

T'as vu mes docs ?

Je les fais en Typst !

Virginie PAGEAUD

BDX I/O, 07/11/2025

Quarkslab

Virginie PAGEAUD (CASAVECCHIA)

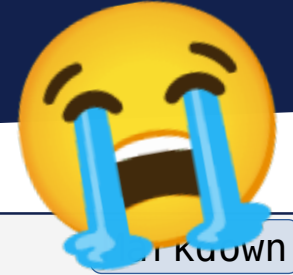


- 🦄 Customer Support Engineer @Quarkslab
- 👩💻 ~15 ans d'XP en dev, test, intégration, support client
- 🔧 Couteau suisse, touche à tout et curieuse
- 🧠 Amatrice d'énigmes et problèmes à résoudre
- 👉 Passionnée par la transmission de connaissances
- 📄 <https://virginie-blog.pageaud.net/>
- 🦋 @la-fee-dragee.bsky.social
- 🌐 Virginie (Casavecchia) Pageaud

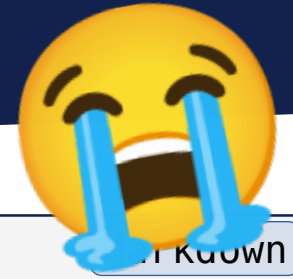


Markdown

```
1 <table>
2 <tr>
3 <td style="text-align:right;"> LinkedIn </td> <td> Virginie (CASAVECCHIA)
  PAGEAUD </td>
4 </tr>
5 <tr>
6 <td style="text-align:right;"> Bluesky </td><td> @la-fee-dragee.bsky.social </
  td>
7 </tr>
8 <tr>
9 <td style="vertical-align: middle; text-align:right;"> Feedback : </td><td>
  <div>  </div> </td>
10 </tr>
```



```
1 <table>
2 <tr>
3 <td style="text-align:right;"> LinkedIn </td> <td> Virginie (CASAVECCHIA)
  PAGEAUD </td>
4 </tr>
5 <tr>
6 <td style="text-align:right;"> Bluesky </td><td> @la-fee-dragee.bsky.social </
  td>
7 </tr>
8 <tr>
9 <td style="vertical-align: middle; text-align:right;"> Feedback : </td><td>
  <div>  </div> </td>
10 </tr>
```



```
1 <table>
2 <tr>
3 <td style="text-align: center;">PAGEAUD </td>
4 </tr>
5 <tr>
6 <td style="text-align: center;">
7 </td>
8 <tr>
9 <td style="vertical-align: top;">
10 <div> 
</div>
</td>
</tr>
</table>
```





The **new foundation** for documents

Limitless power to write, create, and
automate anything that you can fit on a page.

[Try it yourself!](#)

[Sign up](#)



Welcome to Typst's documentation! Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable *and* a pleasure to use.



Welcome to Typst's documentation! Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable *and* a pleasure to use.

typst Public

A markup-based typesetting system that is powerful and easy to learn.

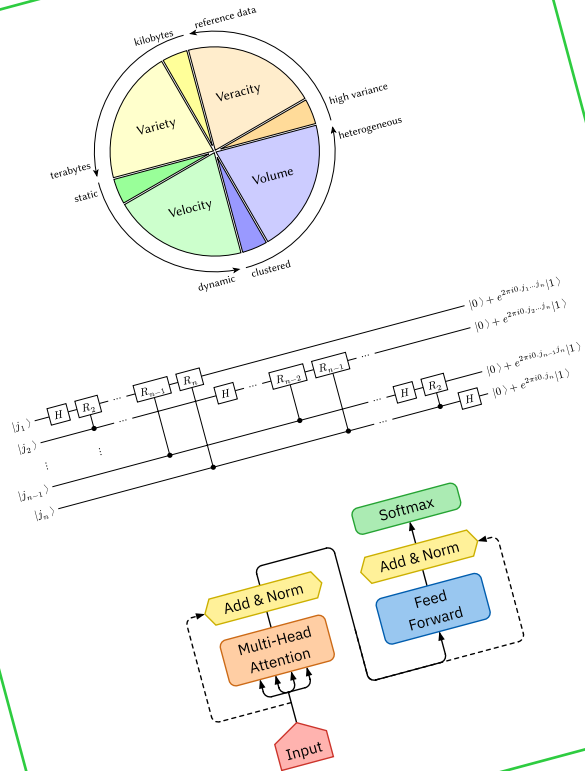


Rust ☆ 47,704 📄 Apache-2.0 👤 1,292 🔍 987 (5 issues need help) 🔗 43 Updated yesterday

