utiliser la commande `#cite(⟨key⟩)` ou plus simplement `@key` (où `key` est la clé associée à la référence).

Pour plus d'informations sur la gestion des références bibliographiques, il faut se référer à la documentation de la fonction `#bibliography()` de Typst (accessible ici).

2.4. Gestion des annexes

Le modèle propose un environnement `appendix` pour différencier le corps du document des informations annexes. Dans cet environnement, la numérotation des titres de section, des figures, tableaux et équations est adapté au contexte. Pour activer cet environnement, il faut insérer dans le fichier principal `typ` à l'endroit souhaité la commande suivante :

```
#show: appendix
```

2.5. Fonctionnalités additionnelles

Cette section présente les fonctions complémentaires implémentées dans le modèle pour faciliter la rédaction du document.

Figures

D'une manière générale, les figures sont insérées dans le document à l'aide de la fonction `#figure()` de Typst. Cependant, Typst ne dispose pas actuellement de mécanismes permettant de gérer les sous-figures (numérotation et référencement). Pour pallier ce manque, le modèle définit une fonction `#subfigure()` permettant de gérer les sous-figures de manière adaptée. Cette fonction encapsule la fonction `#subpar.grid()` du package `subpar`.

```
#subfigure(
  figure(image("image1.png"), caption: []),
  figure(image("image2.png"), caption: []), <b>,
  columns: (1fr, 1fr),
  caption: [Titre de la figure],
  label: <fig:subfig>,
)
```

L'exemple précédent illustre le cas d'une figure composée de deux sous-figures. La première sous-figure est accompagnée d'une légende, tandis que la seconde possède un `label` mais pas de titre. La seconde sous-figure peut ainsi être référencée dans le texte à l'aide de la commande `@b`.

Équations

Pour encadrer une équation importante, la fonction `#boxeq()` doit être utilisée.

```
$
#boxeq[$p(A|B) prop p(B|A) space p(A)$]
$
```

Pour créer une équation sans numérotation, il faut utiliser la fonction `#nonumeq()`.

```
#nonumeq[$ integral_0^1 f(x) dif x = F(1) - F(0) $]
```

3. Paquets recommandés

Cette section présente une liste de paquets qui peuvent être pertinents lors de la rédaction d'un document en Typst.

Dessin – `CeTZ`
- **Description** : Ce paquet se veut être un équivalent Typst de TiKZ pour LaTeX.
- **Liens** - Typst: Universe, dépôt GitHub et documentation.

Boîtes – `showybox`
- **Description** : Ce paquet permet de créer des boîtes de contenus (textes, …) personnalisables.
- **Liens** - Typst: Universe, dépôt GitHub et documentation.

Code – `codelst`
- **Description** : Ce paquet permet de formatter des blocs de code source en incluant notamment la numérotation des lignes.
- **Liens** Typst: Universe, dépôt GitHub et documentation.

Algorithme – `lovelace`
- **Description** : Ce paquet permet d'écrire du pseudo-code, dont le formattage est personnalisable.
- **Liens** - Typst: Universe, dépôt GitHub et documentation (voir ReadMe sur GitHub).

Mathématiques - `physica`
- **Description** : Ce paquet propose des raccourcis pour l'écriture de symboles mathématiques.
- **Liens** - Typst: Universe, dépôt GitHub et documentation.

Glossaire - `glossarium`
- **Description** : Ce paquet permet de créer un glossaire.
- **Liens** - Typst: Universe, dépôt GitHub et documentation.

Index - `in-dexter`

- **Description** : Ce paquet permet de créer aisément un index.
- **Liens** : Typst: Universe, dépôt GitHub et documentation.

Présentation - `polylux`

- **Description** : Ce paquet permet de créer des présentations de type `PowerPoint` ou `Beamer`.
- **Liens** : Typst: Universe, dépôt GitHub et documentation.

Bibliographie

[1] L. Mädje, « A Programmable Markup Language for Typesetting », M.S. Thesis, Technische Universität Berlin, 2022.

[2] M. Haug, « Fast Typesetting with Incremental Compilation », M.S. Thesis, Technische Universität Berlin, 2022.