T’as vu mes docs ?

Je les fais en Typst !

Virginie PAGEAUD

Volcamp, 02/10/2025

Quarkslab

% whoami

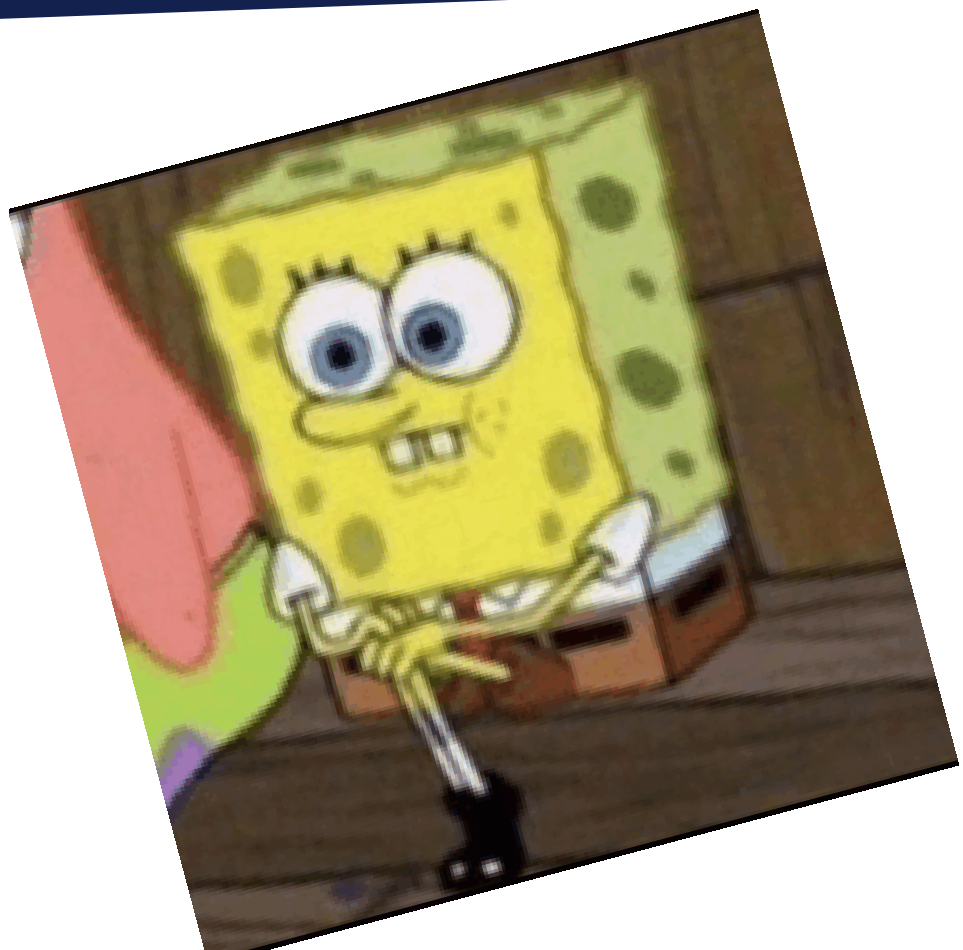
Virginie PAGEAUD (CASAVECCHIA)

- 🦄 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

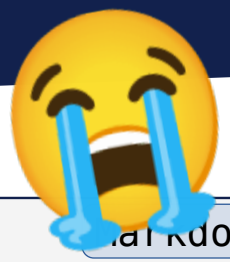
📰 https://virginie-blog.pageaud.net/

@la-fee-dragee.bsky.social

Virginie (Casavecchia) Pageaud

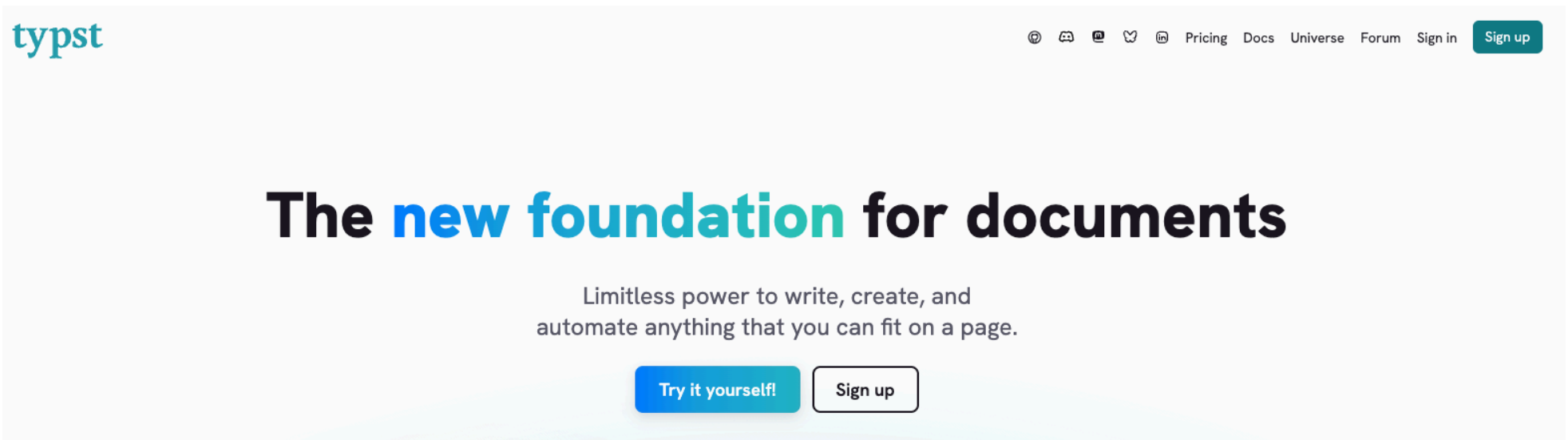


```Markdown
<table>
<tr>
<td style="text-align:right;"> LinkedIn </td> <td> Virginie (CASAVECCHIA) PAGEAUD </td>
</tr>
<tr>
<td style="text-align:right;"> Bluesky </td><td> @la-fee-dragee.bsky.social </td>
</tr>
<tr>
<td style="vertical-align: middle; text-align:right;"> Feedback : </td><td> <div> <img src="images/feedback.png" height=150> </div> </td>
</tr>
```



