



## Documentation du package course-style

### A ♦ Système de versioning git worktrees

Ce template utilise un système de versioning avec **git worktrees** :

#### ① INFORMATION

##### Structure :

- `current/` : Version de développement (repo git principal)
- `0.0.1/` , `0.0.2/` ... : Versions figées (worktrees pointant sur des tags)
- Import selon le contexte : `@local/course-style:current` vs `@local/course-style:0.0.1`

### A1 ▶ Configuration VSCode Tinymist

Pour activer les imports `@local` , ajouter à `settings.json` :

```
{
  "tinymist.compileArgs": [
    "--root",
    "C:/Users/votre-nom/Documents/_Education Nationale/Cours/_templates"
  ]
}
```

### B ♦ Installation et configuration

#### B1 ▶ Installation du package

1. **Créer la structure de dossiers** : Créez le dossier suivant dans votre répertoire de packages Typst :

`C:\Users\xxxx\AppData\Local\typst\packages\local\course-style\0.0.1`

2. **Copier les fichiers** :

- Placez le fichier `style.typ` dans ce dossier
- Placez le fichier `typst.toml` dans ce dossier
- Ajoutez votre logo `logo.png` **au même niveau** que `style.typ`

3. **Structure finale** :

```
C:\Users\xxxx\AppData\Local\typst\packages\local\course-style\0.0.1
├─ typst.toml
├─ style.typ
└─ logo.png
```

#### B2 ▶ Configuration du package (typst.toml)

Le fichier `typst.toml` doit contenir la configuration suivante :

```
[package]
name = "course-style"
version = "0.0.1"
entrypoint = "style.typ"
authors = ["CADI TAZI Tawfiq"]
```

```
license = "MIT"
description = "Template for educational documents and courses"
```

# C ♦ Exemple d'arborescence de titres

## C1 ▶ Titre de niveau 2 (exemple)

### C1.1 ♦ Titre de niveau 3 (exemple)

#### C1.1.1 • Titre de niveau 4 (exemple de hiérarchie)

##### C1.1.1.1 ▶ Titre de niveau 5 (structure complète)

Ce document présente toutes les fonctionnalités disponibles dans le module `course-style:0.0.1` pour créer des documents pédagogiques.

# D ♦ Configuration de base

## D1 ▶ Import et utilisation

```
#import "@local/course-style:0.0.1": *

#show: doc => COURSE(
  class: "Seconde",
  level: "SNT",
  duration: "2h",
  title: "Mon cours",
  doc
)
```

- `class` : Classe concernée (défaut: « Seconde »)
- `level` : Niveau/matière (défaut: « SNT »)
- `duration` : Durée du cours (défaut: « 2h »)
- `title` : Titre du document (défaut: « Les données structurées »)
- `academy` : Académie (défaut: « ACADÉMIE »)
- `school` : Établissement (défaut: « Lycée XX »)
- `logo_path` : Chemin du logo (défaut: « logo.png »)
- `logo_width` : Taille du logo (défaut: 1cm)
- `cell_fill_color` : Couleur des cellules signature (défaut: yellow)

# E ♦ Encadrés personnalisés

## E1 ▶ Encadré générique

### Mon titre

```
#format_box(title: "Mon titre", color: blue)
[Contenu de l'encadré]
```

### Titre vert

```
#format_box(title: "Titre vert", color: green,
breakable: true)[Contenu cassable]
```

### Variables

```
#definition_box(title: "Variables")[Une
variable stocke une valeur]
```

### EXEMPLE

```
#example_box[Voici un exemple concret
d'utilisation]
```

### ❗ INFORMATION

```
#info_box[Information importante à retenir]
```

### ⚠ ATTENTION

```
#warning_box[Attention : risque d'erreur !]
```

### 📌 À retenir

```
#synthesis_box(title: "À retenir", color: orange)[Points clés du chapitre]
```

## E2 ▶ Activités numérotées

La fonction `#activity()` permet de créer des en-têtes d'activités avec numérotation automatique :

Activité sans titre :

```
#activity("")
```

🎯 Activité 1 🎯

Activité avec titre :

```
#activity("Découverte de Python")
```

🎯 Activité 2 - Découverte de Python 🎯

❗ INFORMATION

**Fonctionnement :**

- Numérotation automatique globale (ne se réinitialise pas)
- Style cohérent avec les encadrés pédagogiques
- Icône 🎯 pour identifier rapidement les activités
- Titre optionnel pour décrire l'activité

