

Modèle de présentation Typst

Présentations de type Beamer en Typst

25-05-2024

Template 0.1.0

Typst 0.11.1

Mathieu AUCEJO

Ce package Typst est une proposition de modèle de présentation de type Beamer en Typst pour les personnels du Laboratoire de Mécanique des Structures et des Systèmes Couplés du Conservatoire National des Arts et Métiers.

Table des matières

1. Introduction	1
2. Utilisation	1
2.1. Initilisation du modèle	2
2.2. Contenu de la présentation	3
2.2.1. Diapositive de titre	3
2.2.2. Diapositive de sommaire	5
2.2.3. Diapositive de section	5
2.2.4. Diapositive de contenu	6
2.2.5. Diapositive d'annexe	6
2.2.6. Diapositive « focus »	6
2.2.7. Fonctions complémentaires	7
3. Feuille de route	9
4. Note de versions	9

1. Introduction

Dans l'univers [Typst](#), plusieurs package propose des modèles de présentation de type Beamer. Il en existe actuellement 3 que l'on trouver sur [Typst Universe](#) :

- Polylux – [Typst: Universe](#), [dépôt Github](#) et [documentation](#)
- Touying – [Typst: Universe](#), [dépôt Github](#) et [documentation](#)
- Slydst – [Typst: Universe](#), [dépôt Github](#) et [documentation](#) (voir le ReadMe)

Pour créer ce modèle de présentation, mon choix s'est porté sur Polylux, car je le trouve plus simple d'utilisation que Touying (syntaxe plus typique de Typst) et permettant une plus grande évolutivité que Slydst.

2. Utilisation

Pour utiliser le modèle, il faut l'importer dans votre fichier principal typ en utilisant la commande suivante.

```
#import "../template/pres-template.typ": *
```

2.1. Initilisation du modèle

Après avoir importé le modèle, celui doit être initialisé en appliquant une règle d'affichage (show rule) avec la commande `#pres-template()` en passant les options nécessaires avec l'instruction `with` dans votre fichier principal `typ` :

```
#show pres-template.with(
  ...
)
```

Le modèle `#pres-template()` possède un certain nombre de paramètres permettant de personnaliser le document. Voici la liste des paramètres disponibles :

```
#pres-template(
  <aspect-ratio>: "16-9",
  <lang>: "fr",
  <logo>: "../images/logo_cnam.png",
  <footer-logo>: "../images/logo_cnam.png",
  <font>: "Noto Sans",
  <math-font>: "Noto Sans Math"
)[<body>]
```

Argument

`<aspect-ratio>: "16-9"`

str

Rapport d'aspect de la présentation. Autre format disponible : "4-3".

Argument

`<lang>: "fr"`

str

Langue du document. En fonction de la valeur prise par ce paramètre, la localisation du document sera adaptée.

Outre le français, la seule langue prise en compte est l'anglais (`lang: "en"`).

Argument

`<logo>: "../images/logo_cnam.png"`

str | array

Chemin vers les logos pour la disposition de titre.

Il faut que le template soit à la racine du répertoire pour que le chemin soit correctement interprété. Dans le cas contraire, une erreur de compilation sera générée.

Pour utiliser plusieurs logos, il faut créer une liste de liens vers les images.
Exemple - `logo: ("../images/logo1.png", "../images/logo1.png")`.

2.1. Utilisation

Argument

`<footer-logo>`: `"../images/logo_cnam.png"`

array

Chemin vers le logo principal.

Argument

`<body-font>`: `"Noto Sans"`

str

Nom de la police de caractère du corps du texte.

Argument

`<math-font>`: `"Noto Sans Math"`

str

Nom de la police de caractère des équations mathématiques.

Les polices de caractère doivent être préalablement installées sur votre système pour être utilisées dans le document.

Pour vérifier la disponibilité de la police choisie, vous pouvez entrer la commande `typst font` dans un terminal.

2.2. Contenu de la présentation

D'une manière générale, le fichier contenant votre présentation est structuré de la manière suivante :

```
// Diapositive de titre
#title-slide(...)

// Sommaire
#content-slide()

// Diapositive de section
#new-section-slide(...)

// Diapositive
#slide(...)[
  // Contenu de la diapositive
]

// Pour réveiller l'auditoire
#focus-slide[
  // Contenu de la diapositive
]

// Annexes
#appendix-slide[
  // Contenu de la diapositive
]
```

Le contenu de la présentation est structuré grâce à six types de diapositives, dont les spécificités sont décrites dans les sections qui suivent.