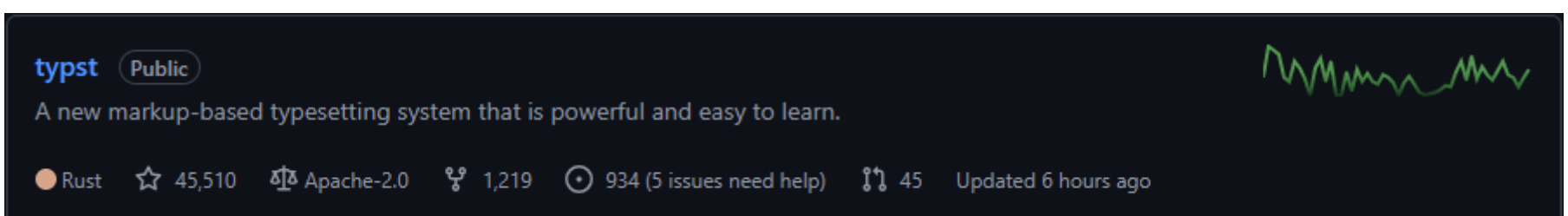
```
1  <table>
2  <tr>
3  <td style="text-align:right;"> LinkedIn </td> <td> Virginie (CASAVECCHIA)
   PAGEAUD </td>
4  </tr>
5  <tr>
6  <td style="text-align:right;"> Bluesky </td><td> @la-fee-dragee.bsky.social </
   td>
7  </tr>
8  <tr>
9  <td style="vertical-align: middle; text-align:right;"> Feedback : </td><td>
   <div> <img src="images/feedback.png" height=150> </div> </td>
10 </tr>
```

```
1  <table>
2  <tr>
3  <td style="tex
   PAGEAUD </td>
4  </tr>
5  <tr>
6  <td style="
   td>
7  </tr>
8  <tr>
9  <td style="vertical-align: mi
   <div> <img src="images/feedback.png"
10 </tr>
```

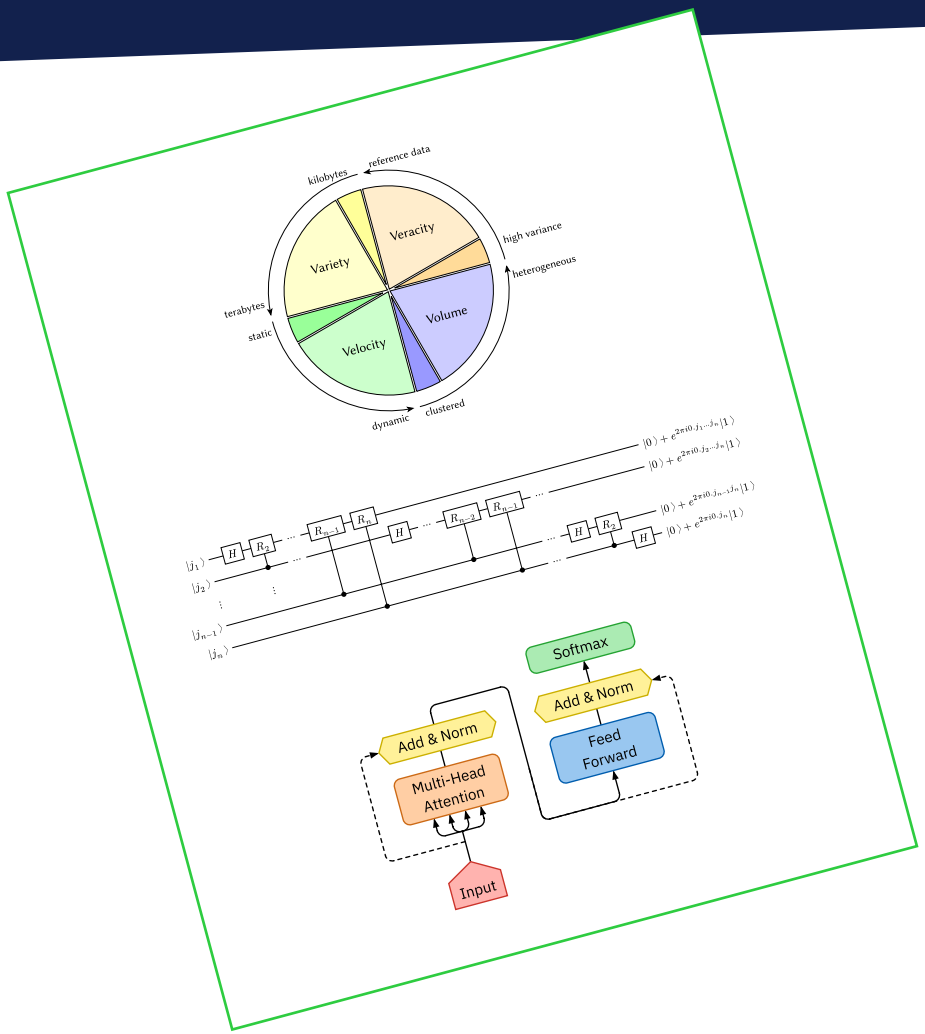

typst
Pricing
Docs
Universe
Forum
Sign in
Sign up
The new foundation for documents
Limitless power to write, create, and
automate anything that you can fit on a page.
Try it yourself!
Sign up

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.