E3 ▶ Tableaux vides

Tableau par défaut (solide) :

```
#empty_table(cols: 3, rows: 4)
```


Tableau en pointillés :

```
#empty_table(cols: 4, rows: 3, solid: false)
```


Tableau entièrement en pointillés (y compris en-tête) :

```
#empty_table(cols: 4, rows: 3, solid: false, header_solid: false)
```


Tableau avec lignes alternées :

```
#empty_table(cols: 5, rows: 4, striped: true)
```



Tableau avec en-tête coloré :

```
#empty_table(cols: 4, rows: 3, header_fill: true)
```


Tableau complet (solide + rayé + en-tête coloré) :

```
#empty_table(cols: 3, rows: 5, striped: true, header_fill: gray.darken(20%))
```


F ♦ Système de corrections

F1 ▶ Lignes d'écriture

Lignes complètes sans correction :

```
#set_correction(false)
#full_lines(count: 2, correction: "Première réponse seulement")
```

Lignes complètes avec correction :

```
#set_correction(true)
#full_lines(count: 3, correction: "Première réponse seulement")
```

Première réponse seulement

Ligne à compléter sans correction :

```
#set_correction(false)
Nom : #lines()
```

Nom : \_\_\_\_\_

**Ligne à compléter avec correction :**

```
#set_correction(true)
```

Nom : `#lines`(correction: "Jean Dupont")

Nom : Jean Dupont

**Réponse multi-lignes avec correction sur la première ligne :**

```
#set_correction(true)
```

Que pensez-vous ? `#lines`(count: 3, correction: "C'est très pratique pour les corrections")

Que pensez-vous ? C'est très pratique pour les corrections

## F2 ► Configuration des corrections

```
#set_correction(true) // Active les corrections
```

```
#set_correction(false) // Masque les corrections
```

## F3 ► Trous à compléter

```
#set_correction(false)
```

```
#gap(width: 6cm)
```

Voici un trou vide :

```
#set_correction(true)
```

```
#gap(width: 6cm, correction: "La réponse")
```

Trou avec correction :

La réponse

## F4 ► Blocs de réponse

**Bloc de réponse sans correction :**

```
#set_correction(false)
```

```
#response_block(height: 3cm, correction: "Réponse attendue dans ce bloc")
```

Que remarquez-vous sur cette image ?

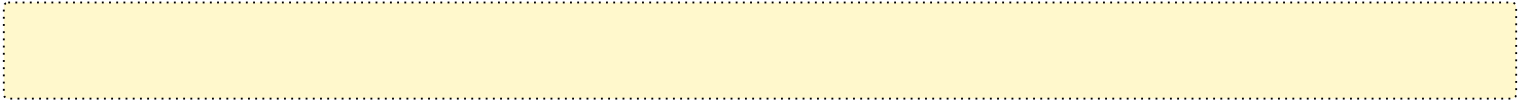
**Bloc de réponse avec correction :**

```
#set_correction(true)
```

```
#response_block(height: 3cm, correction: "Réponse attendue dans ce bloc")
```

Expliquez le concept de variable :

Une variable est un espace de stockage nommé qui contient une valeur



Bloc personnalisé (largeur, encadrement) :

```
#set_correction(true)
#response_block(
  height: 2.5cm,
  width: 80%,
  inset: 1cm,
  stroke: (dash: "solid", thickness: 1pt),
  correction: "Réponse courte"
)
```

Résumé en une phrase :

Réponse courte et précise

G ♦ Modules externes intégrés

G1 ▶ gentle-clues : Encadrés pédagogiques

🚩 Objectif

#goal(title: "Objectif")[Comprendre les variables en Python]

i Info

#info[Information générale sur le sujet]

⚠ Avertissement

#warning[Attention aux erreurs courantes !]

✓ Succès

#success[Bravo ! Exercice réussi]

✖ Erreur

#error[Erreur détectée dans le code]

🔥 Conseil

#tip[Astuce : utilisez Ctrl+S pour sauvegarder]

” Citation

#quote[Citation importante d'un auteur]

! À retenir

#memo[Mémo à retenir absolument]

☰ Tâche 1

#task[Tâche à accomplir avant la prochaine séance]