

**T'as vu mes docs ?
Je les fais en Typst !**

Virginie PAGEAUD

Volcamp, 02/10/2025

Quarkslab

% whoami

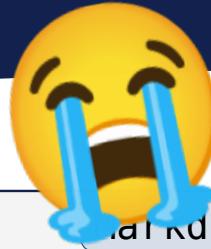


Virginie PAGEAUD (CASAVECCHIA)

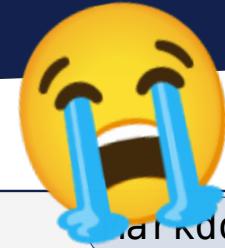
- 🦄 Customer Support Engineer @Quarkslab
- 💻 ~15 ans d'XP en dev, test, intégration, support client
- 🔧 Couteau suisse, touche à tout et curieuse
- 🧠 Amatrice d'énigmes et problèmes à résoudre
- 👉 Passionnée par la transmission de connaissances
- 💻 <https://virginie-blog.pageaud.net/>
- 🦋 @la-fee-dragee.bsky.social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



```
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;"></td>
7 </tr>
8 <tr>
9 <td style="vertical-align: middle; text-align: center;">
<div> </div>
10 </tr>
```





typst

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

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.

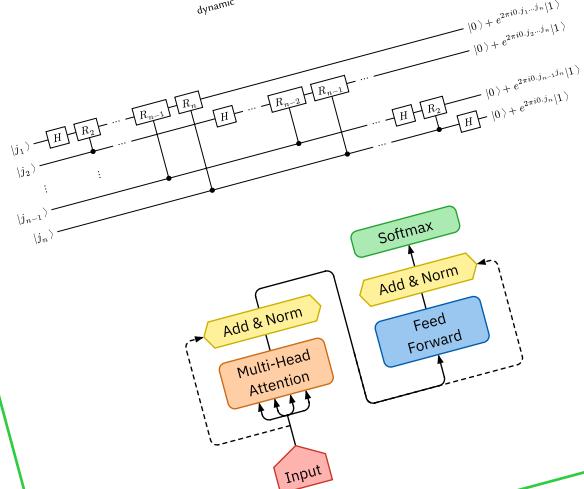
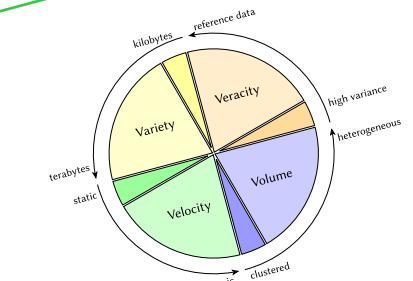
typst Public



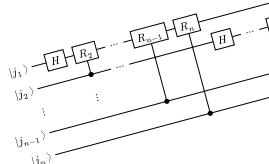
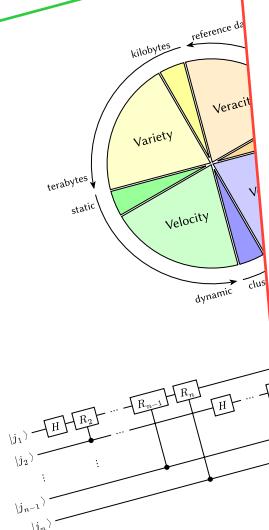
A new markup-based typesetting system that is powerful and easy to learn.

 Rust  45,510  Apache-2.0  1,219  934 (5 issues need help)  45 Updated 6 hours ago

Showroom



Showroom



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

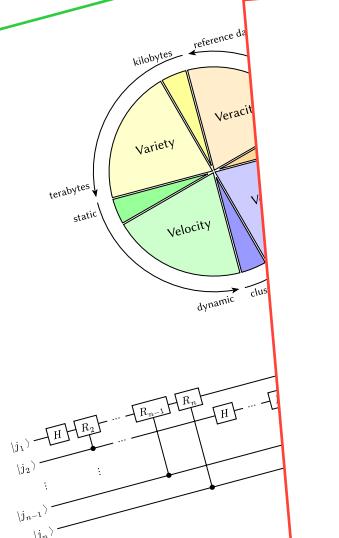
Curved arrows show the flow of terms: $a \rightarrow cx$, $b \rightarrow d$, $c \rightarrow ad$, and $d \rightarrow bc$.