2.2.1. Diapositive de titre

2.2.1. Utilisation

La diapositive de titre est définie par la fonction `#title-slide()` qui dispose d'un certain nombre de paramètres permettant de personnaliser la diapositive. Voici la liste des paramètres disponibles :

```
#title-slide(<author>: "Votre nom", <laboratory>: "Laboratoire de  
recherche", <title>: "Titre de la présentation", <short-title>: "Titre  
court") [<body>]
```

Argument

`<author>`

str | content

Auteurs de la présentation.

Exemple

```
#let author = [#text(fill: colors.red, [Author 1]) #h(1em)  
Author 2]
```

Argument

`<laboratory>`

str | content

Nom du laboratoire de recherche.

Exemple

```
#let laboratory = [Laboratoire de Mécanique des Structures et  
des Systèmes Couplés \  
Conservatoire National des Arts et Métiers  
]
```

Argument

`<title>`

str | content

Titre de la présentation.

Exemple

```
#let title = [Titre \  
#line(length: 15%, stroke: 0.075em + red) \  
#text([Sous-titre], size: 0.8em)  
]
```

Argument

`<short-title>`

str

Titre court de la présentation.



Figure 1 – Exemple de diapositive de titre

2.2.2. Diapositive de sommaire

Pour créer une diapositive de sommaire, il faut insérer la commande `#content-slide()` dans votre fichier de présentation typ.

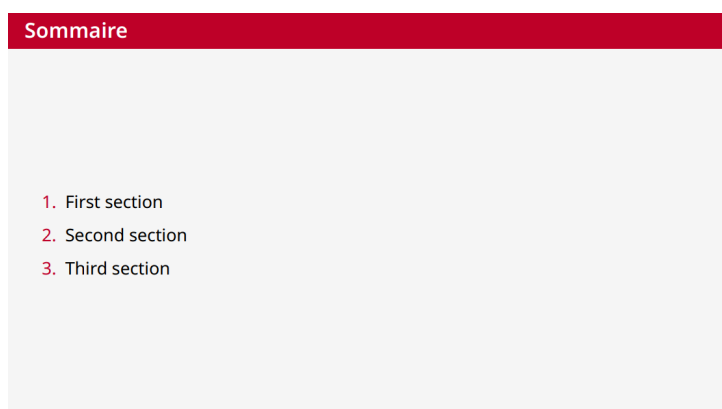


Figure 2 – Exemple de diapositive de sommaire

2.2.3. Diapositive de section

Pour créer une diapositive de section, il faut insérer la commande `#new-section-slide()` dans votre fichier de présentation typ. Cette commande prend en paramètre le titre de la section sous la forme d'un élément de type `str` ou `content`.

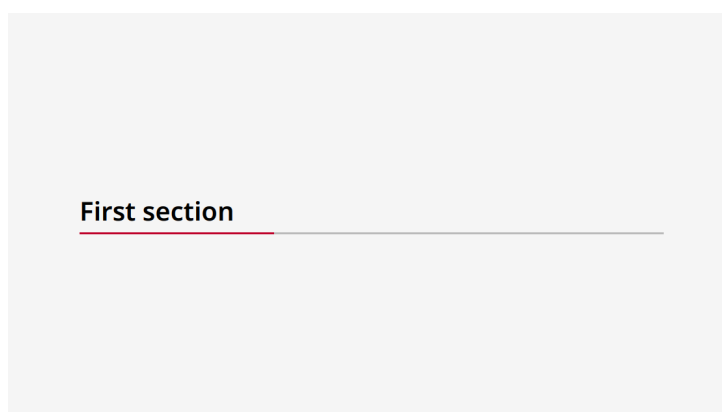


Figure 3 – Exemple de diapositive de section

2.2.4. Diapositive de contenu

Les diapositives de contenu sont les diapositives classiques de votre présentation. Elles sont définies par la fonction `#slide()` qui dispose de deux paramètres `title` et `subtitle` permettant de personnaliser la diapositive. Pour appeler la fonction, il suffit d'insérer la commande :

```
#slide(<title>: none, <subtiltle>: none)[<body>]
```

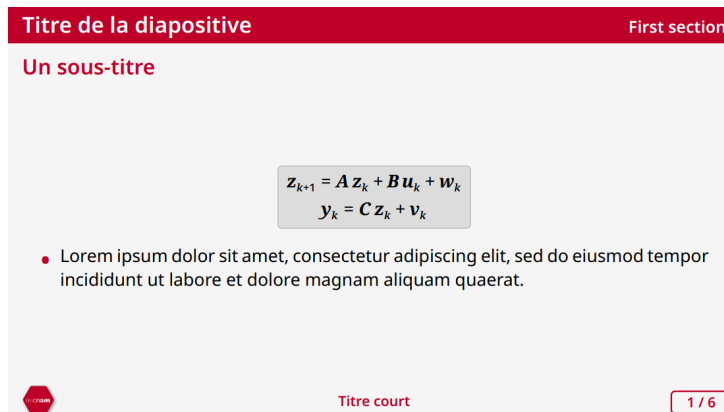


Figure 4 – Exemple de diapositive standard

2.2.5. Diapositive d'annexe

Pour créer une diapositive d'annexe, il faut insérer la commande `#appendix-slide()` dans votre fichier de présentation typ.