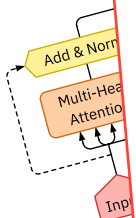
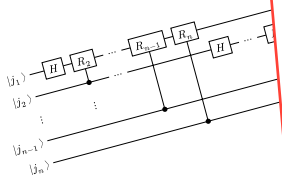
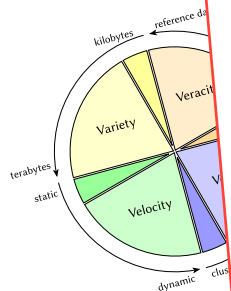
Version 0.14.0 (October 24, 2025) Latest Compare ▾

 laurmaedje released this last week · [63 commits](#) to main since this release  v0.14.0  dd1e6e9

Showroom



Showroom

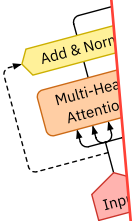
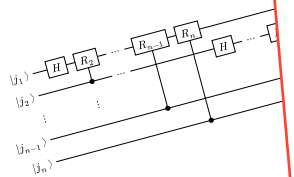
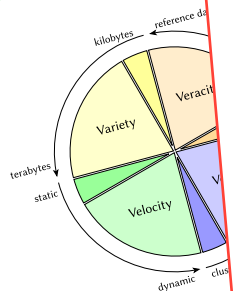


$$\dot{x}_j = \frac{q}{c} \sum_l \dot{x}_l \left\{ \frac{\partial A}{\partial x_j} - \frac{\partial A_j}{\partial x_l} \right\} - \frac{\partial A_j}{\partial t} - c \frac{\partial \varphi}{\partial x_j}$$

$$F_{\mu,\nu} = \begin{pmatrix} 0 & B_z & -B_y & -iE_x \\ -B_z & 0 & B_x & -iE_y \\ B_y & -B_x & 0 & -iE_z \\ iE_x & iE_y & iE_z & 0 \end{pmatrix}$$

$$(ax + b)(cx + d) = acx^2 + (ad + bc)x + bd$$

Showroom

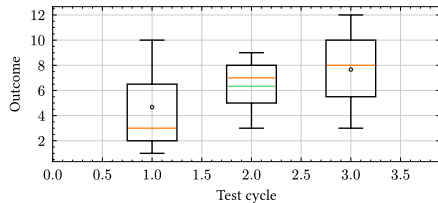
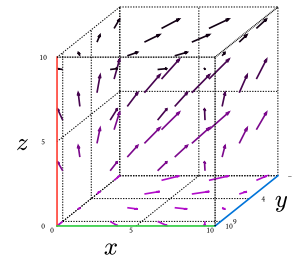
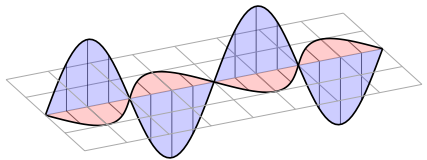


$$\dot{x}_j = \frac{q}{c} \sum_l$$

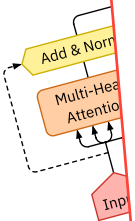
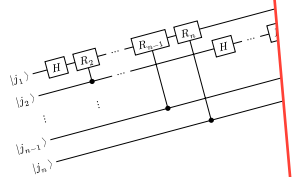
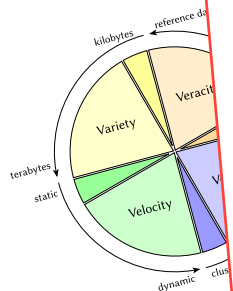
$$-\frac{\partial A}{\partial}$$

$$F_{\mu,\nu} = \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix}$$

$$(ax + b)(cx + d)$$



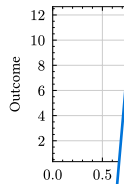
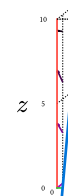
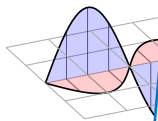
Showroom