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

$$- \frac{\partial A_j}{\partial t} - c \frac{\partial \varphi}{\partial x_j}$$

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

Showroom

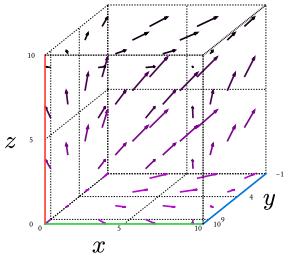
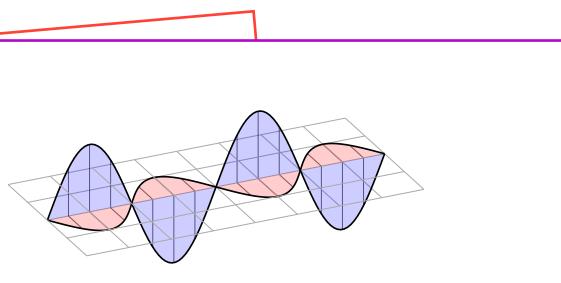
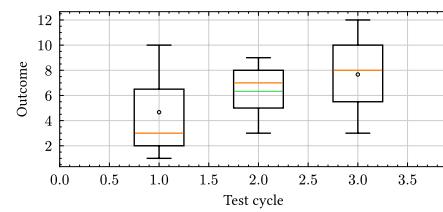


$$\dot{x}_j = \frac{q}{c} \sum_l \frac{\partial f_l}{\partial x_j} - \frac{\partial f_l}{\partial \dot{x}_j}$$

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

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

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



Showroom

The collage consists of four distinct panels:

- Panel 1 (Left):** A circular diagram representing a data storage hierarchy. It is divided into segments labeled "terabytes", "static", "Velocity", "Variety", "Veracity", "dynamic", and "clus". Arrows point from "terabytes" to "static" and "dynamic", and from "static" to "clus".
- Panel 2 (Center Left):** A mathematical equation for a derivative of a function \dot{x}_j with respect to δ . The equation is:

$$\dot{x}_j = \frac{q}{c} \sum_l \frac{\partial f_j}{\partial \delta_l}$$
- Panel 3 (Center Right):** A diagram of a neural network layer. It shows input nodes $|j_1\rangle, |j_2\rangle, \dots, |j_{n-1}\rangle, |j_n\rangle$ connected to hidden nodes H_1, H_2, \dots, H_m , which are then connected to output nodes R_1, R_2, \dots, R_m . A yellow box labeled "Add & Norm" is shown above the hidden layer, and an orange box labeled "Multi-Head Attention" is shown above the output layer. An "Inp" node is at the bottom.
- Panel 4 (Right):** A composite image showing a timeline chart, a 3D surface plot, and a financial statement.
 - Timeline Chart:** A Gantt chart for a project spanning from Q1 2023 to Q4 2024. Tasks include Research, Development, QA, and Marketing activities, with specific milestones like "Conference demo" (Dec 2023) and "App store launch" (Aug 2024).
 - 3D Surface Plot:** A 3D surface plot showing a wavy surface over a grid.
 - Financial Statement:** A table titled "Value in EUR" showing financial metrics:

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

Showroom

The image is a collage of four distinct visual elements:

- Top Left:** A 3D pie chart illustrating data dimensions. The chart is divided into segments labeled "Variety" (yellow), "Velocity" (green), "Veracity" (orange), "dynamic" (blue), and "static" (light green). Arrows point from labels like "kilobytes" and "terabytes" to their respective segments.
- Middle Left:** A mathematical equation for vector addition. It shows two vectors, \vec{x}_j and \vec{y}_j , being added together. The result is expressed as a sum of components along basis vectors e_i :
$$\vec{x}_j = \frac{q}{c} \sum_l e_i - \frac{\partial \vec{x}_j}{\partial t}$$

$$F_{\mu,\nu} = \begin{pmatrix} & \\ & \\ & i \\ & \\ & \end{pmatrix}$$
- Middle Right:** A 3D surface plot showing a wavy, bell-shaped surface over a grid.
- Bottom Right:** A software code snippet written in a language that uses curly braces {}, colons :, and semicolons ;. The code is part of a class or function that handles floating-point operations. It includes comments explaining its logic, such as skipping already processed locations and queuing new ones if they are not yet present.

RTFM !

The screenshot shows a documentation page for Typst. At the top left is a search bar labeled "Search (\$)" with a magnifying glass icon. To its right is a breadcrumb navigation bar with a home icon and the text "> Reference". On the far left is a vertical sidebar with a "Reference" tab highlighted in blue. Below the sidebar are two sections: "LANGUAGE" and "LIBRARY". Under "LANGUAGE", the following items are listed: Syntax, Styling, Scripting, Context. Under "LIBRARY", the following items are listed: Foundations, Model, Text, Math, Symbols, Layout, Visualize, Introspection, Data Loading. The main content area has a title "Reference" with a subtitle "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." Below this is a section titled "Language" with the following text: "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](#)." Another section titled "Functions" follows, with the text: "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 section is "Syntax", which is described as "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." At the bottom of the page are two navigation buttons: "Making a Template < Previous page" and "Syntax > Next page".