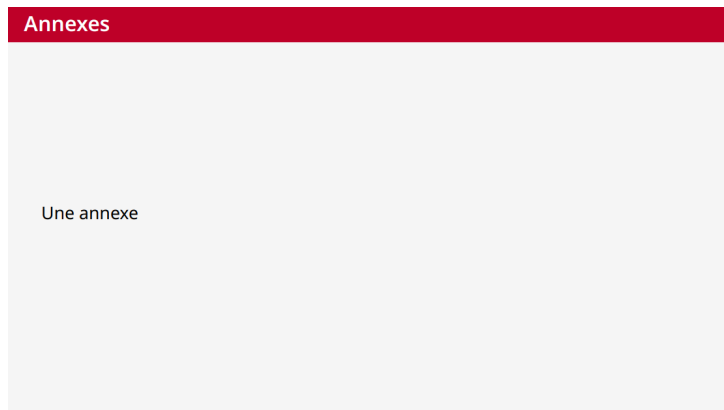


Figure 5 – Exemple de diapositive d'annexe

Par défaut, les diapositives d'annexe ne sont pas numérotées

2.2.6. Diapositive « focus »

Pour créer une diapositive pour attirer l'attention de votre auditoire, il faut insérer la commande `#focus-slide()` dans votre fichier de présentation typ.



Figure 6 – Exemple de diapositive « focus »

2.2.7. Fonctions complémentaires

Le modèle de présentation Typst propose plusieurs fonctions complémentaires permettant d'enrichir le contenu des diapositives. Voici la liste des fonctions disponibles :

- `#boxeq()` – Cette fonction permet de créer une boîte autour d'une équation mathématique importante.

Exemple

```
#boxeq[$sum_{i=1}^n i = (n(n+1))/2$]
```

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

- Boîtes permettant de mettre en avant des informations importantes dans votre présentation.
 - `#info()` – Boîte d'information
 - `#tip()` – Boîte de conseil
 - `#important()` – Boîte pour les remarques importantes
 - `#question()` – Boîte pour les questions

Exemple

```
#info[#lorem(10)]
#tip[#lorem(10)]
#important[#lorem(10)]
#question[#lorem(10)]
```

❗ Remarque

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

2.2.7. Utilisation

Astuce

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Important

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Question

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

- `#code()` – Cette fonction permet d’insérer un bloc de code dans votre présentation. Elle prend en paramètre le langage de programmation utilisé pour la coloration syntaxique.

Exemple

```
#code(lang:"Julia")[\n  ``julia\n  # Un commentaire\n  function squared(x)\n    return x^2\n  end\n  ``\n]
```

Code

Julia

```
1 # Un commentaire\n2 function squared(x)\n3   return x^2\n4 end
```

- `#link-box()` – Cette fonction permet de créer une boîte pour faire des liens entre les diapositives.

Exemple

```
#slide[\n  #link-box(<slide2>, "Aller à la diapositive 2")\n] <slide1>\n\n#slide[\n  #link-box(<slide1>, "Retour à la diapositive 1")\n] <slide2>
```


3. Feuille de route

Le modèle est en cours de développement. Voici la liste des fonctionnalités qui sont implémentées ou le seront dans une prochaine version.

Types de diapositives

- ✓ Diapositive de titre
- ✓ Diapositive de sommaire
- ✓ Diapositive de section
- ✓ Diapositive de contenu
- ✓ Diapositive d'annexe
- ✓ Diapositive pour focaliser l'attention de l'auditoire

Éléments de mise en page

- ✓ Personnalisation de l'en-tête des diapositives standard avec l'affichage du titre de la diapositive et du nom section courante
- ✓ Personnalisation du pied de page (logo, titre court de la présentation, numérotation des diapositives)
- ✓ Numérotation des diapositives d'annexe en lettre + chiffres (A | 1/1)

Éléments de contenu

- ✓ Création d'une boîte autour d'une équation mathématique via la fonction `boxeq`
- ✓ Blocs de code avec coloration syntaxique via le package `code1st`
- ✓ Boîtes permettant de mettre en avant certains contenus via le package `showybox`
- ✓ Création de boîte pour faire des liens entre les diapositives

4. Note de versions

- `polylux` – version 0.3.1
- `code1st` – version 2.0.1
- `showybox` – version 2.0.1