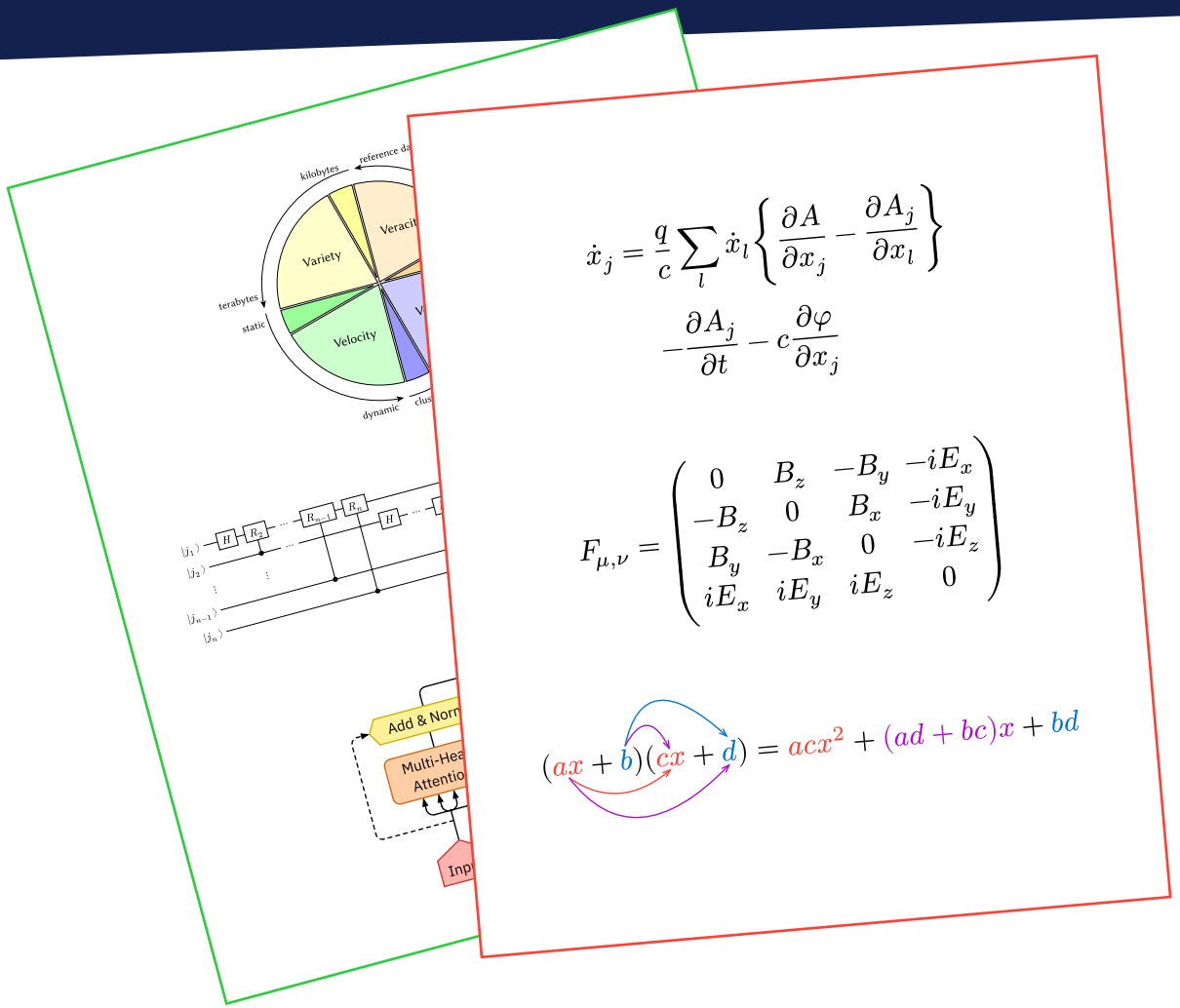
Welcome to Typst's documentation! Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable *and* a pleasure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



Showroom


kilobytes
reference data
Veracity
high variance
heterogeneous
Variety
terabytes
static
Volume
Velocity
dynamic
clustered
Softmax
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Input

Showroom

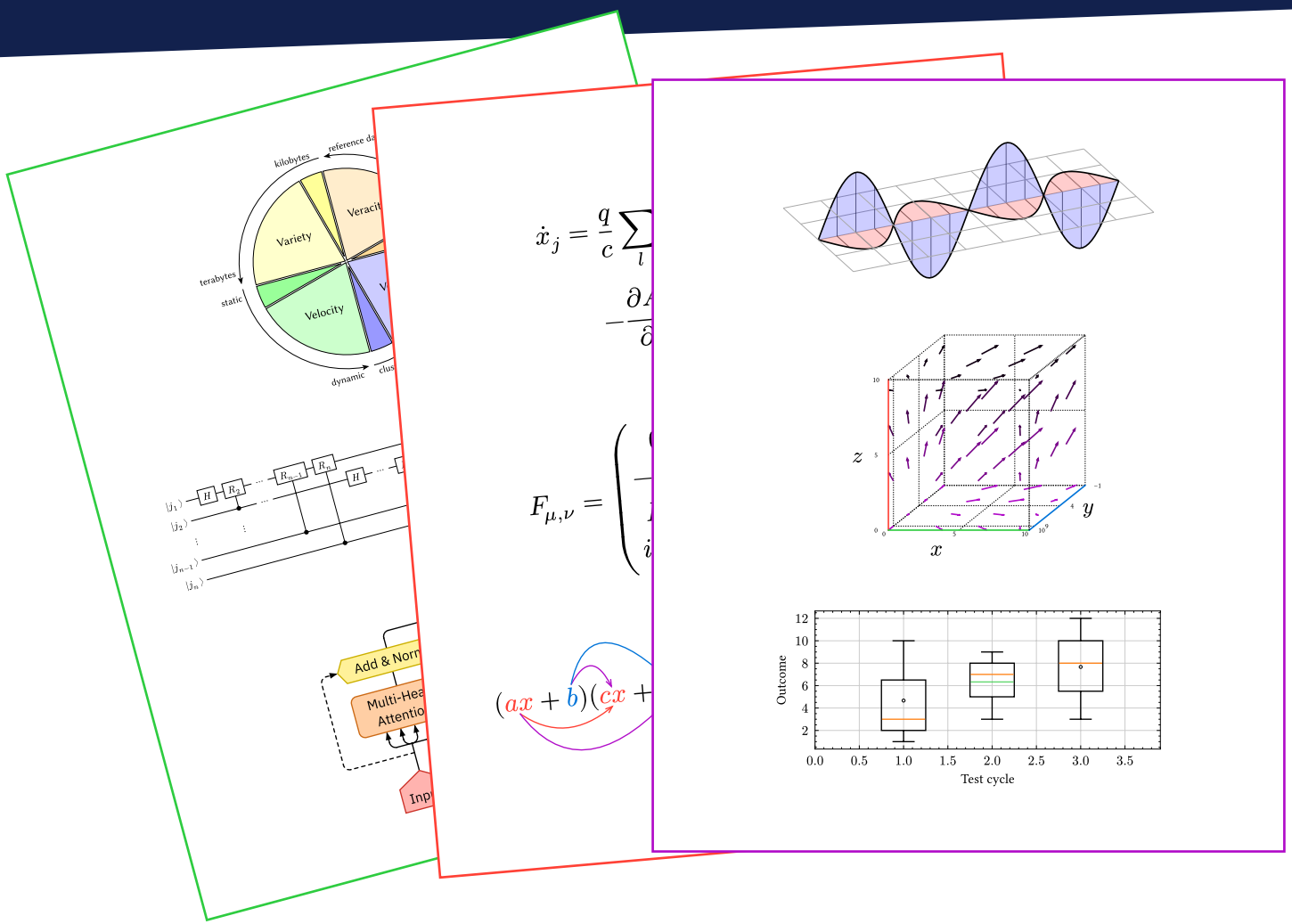


$$\dot{x}_j = \frac{q}{c}\sum_l \dot{x}_l \left\{ \frac{\partial A}{\partial x_j} - \frac{\partial A_j}{\partial x_l} \right\} - \frac{\partial A_j}{\partial t} - c\frac{\partial \varphi}{\partial x_j}$$