Search (\$)

> 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

RTFM !

Search (S)

- Overview
- Tutorial
- Reference ▾
- LANGUAGE
- Syntax
- Styling
- Scripting
- Context
- LIBRARY
- Foundations
- Model
- Text
- Math
- Symbols
- Layout
- Visualize
- Introspection
- Data Loading

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: ()



Main...
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

typst Universe

Start Packages Templates Search Browse Categories Submit

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

bamdone-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.

SORT BY Recommended KIND Package Template CATEGORY Office CV Presentation Flyer Poster



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

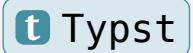
Gallery

License



Les joies du code

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



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 ````
```

t 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 ````
```

t Typst

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

☕ Java

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  )
```

t Typst

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 )
```

t 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  
6     me re-voit]] ],  
7  
8   [ #only("1,3") [#image("../img/mcmahon1.png", width: 70%) ]],  
9  
10  [ #uncover("2") [ On le voit plus ]], [ #uncover("2") [ #image("../img/  
11    mcmahon2.png", width: 70%) ]],  
12  [ ], [ #uncover("3") [ #image("../img/mcmahon3.png", width: 70%) ]],  
13 )
```

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  
6     me re-voit] ] ,  
7  
8   [ #only("1,3") [#image("../img/mcm  
9  
10  [ ], [#uncover("2") [ On le voit plus ]  
11    mcmahon2.png", width: 70%) ]],  
12  
13  [ ] , [#uncover("3") [ #image("../img/mcmahon3.png", width: 70%) ]],  
14  
15  [ ] ] )
```



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 ]
```

t Typst

Des arborescences de fichiers

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

t Typst

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

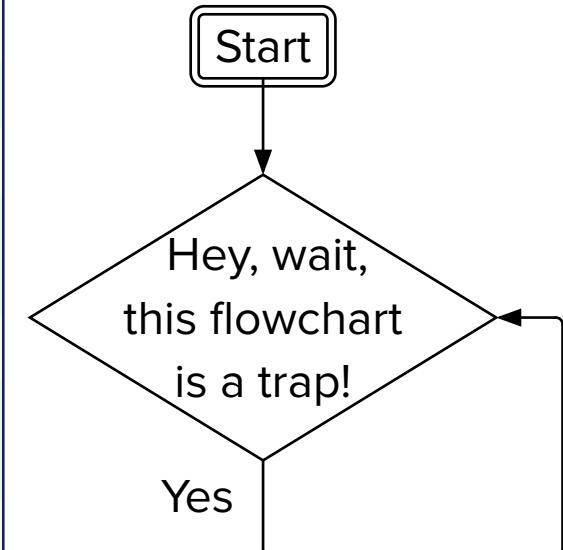
Des graphes aussi beaux qu'xkcd

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

t Typst

Des graphes aussi beaux qu'xkcd

```
1 #fletcher-diagram(                                t Typst
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 )
```



Speaker cheatsheet

```
1 #speaker-note[t Typst  
2 Ceci est mon antisèche \  
3 Penser à demander du rab de truffade \  
4 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ca veut rien dire  
mais ça fait classe.  
5 ]
```

Speaker cheatsheet

Speaker cheatsheet

1 #speaker-note[
2 Ceci est mon antisèche \
3 Penser à demander du rab de truffade \
4 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ca veut rien dire
mais ça fait classe.
5]

17 / 21

Génération des output

1 typst compile fichier.typ

1 touying compile fichier.typ

18 / 21

Ceci est mon antisèche

Penser à demander du rab de truffade

Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Ca veut rien dire mais ça fait
classe.

Next

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 perf's ?

Compiler 180 CV en pdf ?

LaTeX	Typst

Et niveau perf's ?

Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	

Et niveau perf's ?

Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	30 secondes 

Et niveau perf's ?

Compiler 180 CV en pdf ?

LaTeX	Typst
15 minutes	30 secondes 

Générer des slides HTML ?

reveal.js	Typst

Et niveau perf's ?

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 perf's ?

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 perf's ?

Compiler 180 CV en

LaTeX	Type	Typst
15 minutes	30 seconds	30 seconds standalone 



Takeaway

Les atouts :

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

Takeaway

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

Takeaway

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



Thank you

Quarkslab