



Typst : une nouvelle façon d'écrire, de publier

publié le 01/03/2026

Descriptif :

Depuis plusieurs décennies, la référence pour la composition de documents techniques et scientifiques est LaTeX. Pourtant, une nouvelle génération d'outils cherche à concilier puissance typographique et simplicité d'usage. Parmi eux, Typst s'impose progressivement comme une alternative moderne, rapide et élégante.

Sommaire :

- Comment utiliser Typst

Depuis plusieurs décennies, la référence pour la composition de documents techniques et scientifiques est LaTeX. Pourtant, une nouvelle génération d'outils cherche à concilier puissance typographique et simplicité d'usage. Parmi eux, Typst s'impose progressivement comme une alternative moderne, rapide et élégante.

Typst est un système de composition de documents qui combine :

- une syntaxe claire,
- un moteur très rapide,
- un rendu professionnel,
- et une approche proche de la programmation.

Pourquoi utiliser Typst en contexte pédagogique ?

Typst présente plusieurs avantages pour l'enseignement :

- Une syntaxe lisible, proche du Markdown
- Une compilation rapide
- Une gestion native des formules mathématiques
- Une excellente qualité typographique
- Une prise en main rapide pour les collègues peu familiers avec LaTeX

Il est particulièrement adapté pour :

- fiches de cours
- TP guidés
- sujets d'examen
- photocopiés structurés

● Comment utiliser Typst

- Pour débiter, c'est la méthode la plus simple. Il suffit de s'inscrire (gratuitement, du moins lors de la rédaction de cet article avec 200 Mo de stockage et jusqu'à 100 fichiers par projet) sur le [site officiel](#) afin d'accéder à une interface permettant de rédiger directement via son navigateur des documents Typst. Il est même possible de collaborer à plusieurs sur le même document !
- En local (via Visual Studio Code) : Visual Studio Code est un éditeur de texte gratuit (développé par Microsoft) adapté à la programmation. Un plugin permettant d'utiliser TinyMist Typst est disponible, ce qui facilite

grandement l'utilisation de ce dernier puisqu'il n'est plus nécessaire d'utiliser l'invite de commande et que la prévisualisation se fait directement depuis Visual Studio Code.

- Il est possible de compiler un document Typst directement depuis son ordinateur. Vous trouvez des instructions (en anglais) sur comment utiliser typst en local sur la [page du projet github](#) ↗

● Écrire avec Typst

○ Structurer un document avec des titres

Typst propose une syntaxe lisible, souvent plus intuitive que celle de LaTeX. Là où LaTeX multiplie les commandes parfois obscures, Typst privilégie la cohérence.

Exemple :

```
= Chapitre 1 : Algorithmique
```

```
== Objectifs
```

```
Comprendre les bases des algorithmes.
```

```
=== Prérequis
```

```
Notions élémentaires de programmation.
```

Chapitre 1 : Algorithmique

Objectifs

Comprendre les bases des algorithmes.

Prérequis

Notions élémentaires de programmation.

○ Créer des listes

Liste non numérotée

- Comprendre une boucle
- Manipuler des variables
- Tester un programme

- Comprendre une boucle
- Manipuler des variables
- Tester un programme

Liste numérotée

```
+ Lire l'énoncé
```

```
+ Proposer un algorithme
```

```
+ Implémenter en Python
```

```
+ Tester avec plusieurs jeux de données
```

1. Lire l'énoncé
2. Proposer un algorithme
3. Implémenter en Python
4. Tester avec plusieurs jeux de données

○ Insérer des mathématiques

Typst permet d'écrire des formules très naturellement.

Formule en ligne

La complexité de la recherche dichotomique est $O(\log n)$.

Formule centrée

```
$  
sum_(k=1)^n k = (n(n+1))/2  
$
```

Soit une suite définie par :

```
$
u_(n+1) = 2u_n + 3
$
```

avec $u_0 = 1$.

La lisibilité est excellente sans nécessiter de nombreuses commandes.

La complexité de la recherche dichotomique est $O(\log n)$.

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

Soit une suite définie par :

$$u_{n+1} = 2u_n + 3$$

avec $u_0 = 1$.

○ Intégrer du code informatique

Pour les enseignants de NSI, SNT ou mathématiques, l'insertion de code est essentielle.

Exemple en Python

```
```python
def recherche_max(tab):
 maximum = tab[0]
 for valeur in tab:
```

```
 if valeur > maximum:
 maximum = valeur
 return maximum
```
```

#v(20pt)

Exemple en HTML

```
```html
<!DOCTYPE html>
<html>
<head>
 <title>Ma première page</title>
</head>
<body>
 <h1>Bonjour !</h1>
</body>
</html>
```
```

#v(20pt)

Exemple en SQL

```
```sql
SELECT nom, prenom
FROM Eleves
WHERE moyenne > 10
ORDER BY moyenne DESC;
```
```

○ Exemple de "petit" document

= Recherche dichotomique

Exemple en Python

```
def recherche_max(tab):
    maximum = tab[0]
    for valeur in tab:
        if valeur > maximum:
            maximum = valeur
    return maximum
```

Exemple en HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Ma première page</title>
</head>
<body>
    <h1>Bonjour !</h1>
</body>
</html>
```

Exemple en SQL

```
SELECT nom, prenom
FROM Eleves
WHERE moyenne > 10
ORDER BY moyenne DESC;
```

== Objectif

Écrire une fonction réalisant une recherche dichotomique.

== Rappel mathématique

$\$ \text{mid} = \text{floor}((\text{"gauche"} + \text{"droite"})/2)$

== Travail demandé

- + Écrire la fonction en Python
- + Tester avec un tableau trié
- + Mesurer le temps d'exécution

```
```python
def dico(tab, x):
 gauche, droite = 0, len(tab)-1
 while gauche <= droite:
 mid = (gauche + droite)//2
 if tab[mid] == x:
 return mid
 elif tab[mid] < x:
 gauche = mid + 1
 else:
 droite = mid - 1
 return None
```
```

Recherche dichotomique

Objectif

Écrire une fonction réalisant une recherche dichotomique.

Rappel mathématique

$$\text{mid} = \left\lfloor \frac{\text{gauche} + \text{droite}}{2} \right\rfloor$$

Travail demandé

1. Écrire la fonction en Python
2. Tester avec un tableau trié
3. Mesurer le temps d'exécution

```
def dico(tab, x):
    gauche, droite = 0, len(tab)-1
    while gauche <= droite:
        mid = (gauche + droite)//2
        if tab[mid] == x:
            return mid
        elif tab[mid] < x:
            gauche = mid + 1
        else:
            droite = mid - 1
    return None
```