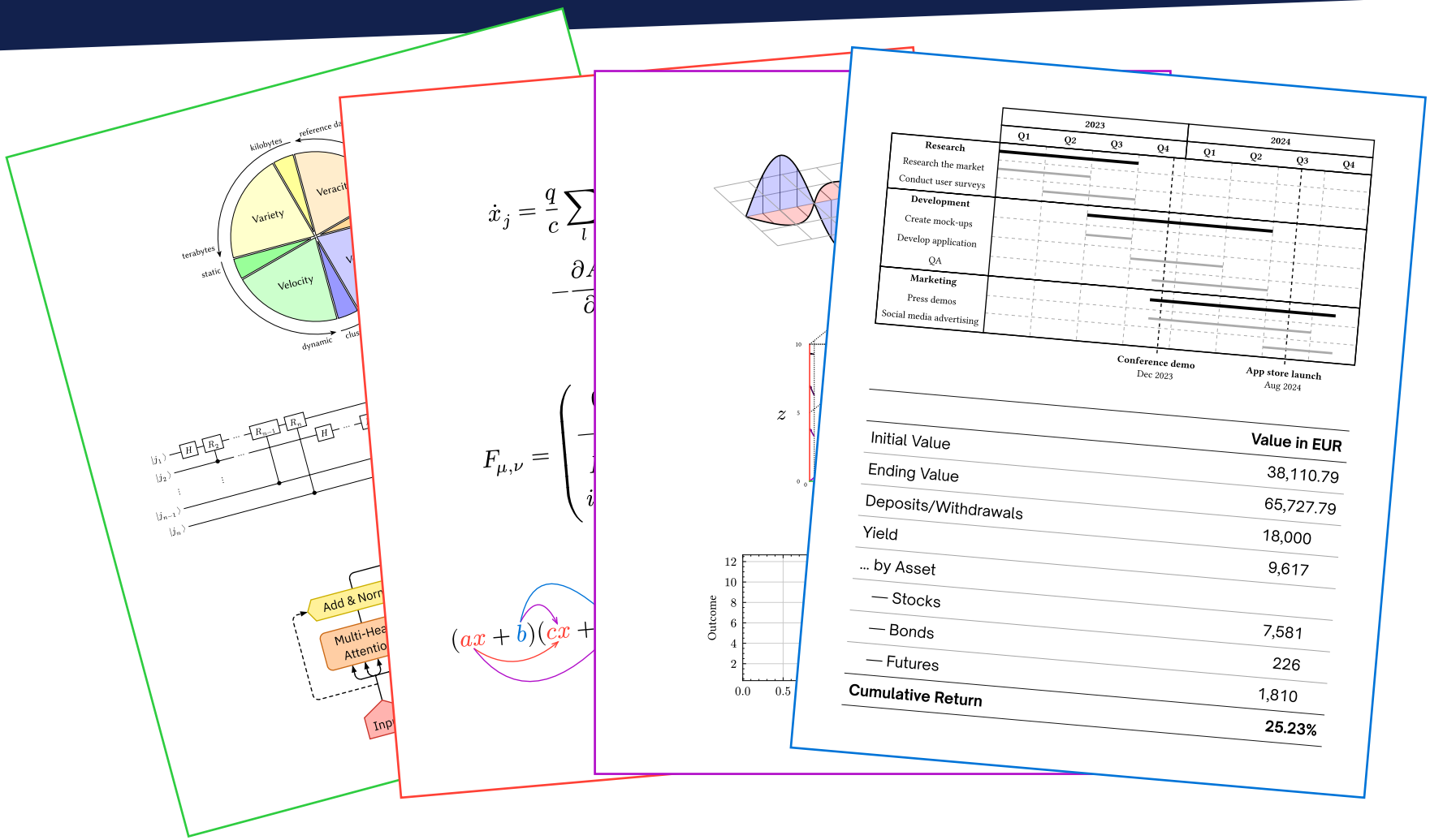
$$F_{\mu,\nu} = \begin{pmatrix} 0 & B_z & -B_y & -iE_x \\ -B_z & 0 & B_x & -iE_y \\ B_y & -B_x & 0 & -iE_z \\ iE_x & iE_y & iE_z & 0 \end{pmatrix}$$

