

Introduction à L^AT_EX

Olivier MOREL

15 juillet 2022

Table des matières

1 Config VS code

Activer l'option auto clean sur on built

2 Introduction

L^AT_EX est rempli de commandes parfois obscures et il est souvent compliqué de s'en souvenir. Ce document cherche à faciliter la création de documents. Il se base sur les configurations des templates qui se trouvent dans le repo <https://github.com/morel-olivier/latex-templates>.

3 Liens utiles

<https://fr.overleaf.com/learn>

Très bon site d'aide. Toutefois, il ne va pas toujours dans les détails très techniques.

<https://github.com/Wookai/paper-tips-and-tricks>

Excellent document sur différentes combines pour se faciliter la vie et améliorer la qualité des documents.

<https://garsia.math.yorku.ca/~zabrocki/latexpanel/mathaccents.html>

Comment utiliser des accents dans l'environnement math.

<https://www.mathcha.io/>

Editeur de dessin, formule et autre qui permet d'exporter du L^AT_EX.

4 Siunitx

4.1 Introduction

Le package *siunitx* aide le rédacteur à écrire des valeurs numériques et des unités simplement et de manière consistante. Il offre des possibilités de configurations très complexes. La documentation complète se trouve sur <https://ctan.org/pkg/siunitx>.

4.2 Configuration

La configuration proposée ici écrit les nombres en notation ingénieur avec 4 chiffres de précision et des exposants au lieu des puissances de dix si il y a une unité.

```
\sisetup{
  locale = FR,
  round-mode = figures,
  % 4 digits in total
  round-precision = 4,
  % exponnant by multiple of 3
  scientific-notation = engineering,
  per-mode = symbol,
  per-symbol = /,
  exponent-to-prefix = true,
  % decimal marker will always be printed as a '.'
  output-decimal-marker = {.},
  % imaginary unit will always be 'j'
  output-complex-root = j,
  % exponant product will be a dot
  exponent-product = \cdot,
  % keep same police as all the text
  detect-all
}
```

Listing 1: Configuration de siunitx

4.3 Usage basique

Les exemples proposés ici sont spécifiques à la configuration de siunitx proposée ci-dessus (Listing ??).

action	commande	résultat
écrire un nombre	<code>\num{3.14}</code>	3.140
notation simple	<code>\num[scientific-notation=fixed]{0.1}</code>	0.1000
écrire une unité	<code>\si{\m}</code>	m
	<code>\si{\km}</code>	km
	<code>\si{\kg}</code>	kg
	<code>\si{\per\s}</code>	1/s
	<code>\si{\m\per\kg}</code>	m/kg
écrire un résultat	<code>\SI{1.234e14}{\mol}</code>	123.4 Tmol
	<code>\SI{1.23456}{\mol}</code>	1.235 mol
	<code>\SI{123456}{\mol}</code>	123.5 kmol
écrire un angle	<code>\ang{13.346}</code>	13.35°
écrire une valeur ±	<code>\SI{+-3.145}{\volt}</code>	±3.145 V

5 Inclure du code python dans L^AT_EX

5.1 Introduction

Inclure du python dans un document permet de faire les calculs directement dans le document. Cela permet de ne pas avoir besoin de copier les nouveaux résultats des calculs à chaque correction des calculs. De plus, cela permet de générer directement des graphiques dans L^AT_EX. Le package utilisé dans ce cas est [pythontex](#). Il est démontré comment l'installer et l'utiliser sur VS Code avec l'extension latex workshop

5.2 Installation

5.3 Windows 10

5.4 Linux

L'installation a été testée sur Ubuntu 21.10.

Premièrement, il faut vérifier si python est bien installé. Pour ce faire, il faut entrer qu'une de ces commandes donne un résultat.

```
$ python2 -V
Python 2.7.18
$ python3 -V
Python 3.9.7
```

Si ce n'est pas le cas, installez python avec la commande :

```
$ sudo apt install python3
```

Il semblerait que certaine librairie puisse ne pas être compatible avec python3 dans ce cas, vous pouvez installer python2.

Pour l'Utilisation avec VS Code, il faut préciser quel python doit être appelé par défaut. Vérifiez ensuite que vous avez un python qui peut être appelé par défaut :

```
$ sudo update-alternatives --list python
update-alternatives: error: no alternatives for python
```

Dans ce cas, entrez la commande suivante :

```
$ sudo update-alternatives --install /usr/bin/python python
↪ /usr/bin/python3 1
```

Vous avez maintenant une version de python par défaut que vous pouvez vérifier avec :

```
$ python -V
Python 3.9.7
```

5.5 Configuration de VS Code

Afin de pouvoir utiliser pythontex, il faut encore créer une règle pour que VSCode compile correctement. Pour ce faire il faut modifier le fichier settings.json qui contient tous les réglages de VS code. Pour l'ouvrir, utiliser le raccourci CTRL + MAJ + p. Ensuite taper : "Settings : Open Settings (JSON)". Sous latex-workshop.latex.tools ajouter le code du Listing ??.

```
{
    "name": "pythontex",
    "command": "pythontex",
    "args": [
        "%DOC%"
    ],
    "env": {}
},
```

Listing 2: Configuration de pythonrex

Vérifiez ensuite qu'il existe bien une configuration pour pdf_latex. Sinon ajoutez le code du Listing ??.

```
{
    "name": "pdflatex",
    "command": "pdflatex",
    "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "--shell-escape",
        "-file-line-error",
        "%DOC%"
    ],
    "env": {}
},
```

Listing 3: Configuration de pdf_latex

Enfin aller sous "latex-workshop.latex.recipes" et ajouter le code du Listing ??.

```
{
    "name": "pdflatex -> pythontex -> pdflatex",
    "tools": [
        "pdflatex",
        "pythontex",
        "pdflatex"
    ]
},
```

Listing 4: Configuration de la recette

5.6 Utilisation