$$\dot{x}_j = \frac{q}{c} \sum_l \frac{\partial A}{\partial x_j}$$

$$F_{\mu,\nu} = \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix}$$

$$(ax + b)(cx + d)$$



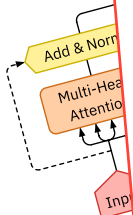
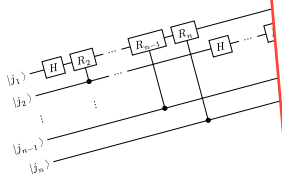
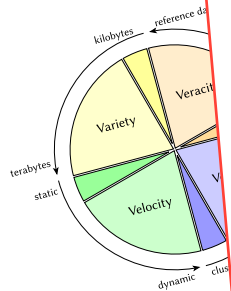
	2023				2024			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Research								
Research the market								
Conduct user surveys								
Development								
Create mock-ups								
Develop application								
QA								
Marketing								
Press demos								
Social media advertising								

Conference demo
Dec 2023

App store launch
Aug 2024

	Value in EUR
Initial Value	
Ending Value	38,110.79
Deposits/Withdrawals	65,727.79
Yield	18,000
... by Asset	9,617
— Stocks	
— Bonds	7,581
— Futures	226
Cumulative Return	1,810
	25.23%

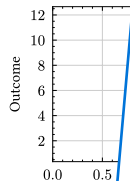
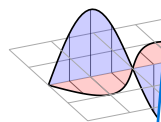
Showroom



$$\dot{x}_j = \frac{q}{c} \sum_l \frac{\partial A}{\partial x_j}$$

$$F_{\mu,\nu} = \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix}$$

$$(ax + b)(cx + d)$$



	2023			
	Q1	Q2	Q3	Q4
Research				
Research the market				
Conduct user surveys				
Development				
Create mock-ups				
Develop application				
QA				
Marketing				
Press demos				
Social media advertising				

Initial Value
Ending Value
Deposits/Withdrawals
Yield
... by Asset
— Stocks
— Bonds
— Futures
Cumulative Return

```
// If the float is already processed, skip it.
let loc = placed.location();
if self.skipped(loc) {
  return Ok(());
}

// If there is already a queued float, queue this one as
// well. We don't want to disrupt the order.
if !self.work.floats.is_empty() {
  self.work.floats.push(placed);
}

// Determine the base size of the chosen scope.
let base = match placed.scope {
  PlacementScope::Column => regions.base(),
  PlacementScope::Parent => self.page_base,
};

// Lay out the placed element.
let frame = placed.layout(self.engine, base)?;

// Determine the remaining space in the scope. This is
// exact for column placement, but only an approximation
// for page placement.
let remaining = match placed.scope {
  PlacementScope::Column => regions.size.y,
  PlacementScope::Parent => {
    .iter()
    .map(|size| size.y)
    .take(
      self.config.columns.count - self.column
    )
  }
}
```

Overview

Tutorial

Reference

LANGUAGE

Syntax

Styling

Scripting

Context

LIBRARY

Foundations

Model

Text

Math

Symbols

Layout

Visualize

Introspection

Data Loading

> Reference

Reference

This reference documentation is a comprehensive guide to all of Typst's syntax, concepts, types, and functions. If you are completely new to Typst, we recommend starting with the [tutorial](#) and then coming back to the reference to learn more about Typst's features as you need them.

Language

The reference starts with a language part that gives an overview over [Typst's syntax](#) and contains information about concepts involved in [styling documents](#), using [Typst's scripting capabilities](#).

Functions

The second part includes chapters on all functions used to insert, style, transform, and layout content in Typst documents. Each function is documented with a description of its purpose, a list of its parameters, and examples of how to use it.

The final part of the reference explains all functions that are used within Typst's code mode to manipulate and transform data. Just as in the previous part, each function is documented with a description of its purpose, a list of its parameters, and examples of how to use it.