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

Showroom

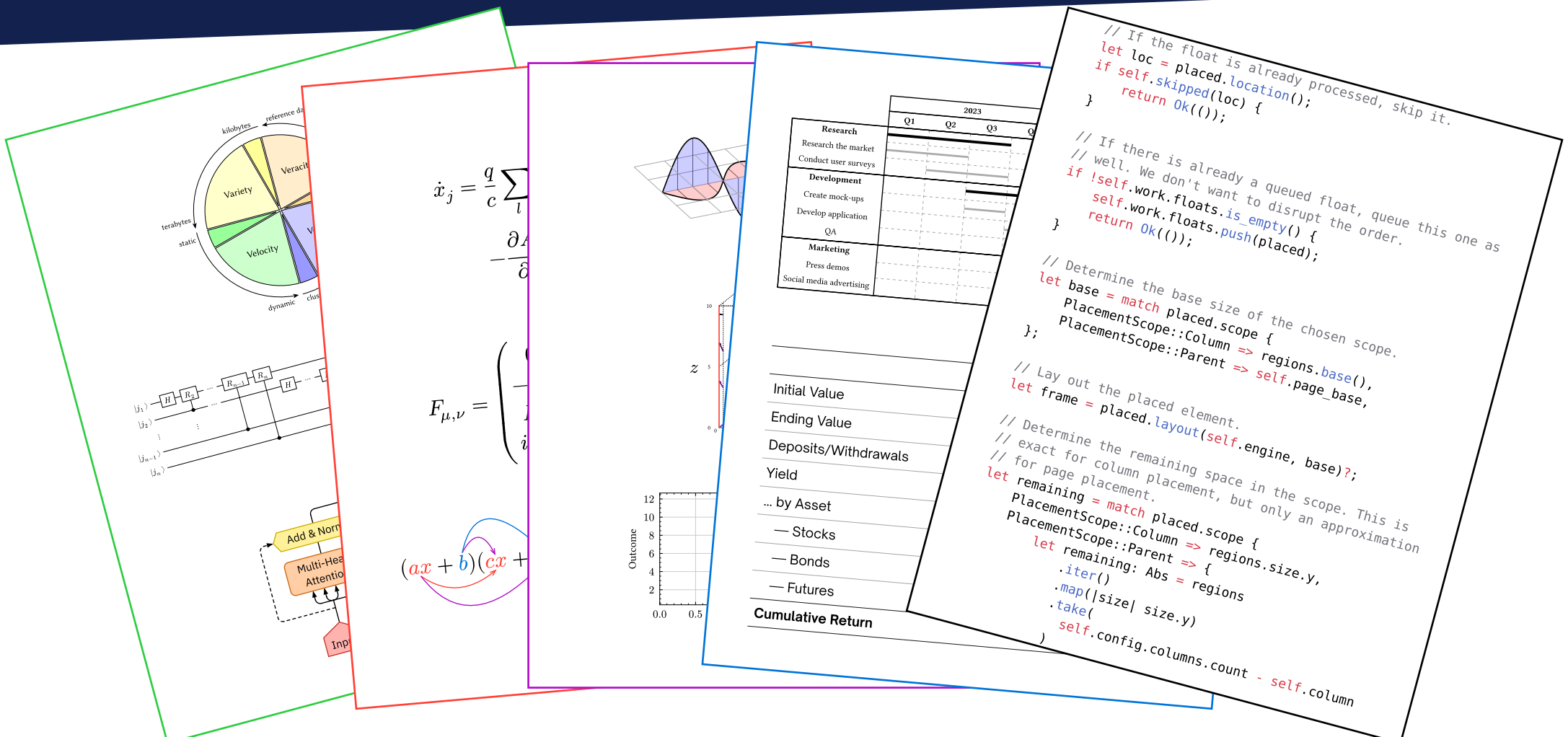

kilobytes
Variety
Velocity
terabytes
static
dynamic
$\dot{x}_j = \frac{q}{c} \sum_l$
$F_{\mu,\nu} =$
$(ax + b)(cx +$
Add & Norm
Multi-Hea
Attentio
Inp
Outcome
Test cycle
z
y
x

Showroom

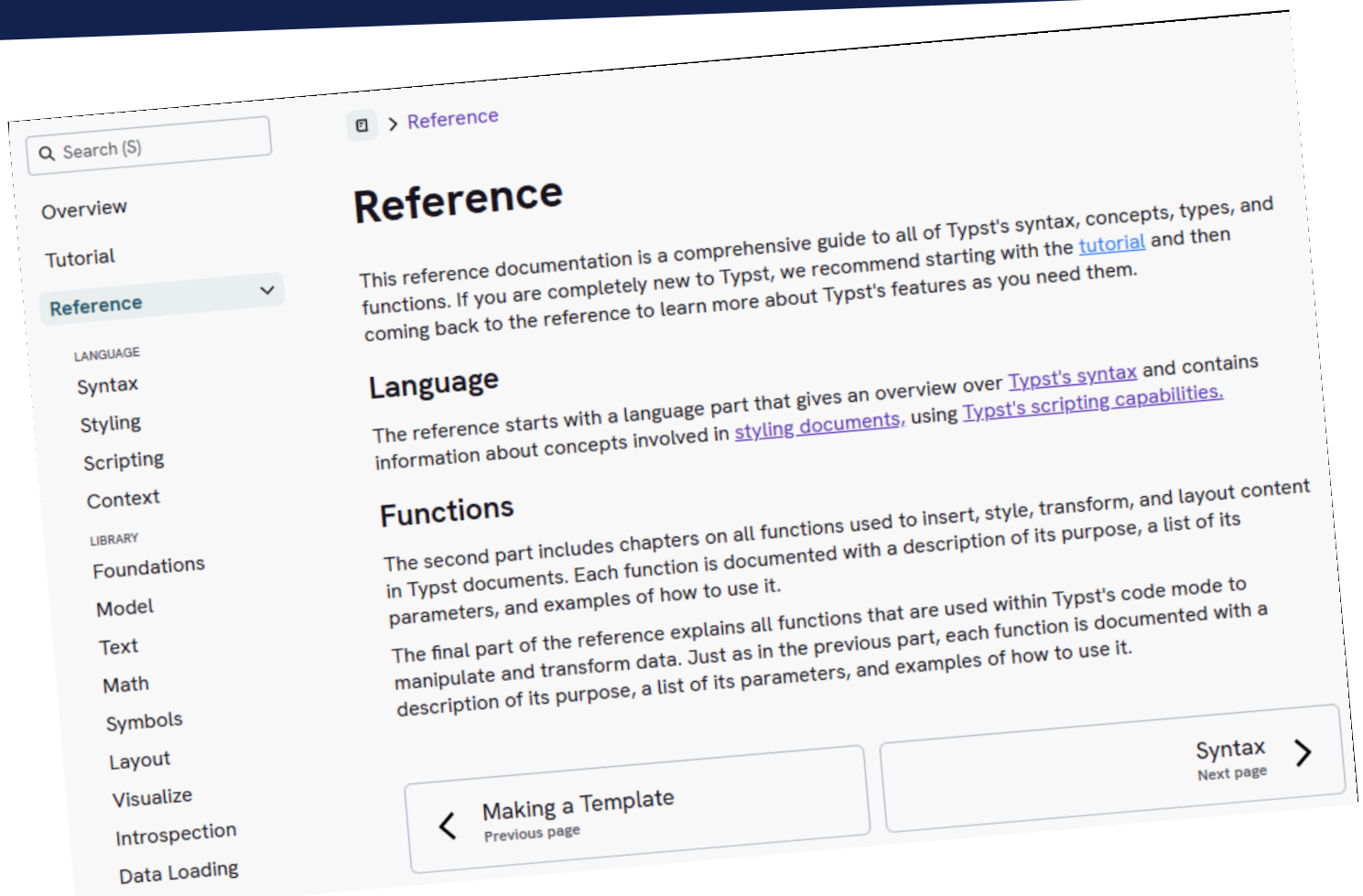


	Value in EUR
Initial Value	38,110.79
Ending Value	65,727.79
Deposits/Withdrawals	18,000
Yield	9,617
... by Asset	
— Stocks	7,581
— Bonds	226
— Futures	1,810
Cumulative Return	**25.23%**

