



Typst, un compilateur de texte

Première rédaction de cet article le 19 novembre 2025

Autres trucs

[Accueil](#)

[Seulement les RFC](#)

[Seulement les fiches de lecture](#)

Mon livre

[« Cyberstructure »](#)

Ève

Recherche dans ce blog :

Au [Capitole du Libre](#) de novembre 2025, j'ai suivi l'exposé de Patrick Massot sur [Typst](#), un concurrent possible de [LaTeX](#). Je ne suis pas un expert de [Typst](#), je n'ai pas encore écrit de vrai texte avec donc ce qui suit est un résumé de l'exposé, agrémenté de quelques essais.

Typst est un **compilateur de texte**, un système où vous tapez du texte et du formatage avec un [éditeur](#) quelconque, texte qui sera ensuite traduit dans des formats de distribution comme [PDF](#). Les avantages de ces compilateurs (ou formatteurs), comme noté par Patrick Massot, sont la reproductibilité, la possibilité de choisir librement son éditeur, la flexibilité, et la possibilité d'automatiser la production de documents. Typst est donc une alternative à [LaTeX](#) ou à [Pandoc+Markdown](#). C'est évidemment un [logiciel libre](#), sinon on n'en parlerait pas à [Capitole du Libre](#). Il se veut plus perfectionné que Pandoc+Markdown (par exemple pour des documents complexes avec de la [mathématique](#)) et plus simple à utiliser que LaTeX, avec une syntaxe moins tordue (pas difficile) et des messages d'erreur plus clairs (pas difficile non plus).

Voici un très simple document (tant que ça reste aussi simple, la syntaxe ressemble vaguement à Markdown) :

= J'essaie Typst

Ça marche. Je peux *insister* ou faire des listes :

+ liberté
+ égalité
+ fraternité

== Des maths

$\$E = m c^2\$$

== Du texte plus long

```
#text(red)[  
On peut colorier.  
]
```

J'ai pas envie de tout taper. #lorem(100)

On le compile :

```
% typst compile essai.typ
```

Et on obtient un [PDF](#).

Comme promis, les messages d'erreur sont clairs et précis, grosse différence avec LaTeX :

```
% typst compile essai.typ  
error: unclosed delimiter  
    └── essai.typ:15:9  
    |  
15 | #set text(  
    |           ^  
  
warning: equation was ignored during HTML export  
    └── essai.typ:11:0  
    |  
11 | $e = m c^2$  
    | ^^^^^^
```

Aucun [CSS](#) n'est produit, le texte en rouge n'est donc pas colorié. Bref, la production de HTML est à l'heure actuelle très limitée. Pour cela, Pandoc est une meilleure solution, car il sait lire le Typst, même les formules mathématiques :

```
% pandoc -i essai.typ -o essai.html
```

Typst dispose d'innombrables autres possibilités, lisez la documentation pour en avoir une idée. On peut créer ses propres fonctions (Typst inclut un langage de programmation complet), définir des [gabarits](#), etc. Il existe déjà plein de paquetages rigolos pour Typst sur [le dépôt officiel](#), notamment pour l'écriture d'articles techniques et scientifiques. Voyons un exemple avec le paquetage [atomic](#) :

```
// (Ceci est un commentaire.) https://typst.app/universe/pac1  
#import "@preview/atomic:1.0.0"
```

= Un essai d'un paquetage Typst

atomic permet de dessiner des atomes, avec leurs électrons

```
// Cf. Article Wikipédia « Électrons par niveau d'énergie  
#atomic.atom(94,239, "Pu", (2, 8, 18, 32, 24, 8, 2))
```

(Attention, à la première utilisation, il faut être connecté à l'Internet, le paquetage étant téléchargé.) Compilé, cela donnera [ce joli PDF](#).

De la même façon, vous pouvez faire des supports de présentation, avec [touying](#) qui est donc un concurrent de [Beamer](#) :

```
#import "@preview/touying:0.6.1": *  
#import themes.simple: *  
#show: simple-theme
```

= Mon exposé

Le sujet de l'exposé

== Première diapo

Je mets du texte.

== Deuxième diapo

Je continue.

Et encore.

Ah, et comment ai-je installé Typst ? C'est [bien documenté](#). Il n'y a apparemment pas de [paquetage](#) de Typst existant, pour aucun système, mais Typst fournit des binaires (déjà compilés). Comme je préfère vérifier que les sources compilent, je l'ai compilé moi-même :

```
% cargo install --locked typst-cli
```

Typst est écrit en [Rust](#) et compiler du Rust est souvent une expérience frustrante. Sur un [Ubuntu stable](#) :

```
error: cannot install package `typst-cli 0.14.0`, it requires  
`typst-cli 0.11.1` supports rustc 1.74
```

Eh oui, comme trop souvent avec Rust, il faut la toute, vraiment toute dernière version du compilateur, le langage changeant tout le temps. (J'ai donc fait mes tests

sur une [Debian *unstable*.](#))

Est-ce que je vais utiliser Typst ? Je ne sais pas encore. Le logiciel est déjà très utilisable, en tout cas sur un système d'exploitation très récent. Je vais faire les supports de ma présentation au [prochain SplinterCon](#) en Typst, ce sera une occasion de tester. L'une de mes inquiétudes est la gestion du projet. Typst est à 100 % du [logiciel libre \(licence Apache\)](#) mais son développement est apparemment étroitement contrôlé par une seule entreprise, qui commercialise des extensions privatives à Typst et dont il n'est pas certain qu'ils acceptent d'ouvrir le projet.

Quelques autres ressources :

- Le [tutoriel officiel](#) est très bien.
- Typst est [bien documenté](#).
- Les exemples utilisés dans la présentation de Patrick Massot : `git clone https://codeberg.org/pmassot/typst_cd12025`.

[Version PDF de cette page](#) (mais vous pouvez aussi imprimer depuis votre navigateur, il y a une feuille de style prévue pour cela)

[Source XML de cette page](#) (cette page est distribuée sous les termes de la licence [GFDL](#))

Si vous aimez, vous pouvez payer avec [Bitcoin](#) : adresse 1HtNJ6ZFUC9yu9u2qAwB4tGdGwPQasQGax (ou voyez le [code QR](#)). Pour toute remarque sur ce blog, s'adresser par courrier à Stéphane Bortzmeyer <stephane+blog@bortzmeyer.org> ou sur le [fédivers](#) à bortzmeyer@mastodon.gougere.fr. Je suis [les règles de Crocker](#) donc pas besoin de faire des excès de diplomatie. Ce blog est strictement personnel et les opinions exprimées ici n'engagent donc que moi, et notamment pas mon employeur présent ou mes employeurs passés ou mes éventuels employeurs futurs.