

# Leçon 5

## Personnalisation

# Règles set

## Code

```
// Appliquer à tout le document
#set text(size: 11pt)
#set page(margin: 2cm)
#set par(justify: true)
#set text(font: police)
```

## Rendu

`set` configure les éléments par défaut.

S'applique à tout ce qui suit.

On peut utiliser toute police installée sur le système. Sur [typst.app](https://typst.app), utilisez les polices disponibles dans le projet ou déjà installée dans l'application en ligne.

# Couleurs

## Code

```
// Couleurs prédéfinies
red, blue, green, gray

// Couleurs personnalisées
#let ma-couleur = rgb("#00695c")
blue.lighten(30%)
red.darken(20%)

#text(fill: blue)[Texte bleu]
#rect(fill: ma-couleur.lighten(70%),
      stroke: ma-couleur)[Encadré]
```

## Rendu

Texte bleu

Encadré

# Manipulation des couleurs

```
couleur.lighten(50%)    // Éclaircir  
couleur.darken(20%)    // Assombrir  
couleur.transparentize(30%) // Transparent
```



# Changer la police

## Code

```
// Police globale
#set text(font: "DejaVu Serif", size:
11pt)

Texte en serif

// Police locale
#text(font: "Open Sans")[
  Texte en sans-serif
]

// Avec fallback
#set text(font: ("DejaVu Serif", "Open
Sans"))
```

## Rendu

Texte en serif

Texte en sans-serif

# Réutiliser avec `.with()`

## Code

```
#let my-tasks = tasks.with(  
  columns: 2,  
  label-format: "a)",  
)  
  
#my-tasks[  
  + Première réponse  
  + Deuxième réponse  
]
```

## Rendu

a) Première  
réponse

b) Deuxième  
réponse

`.with()` crée une version configurée d'une fonction. Cela permet d'avoir facilement, dans un même document, des listes en `1)` et d'autres en `a)` par exemple.

# Utiliser le template

```
#import "../shared/template.typ": *  
  
#show: math-course-template.with(  
  course: "1MA - EG - 2025-2026",  
  license: "CC BY-NC-SA",  
)
```

Le template centralise les règles `set`, `show`, les packages et les helpers.

# Liens utiles : templates

- Templates sur Typst Universe :  
<https://typst.app/universe/search/?kind=templates>
- `texam`, template pour évaluations :  
<https://typst.app/universe/package/texam>

# Liens utiles : packages

- `beautiframe` :  
<https://typst.app/universe/package/beautiframe>
- `beautitled` :  
<https://typst.app/universe/package/beautitled>
- `exercise-bank` :  
<https://typst.app/universe/package/exercise-bank>

# Titres avec beautitled

## Code

```
#import "@preview/beautitled:0.2.6":  
  beautitled-setup  
  
#beautitled-setup(  
  style: "titled",  
  chapter-prefix: "Chapitre",  
)  
  
= Mon chapitre  
== Ma section
```

## Rendu

### Chapitre 1

## Mon chapitre

### 1.1 Ma section

Styles disponibles : `titled`, `schoolbook`, `modern`

# Environnements beautiframe

## Code

```
#import "@preview/beautiframe:0.3.0":  
  definition, theorem, remark  
  
#definition(name: "Fonction") [  
  Une fonction  $f$  associe...  
]  
  
#theorem(name: "Pythagore") [  
   $a^2 + b^2 = c^2$   
]  
  
#remark [  
  Attention au cas particulier.  
]
```

## Rendu

### Définition (Fonction)

Une fonction  $f$  associe...

### Théorème (Pythagore)

$$a^2 + b^2 = c^2$$

Dans cette formation, `shared/template.typ` fait déjà cet import.

# Configuration beamerframe

## Code

```
#preset-french()
#beamerframe-setup(
  style: "modern",
  theorem-variant: "prominent",
  definition-variant: "standard",
  remark-variant: "minimal",
  accent-color: rgb("#00695c"),
)

#definition(name: "Fonction affine")[
  Une fonction de la forme...
]

#theorem(name: "Signe")[
  Si  $a > 0$ , la fonction est
  croissante.
]
```

## Rendu

### Définition 1 — *Fonction affine*

---

Une fonction de la forme ...

### Théorème 1 — *Signe*

---

Si  $a > 0$ , la fonction est  
croissante.

Dans cette formation, cette configuration est déjà dans `shared/template.typ`.

# Tableaux de signes

## Code

```
#sign-table(  
  factors: (  
    (expr:  $x - 2$ ,  
      zeros: ((value:  $2$ , approx:  $2$ )),  
      signs: ("-", "+")),  
    (expr:  $x + 1$ ,  
      zeros: ((value:  $-1$ , approx:  
-1)),  
      signs: ("-", "+")),  
  ),  
  summary-label:  $f(x)$ ,  
  variation: true,  
)
```

## Rendu

| $x$     | $-1$    | $2$   |
|---------|---------|-------|
| $x - 2$ | $-$     | $-0+$ |
| $x + 1$ | $-0+$   | $+$   |
| $f(x)$  | $+0-0+$ |       |

`sign-table` est fourni par le template du cours.

# Exercices avec exercise-bank

## Code

```
#exo(  
  exercise: [  
    Résoudre  $x^2 - 4 = 0$ .  
  ],  
  solution: [  
     $x^2 = 4$  donc  $x = \pm 2$   
  ],  
)
```

## Rendu

### Exercice

Résoudre  $x^2 - 4 = 0$ .

**Solution :**  $x = \pm 2$

# Configuration des exercices

## Code

```
#exo-setup(  
  display: "both",  
  solution-label: "Corrigé",  
  exercise-label: "Question",  
  badge-style: "box",  
  correction-color: rgb("#1f4b99"),  
)  
  
#exo(  
  exercise: [Calculer $2 + 3$.],  
  solution: [$5$],  
)
```

## Rendu

**Question 1** Calculer  $2 + 3$ .

**Corrigé 1** 5

Configurer une fois, appliquer partout.

# Checklist

- ☐ Utiliser `set` pour les réglages globaux
- ☐ Utiliser `show` pour transformer l'affichage
- ☐ Personnaliser police et couleurs
- ☐ Importer le template
- ☐ Utiliser `beautitled`, `beautiframe`, `exercise-bank`

**Leçon suivante : helpers**