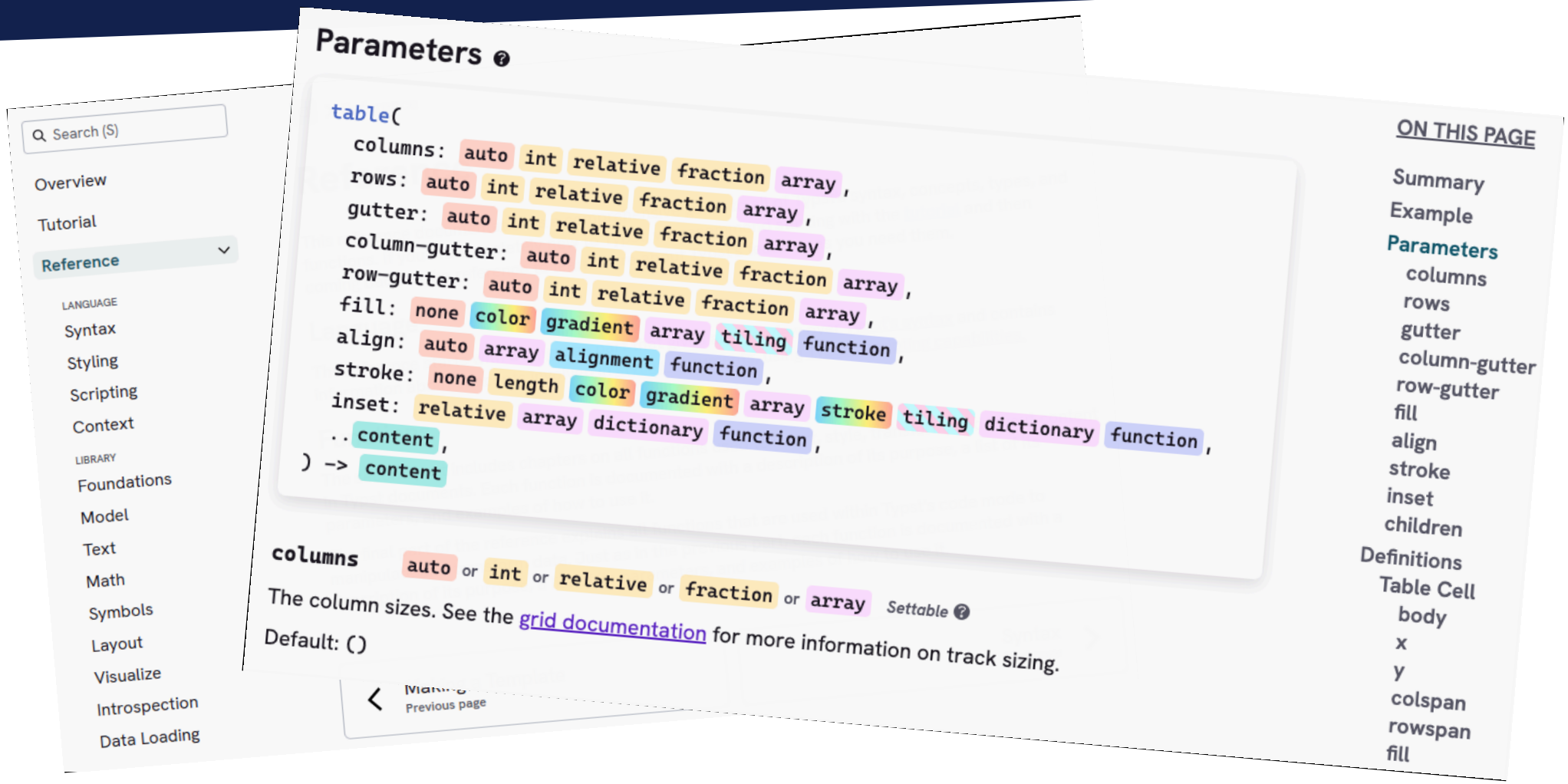
Showroom


kilobytes
Variety
Velocity
terabytes
static
dynamic
Add & Norm
Multi-Hea
Attentio
Inp
$\dot{x}_j = \frac{q}{c} \sum_l$
$F_{\mu,\nu} =$
$(ax + b)(cx +$
2023
Q1
Q2
Q3
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
Outcome
// 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);
return Ok(());
}
// 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 => {
let remaining: Abs = regions
.iter()
.map(|size| size.y)
.take(
self.config.columns.count - self.column

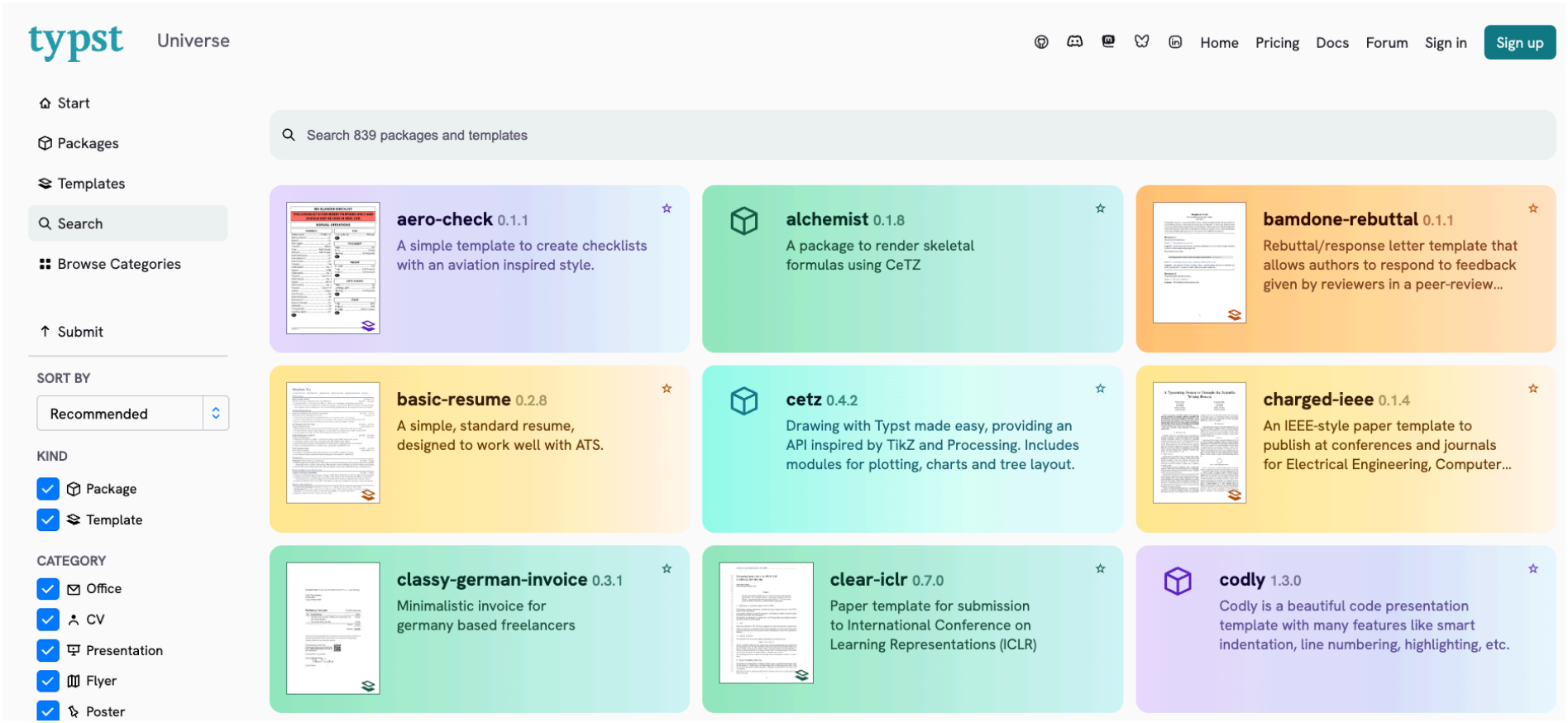
RTFM !