< Making a Template
Previous page

Syntax
Next page >

Parameters ⓘ

```
table(
  columns: auto int relative fraction array,
  rows: auto int relative fraction array,
  gutter: auto int relative fraction array,
  column-gutter: auto int relative fraction array,
  row-gutter: auto int relative fraction array,
  fill: none color gradient array tiling function,
  align: auto array alignment function,
  stroke: none length color gradient array stroke tiling dictionary function,
  inset: relative array dictionary function,
  .. content,
) -> content
```

columns `auto` or `int` or `relative` or `fraction` or `array` Settable ⓘ

The column sizes. See the [grid documentation](#) for more information on track sizing.

Default: ()

< Viewing previous page

ON THIS PAGE

Summary

Example

Parameters

columns

rows

gutter

column-gutter

row-gutter

fill

align

stroke

inset

children

Definitions

Table Cell

body

x

y

colspan

rowspan

fill

Typst universe



Universe

[Home](#) [Pricing](#) [Docs](#) [Forum](#) [Sign in](#) [Sign up](#)

[Start](#)

[Packages](#)

[Templates](#)

[Search](#)

[Browse Categories](#)

[Submit](#)

SORT BY

Recommended

KIND

☒ Package

☒ Template

CATEGORY

☒ Office

☒ CV

☒ Presentation

☒ Flyer

☒ Poster

Search 839 packages and templates



aero-check 0.1.1

A simple template to create checklists with an aviation inspired style.



alchemist 0.1.8

A package to render skeletal formulas using CeTZ



bamdome-rebuttal 0.1.1

Rebuttal/response letter template that allows authors to respond to feedback given by reviewers in a peer-review...



basic-resume 0.2.8

A simple, standard resume, designed to work well with ATS.



cetz 0.4.2

Drawing with Typst made easy, providing an API inspired by TikZ and Processing. Includes modules for plotting, charts and tree layout.



charged-ieee 0.1.4

An IEEE-style paper template to publish at conferences and journals for Electrical Engineering, Computer...



classy-german-invoice 0.3.1

Minimalistic invoice for germany based freelancers



clear-iclr 0.7.0

Paper template for submission to International Conference on Learning Representations (ICLR)



codly 1.3.0

Codly is a beautiful code presentation template with many features like smart indentation, line numbering, highlighting, etc.



Touying pour les slides



Touying

Tutorial

Blog

0.6.x ▼

English ▼

GitHub ↗

Search

Introduction to Touying

Getting Started

Sections and Subsections

Code Style

Page Layout

Global Settings

Multi-File Architecture

Dynamic Slides >

Package Integration >

Themes >

Build Your Own Theme

Progress >

Utilities >

Changelog

External Tools >

Introduction to Touying

Version: 0.6.x

Introduction to Touying

Touying is a slide/presentation package developed for Typst. Touying is similar to LaTeX Beamer but benefits from Typst, providing faster rendering speed and a more concise syntax. After, we use "slides" to refer to slideshows, "slide" for a single slide, and "subslide" for a sub-slide.

Why Use Touying

- Unlike **PowerPoint**, Touying is not a "what you see is what you get" tool. You can write your slides in a "content and style separation" manner, especially with Typst, which offers a concise yet powerful syntax, better supporting content like code blocks, mathematical formulas, and theorems. Another advantage is that, with templates, writing slides with Touying is much faster than PowerPoint. Therefore, Touying is more suitable for users with a demand for "research writing."
- Compared to **Markdown Slides**, Touying, relying on Typst, has more powerful typesetting control, such as headers, footers, layout, and convenient custom functions. These are capabilities that Markdown struggles to provide, or does not do well. Additionally, Touying offers `#pause` and `#meanwhile`

Why Use Touying

About the Name

About the Documentation

Contribution

A cup of coffee

Gallery

License



Les joies du code



```
1 #codly()  
2 ```rust  
3 fn main() {  
4     println!("Hello, there!");  
5 }  
6 ```
```

 Typst

Les joies du code



```
1 #codly()  
2 ```rust  
3 fn main() {  
4     println!("Hello, there!");  
5 }  
6 ```
```