Search (S)
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

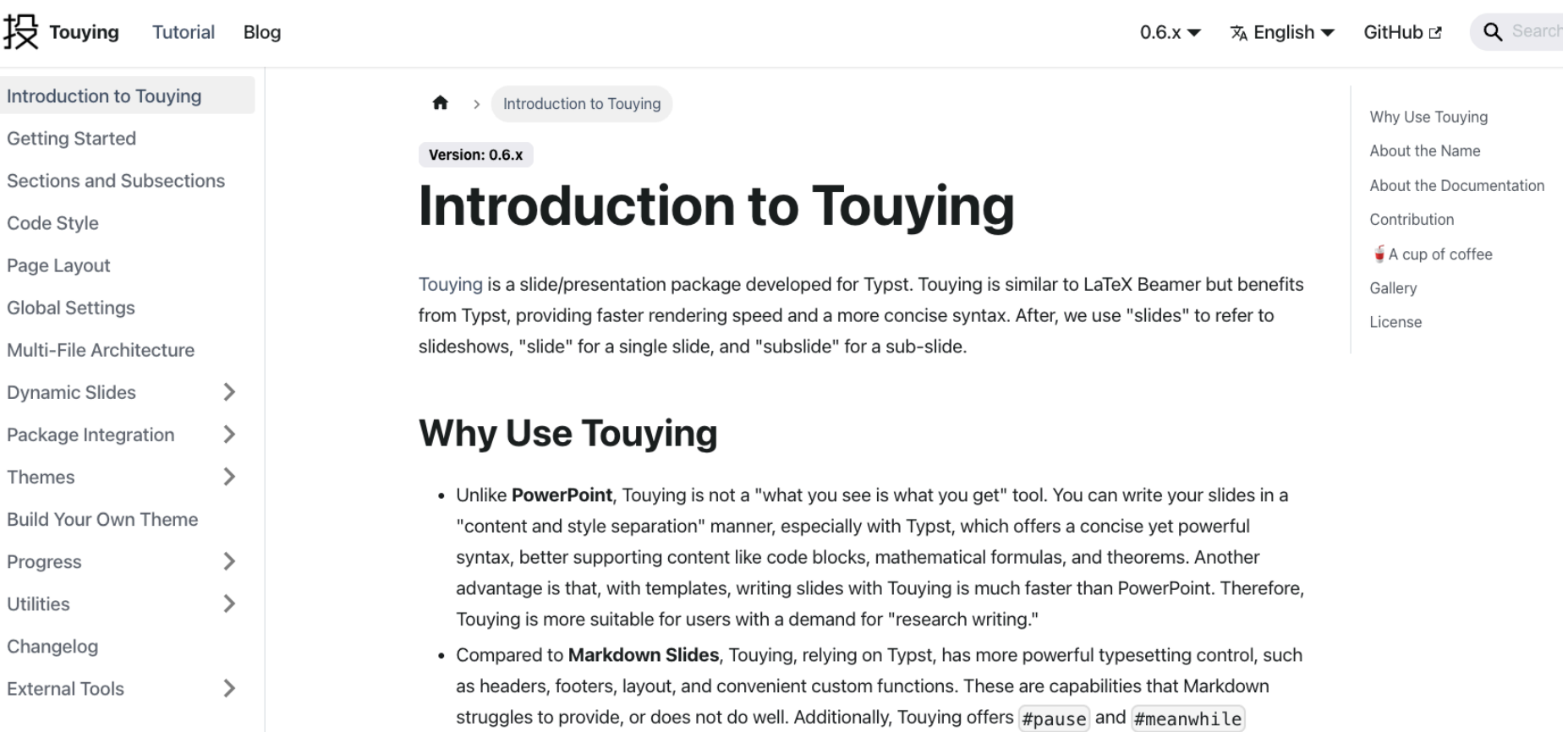
RTFM !


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: ()
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



Touying pour les slides




SHOW ME
NETFLIX

Les joies du code

Typst

```
#codly()
```rust
  fn main() {
  println!("Hello, there!");
  }
```
```

Les joies du code

Typst

```
#codly()
```rust
  fn main() {
  println!("Hello, there!");
  }
```
```

Rust

```rust
fn main() {
    println!("Hello, there!");
}
```

Les joies du code

Typst

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

Les joies du code

Typst

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

Java

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

Les tables

Typst

```
#table(
  columns: (1fr, 1fr),
  stroke: (paint: fuchsia, thickness: 3pt, dash: "dotted"),
  align: center,

  table.header([Prénom], [NOM]),

  [Ada], [#text(purple,[*LOVELACE*])],
  [Grace], [HOPPER],
  [Margaret], [HAMILTON],
)
```

Les tables

```typst
#table(
  columns: (1fr, 1fr),
  stroke: (paint: fuchsia, thickness: 3pt, dash: "dotted"),
  align: center,

  table.header([Prénom], [NOM]),

  [Ada], [#text(purple,[*LOVELACE*])],
  [Grace], [HOPPER],
  [Margaret], [HAMILTON],
)
```

Prénom	NOM
Ada	**LOVELACE**
Grace	HOPPER
Margaret	HAMILTON

On me voit, on me voit plus

Typst

```
#grid(
  columns: (auto, 30%),
  align: (horizon + center, horizon + center),

  [ #only("1")[On me voit] #only("3")[#text(red, weight: "bold", size: 35pt)[On me re-voit]] ],
  [ #only("1,3")[#image("../img/mcmahon1.png", width: 70%) ]],

  [#uncover("2")[ On le voit plus ]], [#uncover("2")[ #image("../img/mcmahon2.png", width: 70%) ]],

  [ ], [#uncover("3")[ #image("../img/mcmahon3.png", width: 70%) ]],
)
```

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 )
```

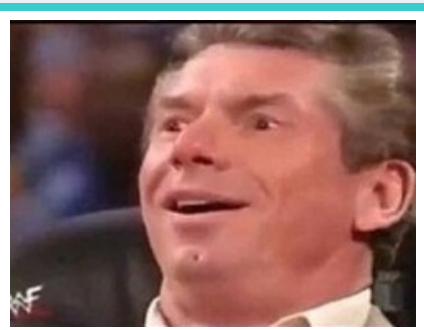
On me voit

On me voit, on me voit plus

Typst

```
#grid(
  columns: (auto, 30%),
  align: (horizon + center, horizon + center),

  [ #only("1")[On me voit] #only("3")[#te
  me re-voit]] ],
  [ #only("1,3")[#image("../img/mcmahon1.

  [#uncover("2")[ On le voit plus ]], [#u
  mcmahon2.png", width: 70%) ]],

  [ ], [#uncover("3")[ #image("../img/mcm
)
```

On le voit plus



On me voit, on me voit plus

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

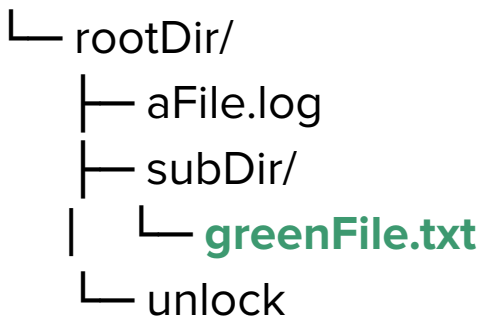
He’s back!

Des arborescences de fichiers

```typst
#tree-list[
  - rootDir/
    - aFile.log
    - subDir/
      - #text(olive, weight: "bold")
        [greenFile.txt]
    - unlock
]
```

Des arborescences de fichiers

```typst
#tree-list[
  - rootDir/
    - aFile.log
    - subDir/
      - #text(olive, weight: "bold")
        [greenFile.txt]
    - unlock
]
```



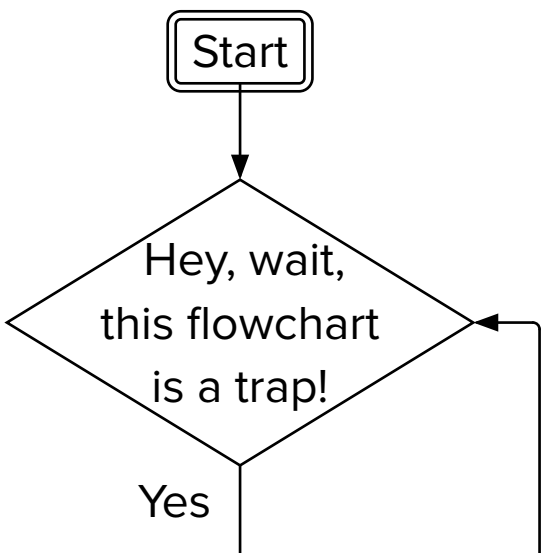
Des graphes aussi beaux qu'xkcd

Typst

```
#fletcher-diagram(
  node-stroke: 1pt,
  edge-stroke: 1pt,
  spacing: 3em,

  node((0, 0), [Start], corner-radius: 2pt, extrude: (0, 3)),
  edge("d", "-|>"),
  node((0, 1), align(center)[ Hey, wait,\ this flowchart\ is a trap! ], shape: diamond),

  edge("d,r,u,l", "-|>", [Yes], label-pos: 0.1),
)
```

Des graphes aussi beaux qu'xkcd

```typst
#fletcher-diagram(
  node-stroke: 1pt,
  edge-stroke: 1pt,
  spacing: 3em,

  node((0, 0), [Start], corner-radius: 2pt, extrude: (0, 3)),
  edge("d", "-|>"),
  node((0, 1), align(center)[ Hey, wait,\ this flowchart\ is a trap! ], shape: diamond),

  edge("d,r,u,l", "-|>", [Yes], label-pos: 0.1),
)
```

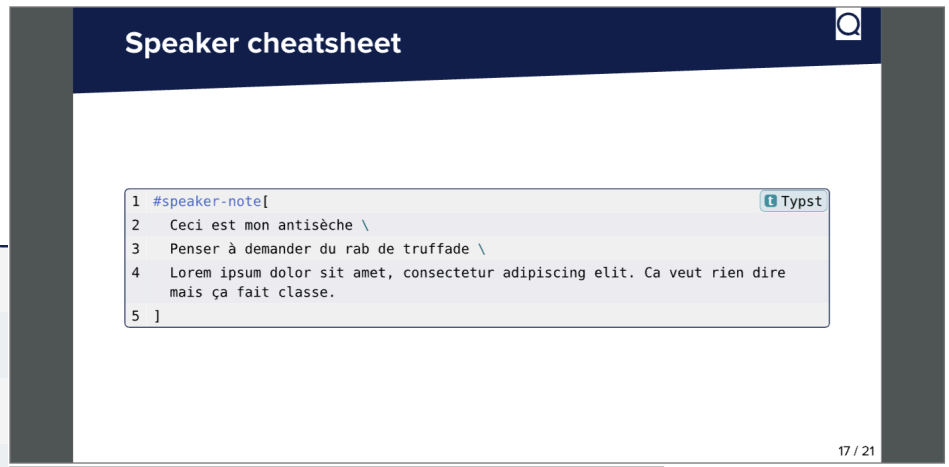


Speaker cheatsheet

```typst
#speaker-note[
  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.
]
```

Speaker cheatsheet

1
2
3
4
5



Next

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.

Génération des output

```Shell
typst compile fichier.typ
```

```Shell
touying compile fichier.typ
```

Génération des output

```shell
typst compile fichier.typ
```

➡️ `fichier.pdf`

```shell
touying compile fichier.typ
```

Génération des output

```shell
typst compile fichier.typ
```

➡️ `fichier.pdf`

```shell
touying compile fichier.typ
```

➡️ `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

LaTeX	Ty	Typst
15 minutes	30 secon	ades andalone 📝

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 princpiales

`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