 Typst

```
1 fn main() {  
2     println!("Hello, there!");  
3 }
```

 Rust



Les joies du code

```
1 #codly(  
2   highlighted-default-color: fuchsia.lighten(80%),  
3   highlighted-lines: (2,),  
4   highlights: ( (line: 3, start: 24, end: 38, fill: orange), ),  
5 )  
6 ```java  
7 public class HelloWorld {  
8   public static void main(String[] args) {  
9     System.out.println("Hello, world!");  
10  } }  
11 ```
```

 Typst



Les joies du code

```
1 #codly(  
2   highlighted-default-color: fuchsia.lighten(80%),  
3   highlighted-lines: (2,),  
4   highlights: ( (line: 3, start: 24, end: 38, fill: orange), ),  
5 )  
6 ```java  
7 public class HelloWorld {  
8   public static void main(String[] args) {  
9     System.out.println("Hello, world!");  
10  } }  
11 ```
```

 Typst

```
1 public class HelloWorld {  
2   public static void main(String[] args) {  
3     System.out.println("Hello, world!");  
4   } }
```

 Java



Les tables

```
1  #table(t Typst
2    columns: (1fr, 1fr),
3    stroke: (paint: fuchsia, thickness: 3pt, dash: "dotted"),
4    align: center,
5
6    table.header([Prénom], [NOM]),
7
8    [Ada], [#text(purple, [*LOVELACE*])],
9    [Grace], [HOPPER],
10   [Margaret], [HAMILTON],
11  )
```

Les tables



```
1  #table(  
2    columns: (1fr, 1fr),  
3    stroke: (paint: fuchsia, thickness: 3pt, dash: "dotted"),  
4    align: center,  
5  
6    table.header([Prénom], [NOM]),  
7  
8    [Ada], [#text(purple, [*LOVELACE*])],  
9    [Grace], [HOPPER],  
10   [Margaret], [HAMILTON],  
11  )
```

 Typst

Prénom	NOM
Ada	LOVELACE
Grace	HOPPER
Margaret	HAMILTON



On me voit, on me voit plus

```
1 #grid(
2     columns: (auto, 30%),
3     align: (horizon + center, horizon + center),
4
5     [ #only("1")[On me voit] #only("3")[#text(red, weight: "bold", size: 35pt)[On
me re-voit]] ],
6     [ #only("1,3")[#image("../img/mcmahon1.png", width: 70%) ]],
7
8     [#uncover("2")[ On le voit plus ]], [#uncover("2")[ #image("../img/
mcmahon2.png", width: 70%) ]],
9
10    [ ], [#uncover("3")[ #image("../img/mcmahon3.png", width: 70%) ]],
11 )
```

t Typst

On me voit, on me voit plus

```
1 #grid(  
2   columns: (auto, 30%),  
3   align: (horizon + center, horizon + cen  
4  
5   [ #only("1")[On me voit] #only("3")[#te  
     me re-voit]] ],  
6   [ #only("1,3")[#image("../img/mcmahon1.  
7  
8   [#uncover("2")[ On le voit plus ]], [#u  
     mcmahon2.png", width: 70%) ]],  
9  
10  [ ], [#uncover("3")[ #image("../img/mcm  
11 )
```

On me voit



On me voit, on me voit plus

```
1 #grid(  
2   columns: (auto, 30%),  
3   align: (horizon + center, horizon + center),  
4  
5   [ #only("1")[On me voit] #only("3")[#te  
     me re-voit]] ],  
6   [ #only("1,3")[#image("../img/mcmahon1.  
7  
8   [#uncover("2")[ On le voit plus ]], [#u  
     mcmahon2.png", width: 70%) ]],  
9  
10  [ ], [#uncover("3")[ #image("../img/mcm  
11 )
```

t Typst

On le voit plus



On me voit, on me voit plus

```
1 #grid(  
2   columns: (auto, 30%),  
3   align: (horizon + center, horizon  
4  
5   [ #only("1")[On me voit] #only("3  
     me re-voit]] ],  
6   [ #only("1,3")[#image("../img/mcm  
7  
8   [#uncover("2")[ On le voit plus ]  
     mcmahon2.png", width: 70%) ]],  
9  
10  [ ], [#uncover("3")[ #image("../i  
11 )
```



He's back!



Des arborescences de fichiers

```
1 #tree-list[  
2   - rootDir/  
3     - aFile.log  
4     - subDir/  
5       - #text(olive, weight: "bold")  
         [greenFile.txt]  
6     - unlock  
7 ]
```

 Typst



Des arborescences de fichiers

```
1 #tree-list[
2   - rootDir/
3     - aFile.log
4     - subDir/
5       - #text(olive, weight: "bold")
6         [greenFile.txt]
7   - unlock
```

 Typst

```
└─ rootDir/
   │
   ├── aFile.log
   │
   ├── subDir/
   │   └─ greenFile.txt
   │
   └─ unlock
```



Des graphes aussi beaux qu'xkcd

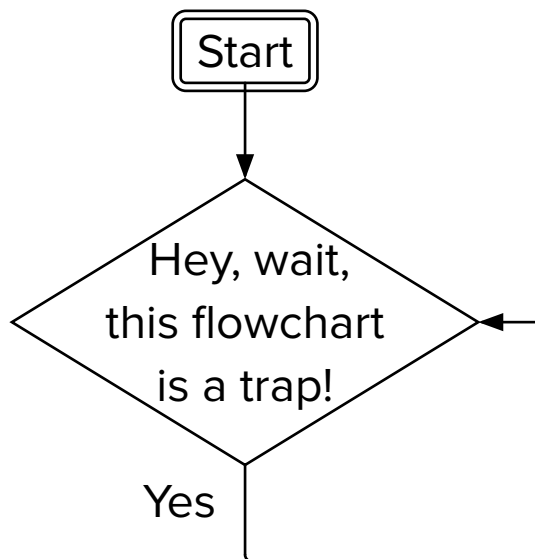
```
1 #fletcher-diagram(  
2   node-stroke: 1pt,  
3   edge-stroke: 1pt,  
4   spacing: 3em,  
5  
6   node((0, 0), [Start], corner-radius: 2pt, extrude:  
7     (0, 3)),  
8   edge("d", "-|>"),  
9   node((0, 1), align(center)[ Hey, wait,\ this  
10     flowchart\ is a trap! ], shape: diamond),  
11   edge("d,r,u,l", "-|>", [Yes], label-pos: 0.1),  
12 )
```

 Typst

Des graphes aussi beaux qu'xkcd

```
1 #fletcher-diagram(  
2   node-stroke: 1pt,  
3   edge-stroke: 1pt,  
4   spacing: 3em,  
5  
6   node((0, 0), [Start], corner-radius: 2pt, extrude:  
7     (0, 3)),  
8   edge("d", "-|>"),  
9   node((0, 1), align(center)[ Hey, wait,\ this  
10     flowchart\ is a trap! ], shape: diamond),  
11   edge("d,r,u,l", "-|>", [Yes], label-pos: 0.1),  
12 )
```

t Typst



Speaker cheatsheet



```
1 #speaker-note[
2 - Ceci est mon antisèche
3 - Penser aux canelés
4 - #lorem(20) (Ca veut rien dire mais ça fait classe.)
5 ]
```

Typst

Speaker cheatsheet



1 #spe
2 - Ce
3 - Pe
4 - #
5]

Speaker cheatsheet

```
1 #speaker-note[
2 - Ceci est mon antisèche
3 - Penser aux canelés
4 - #lorem(20) (Ca veut rien dire mais ça fait classe.)
5 ]
```

Typst

17 / 23

- Ceci est mon antisèche
- Penser aux canelés
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat. (Ca veut rien dire mais ça fait classe.)

Speaker cheatsheet

1 #speake
2 - Ceci
3 - Pense
4 - #lore
5]

Speaker cheatsheet

```
1 #speaker-note[
2 - Ceci est mon antisèche
3 - Penser aux canelés
4 - Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat. (Ca veut rien dire mais ça fait classe.)
5 ]
```

Typst

14:30:34 00m 36s

Next

Ready

Next

17 / 23

Next

Prev

08:21:49

00m 59s

Ready

Next

17 / 23

Génération des output



```
1 typst compile fichier.typ
```

Shell

```
1 touying compile fichier.typ
```

Shell

Génération des output



```
1 typst compile fichier.typ
```

Shell

fichier.pdf

```
1 touying compile fichier.typ
```

Shell

Génération des output



```
1 typst compile fichier.typ
```

 Shell fichier.pdf

```
1 touying compile fichier.typ
```

 Shell fichier.html

Et niveau perfs ?



Compiler 180 CV en pdf ?

LaTeX	Typst

Et niveau perfs ?



Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	

Et niveau perfs ?



Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	30 secondes 

Et niveau perfs ?

Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	30 secondes 

Générer des slides HTML ?

reveal.js	Typst



Et niveau perfs ?

Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	30 secondes 

Générer des slides HTML ?

reveal.js	Typst
2 minutes (voire +) Déploiement serveur	

Et niveau perfs ?

Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	30 secondes 

Générer des slides HTML ?

reveal.js	Typst
2 minutes (voire +) Déploiement serveur	Qq secondes Fichier standalone 

Et niveau perfs ?

Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	30 secondes

Générer des slides HTML ?

Typst
30 secondes standalone 



Les atouts :

-  syntaxe compacte
-  flexibilité
-  rapidité de compilation
-  exports pdf et html légers

Les atouts :

-  syntaxe compacte
-  flexibilité
-  rapidité de compilation
-  exports pdf et html légers

Les limites :

-  pas de support des vidéos
-  pas de défilement dynamique dans les blocs de code

Les atouts :

-  syntaxe compacte
-  flexibilité
-  rapidité de compilation
-  exports pdf et html légers

Les limites :

-  pas de support des vidéos
-  pas de défilement dynamique dans les blocs de code

Références principales

typst : <https://typst.app/>

touying : <https://touying-typ.github.io/>

touying-exporter : <https://github.com/touying-typ/touying-exporter>

Plugin VSCode `Tinymist` `Typst`



On essaye ensemble ?



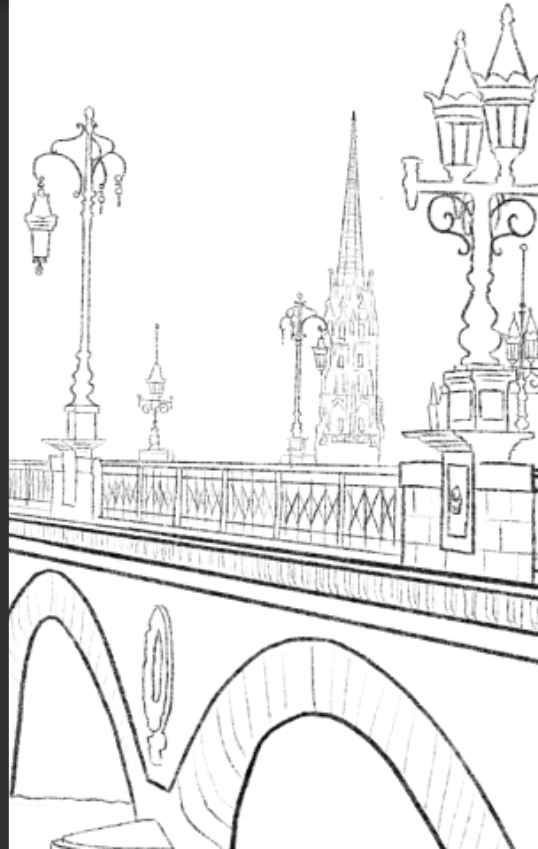
Feedback



T'as vu mes docs ? Je les fait avec typst ! #Frontend

Virginie Pageaud

Drôle/original 😄 1 vote	Très enrichissant 🧐 5 votes
Super intéressant 👍 13 votes	Très bon orateur 🗣️ 7 votes
Pas clair 🤔	Trop technique 📖
Pas assez de démo/exemple 🙄	Trop complexe 🤯



Commentaire

Votre réponse

1 vote

il y a 26 minutes

Super intéressant, j'avoue!

1 vote

il y a 26 minutes

Excellente découverte d'outil.
Pédagogie au top.

1 vote

il y a 26 minutes

Intéressant a connaitre

Thank you

Contact information

LinkedIn:

Virginie (Casavecchia) Pageaud

Bluesky:

@la-fee-dragee.bsky.social

Website:

<https://virginie-blog.pageaud.net/>